Open Source Used In Webex App Android ford-aaos-43.12

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Text Part Number: 78EE117C99-1846694297

This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please submit this form.

In your requests please include the following reference number 78EE117C99-1846694297

Contents

1.1 opus 1.3.1

1.1.1 Available under license :

are met:

- Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

- Neither the name of Internet Society, IETF or IETF Trust, nor the
names of specific contributors, may be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
``AS IS'' AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Opus is subject to the royalty-free patent licenses which are
specified at:

Xiph.Org Foundation:
https://datatracker.ietf.org/ipr/1524/

Microsoft Corporation:
https://datatracker.ietf.org/ipr/1914/

Broadcom Corporation:
https://datatracker.ietf.org/ipr/1526/

1.2 libilbc 2.0.2

1.2.1 Available under license :

Copyright (c) 2011, The WebRTC project authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the
distribution.

* Neither the name of Google nor the names of its contributors may
be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
EVENT SHALL THE COPYRIGHT
HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.3 libsrtp 2.0.0

1.3.1 Available under license :

/*
*
* Copyright (c) 2001-2006 Cisco Systems, Inc.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
*
* Redistributions in binary form must reproduce the above
* copyright notice, this list of conditions and the following
* disclaimer in the documentation and/or other materials provided
* with the distribution.
*
* Neither the name of the Cisco Systems, Inc. nor the names of its

* contributors may be used to endorse or promote products derived
* from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
 AND FITNESS
* FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
* COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
* INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
*
*/

1.4 asm 5.0.4

1.4.1 Available under license :

No license file was found, but licenses were detected in source scan.

2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
* contributors may be used to endorse or promote products derived from
* this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN

* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.

Found in path(s):
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/package.html
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/util/package.html
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/package.html
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/signature/package.html
*
/opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/analysis/package.html
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/package.html
No license file was found, but licenses were detected in source scan.

/***
* ASM XML Adapter
* Copyright (c) 2004-2011, Eugene Kuleshov
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*   contributors may be used to endorse or promote products derived from
*   this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/xml/SAXClassAdapter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/xml/SAXAdapter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/xml/SAXAnnotationAdapter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/xml/Processor.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/xml/SAXFieldAdapter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/xml/ASMContentHandler.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/xml/SAXCodeAdapter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/xml/asm-xml.dtd

No license file was found, but licenses were detected in source scan.

```
/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
/**
   * Creates a new {@link GeneratorAdapter}. <i>Subclasses must not use this
   * constructor</i>. Instead, they must use the
   * {@link #GeneratorAdapter(int, MethodVisitor, int, String, String)}
   * version.
   *
   * @param mv
   *            the
 method visitor to which this adapter delegates calls.
   * @param access
   *            the method's access flags (see {@link Opcodes}).
   * @param name
   *            the method's name.
   * @param desc
   *            the method's descriptor (see {@link Type Type}).
   * @throws IllegalStateException
```

```
    *            If a subclass calls this constructor.
    */
```

Found in path(s):
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/GeneratorAdapter.java
No license file was found, but licenses were detected in source scan.

```
/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
/**
    * Creates a new JSRInliner. <i>Subclasses must not use this
    * constructor</i>. Instead, they must use the
    * {@link #JSRInlinerAdapter(int, MethodVisitor, int, String, String, String, String[])}
    * version.
    *
    * @param mv
    *
   the <code>MethodVisitor</code> to send the resulting inlined
    *            method code to (use <code>null</code> for none).
```

```
   * @param access
   *          the method's access flags (see {@link Opcodes}). This
   *          parameter also indicates if the method is synthetic and/or
   *          deprecated.
   * @param name
   *          the method's name.
   * @param desc
   *          the method's descriptor (see {@link Type}).
   * @param signature
   *          the method's signature. May be <tt>null</tt>.
   * @param exceptions
   *          the internal names of the method's exception classes (see
   *          {@link Type#getInternalName() getInternalName}). May be
   *          <tt>null</tt>.
   * @throws IllegalStateException
   *           If a subclass calls this constructor.
   */
```

Found in path(s):
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/JSRInlinerAdapter.java
No license file was found, but licenses were detected in source scan.

```
/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
```

```
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
/**
   * Constructs a new {@link ClassNode}. <i>Subclasses must not use this
   * constructor</i>. Instead, they must use the {@link #ClassNode(int)}
   * version.
   *
   * @throws IllegalStateException
   *             If a subclass calls this constructor.
    */
```

Found in path(s):
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/ClassNode.java
No license file was found, but licenses were detected in source scan.

```
/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
```

```
/**
   * Constructs a new {@link Textifier}. <i>Subclasses must not use this
   * constructor</i>. Instead, they must use the {@link #Textifier(int)}
   * version.
   *
   * @throws IllegalStateException
   *             If a subclass calls this constructor.
    */
```

Found in path(s):
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/util/Textifier.java
No license file was found, but licenses were detected in source scan.

```
/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
/**
   * Constructs a new {@link LocalVariableAnnotationNode}. <i>Subclasses must
   * not use this constructor</i>. Instead, they must use the
   * {@link #LocalVariableAnnotationNode(int, TypePath, LabelNode[], LabelNode[], int[], String)}
   * version.
```

```
    *

    * @param typeRef
    *          a reference to the annotated type. See {@link TypeReference}.
    * @param typePath
    *          the path to the annotated type argument, wildcard bound, array
    *          element type, or static inner type within 'typeRef'. May be
    *          <tt>null</tt> if the annotation targets 'typeRef' as a whole.
    * @param start
    *          the fist instructions corresponding to the continuous ranges
    *          that make the scope of this local variable (inclusive).
    * @param end
    *          the last instructions corresponding to the continuous ranges
    *          that make the scope of this local variable (exclusive). This
    *          array must have the same size as the 'start' array.
    * @param index
    *          the local variable's index in each range. This array must have
    *          the same size as the 'start' array.
    * @param desc
    *          the class descriptor of
 the annotation class.
    */
```

Found in path(s):
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/LocalVariableAnnotationNode.java
No license file was found, but licenses were detected in source scan.

```
/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
```

* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/Label.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/Type.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/AnnotationWriter.java
*
 /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/analysis/SimpleVerifier.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/optimizer/NameMapping.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/util/Printer.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/optimizer/Shrinker.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/Method.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/util/CheckAnnotationAdapter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/RemappingSignatureAdapter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/RemappingFieldAdapter.java
*
 /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/ClassWriter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/Attribute.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/analysis/SourceInterpreter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/MethodInsnNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/MethodVisitor.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/util/TraceFieldVisitor.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/analysis/SourceValue.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/AdviceAdapter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/Remapper.java

*
/opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/analysis/BasicValue.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/FieldWriter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/ByteVector.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/AbstractInsnNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/analysis/Interpreter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/JumpInsnNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/TableSwitchGenerator.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/FieldNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/FieldInsnNode.java
*
/opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/LabelNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/optimizer/ClassConstantsCollector.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/util/CheckSignatureAdapter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/analysis/BasicVerifier.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/CodeSizeEvaluator.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/InstructionAdapter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/IntInsnNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/RemappingClassAdapter.java
*
/opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/MultiANewArrayInsnNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/ClassReader.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/VarInsnNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/RemappingMethodAdapter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/analysis/Value.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/StaticInitMerger.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/optimizer/AnnotationConstantsCollector.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/AnnotationVisitor.java

* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/InsnNode.java
*
/opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/signature/SignatureReader.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/tree/LdcInsnNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/tree/LookupSwitchInsnNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/tree/LineNumberNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/optimizer/JarOptimizer.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/commons/LocalVariablesSorter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/util/TraceAnnotationVisitor.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/optimizer/Constant.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/tree/TypeAnnotationNode.java
*
/opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/Handle.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/tree/TryCatchBlockNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/optimizer/ClassOptimizer.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/util/ASMifier.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/util/TraceClassVisitor.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/tree/TableSwitchInsnNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/tree/analysis/Frame.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/tree/InvokeDynamicInsnNode.java
*
/opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/tree/analysis/AnalyzerException.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/tree/analysis/Subroutine.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/commons/SerialVersionUIDAdder.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/tree/InnerClassNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/tree/ParameterNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-
jar/org/objectweb/asm/tree/LocalVariableNode.java

* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/analysis/Analyzer.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/analysis/SmallSet.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/RemappingAnnotationAdapter.java
*
 /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/TypeInsnNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/util/TraceSignatureVisitor.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/IincInsnNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/optimizer/MethodOptimizer.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/util/CheckFieldAdapter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/Frame.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/util/CheckMethodAdapter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/SimpleRemapper.java
*
 /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/AnnotationNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/FieldVisitor.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/Context.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/util/TraceMethodVisitor.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/optimizer/MethodConstantsCollector.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/Handler.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/MethodNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/InsnList.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/optimizer/FieldConstantsCollector.java
*
 /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/MethodWriter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/analysis/BasicInterpreter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/signature/SignatureWriter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/TryCatchBlockSorter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/Item.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/Edge.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/ClassVisitor.java

* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/tree/FrameNode.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/optimizer/ConstantPool.java
*
/opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/Opcodes.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/util/CheckClassAdapter.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/signature/SignatureVisitor.java
No license file was found, but licenses were detected in source scan.

/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*   contributors may be used to endorse or promote products derived from
*   this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
/**
    * Creates a new {@link AnalyzerAdapter}. Subclasses must not use this
    * constructor. Instead, they must use the
    * {@link #AnalyzerAdapter(int, String, int, String, String, MethodVisitor)}
    * version.
    *

```
   * @param owner
   *
     the owner's class name.
   * @param access
   *          the method's access flags (see {@link Opcodes}).
   * @param name
   *          the method's name.
   * @param desc
   *          the method's descriptor (see {@link Type Type}).
   * @param mv
   *          the method visitor to which this adapter delegates calls. May
   *          be <tt>null</tt>.
   * @throws IllegalStateException
   *           If a subclass calls this constructor.
   */
```

Found in path(s):
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/commons/AnalyzerAdapter.java
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/xml/package.html
No license file was found, but licenses were detected in source scan.

#All rights reserved.
#Redistribution and use in source and binary forms, with or without
#modification, are permitted provided that the following conditions
#are met:
#1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
#2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
#3. Neither the name of the copyright holders nor the names of its
this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/optimizer/shrink-annotations.properties
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/optimizer/shrink-frames.properties
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/optimizer/shrink-resize.properties
*
 /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/optimizer/shrink-signatures.properties
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/optimizer/shrink.properties
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/optimizer/shrink-writer.properties
No license file was found, but licenses were detected in source scan.

/***
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2013 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*   contributors may be used to endorse or promote products derived from

```
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/
```

Found in path(s):
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/TypePath.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/TypeReference.java
No license file was found, but licenses were detected in source scan.

```
/**
* ASM: a very small and fast Java bytecode manipulation framework
* Copyright (c) 2000-2011 INRIA, France Telecom
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. Neither the name of the copyright holders nor the names of its
*    contributors may be used to endorse or promote products derived from
*    this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
```

* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*/

Found in path(s):
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/util/Textifiable.java
* /opt/cola/permits/1135863767_1613617914.89/0/asm-5-0-4-sources-6-jar/org/objectweb/asm/util/ASMifiable.java

1.5 commons-codec 1.14

1.5.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or
(iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as
of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own
copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names
of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law
(such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your

sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Apache Commons Codec
Copyright 2002-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from http://aspell.net/test/orig/batch0.tab.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

==========

The content of package org.apache.commons.codec.language.bm has been translated
from the original php source code available at http://stevemorse.org/phoneticinfo.htm
with permission from the original authors.

Original source copyright:
Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

1.6 commons-lang3 3.7

1.6.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache Commons Lang
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.7 guava 30.1.1-android

1.7.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
 * This following method is a modified version of one found in
 * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/src/test/tck/AbstractExecutorServiceTest.java?revision=1.30
 * which contained the following notice:
 *
 * Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to
 *
the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/
 *
 * Other contributors include Andrew Wright, Jeffrey Hayes, Pat Fisher, Mike Judd.
 */

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/MoreExecutors.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0

*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/net/package-info.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/Atomics.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Strings.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ThreadFactoryBuilder.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Ascii.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ContiguousSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/Monitor.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ForwardingBlockingQueue.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Equivalence.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/SortedLists.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/UncaughtExceptionHandlers.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/package-info.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/annotations/Beta.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ListeningExecutorService.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/annotations/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at

*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/eventbus/SubscriberRegistry.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/MoreObjects.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/math/Quantiles.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/eventbus/Subscriber.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/eventbus/Dispatcher.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/TrustedListenableFutureTask.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ListenerCallQueue.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/*
  * This method was rewritten in Java from an intermediate step of the Murmur hash function in
  * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
  * following header:
  *

```
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby
disclaims copyright to this source code.
 */
```

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Hashing.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/AbstractExecutionThreadService.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ForwardingFuture.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Splitter.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/ByteArrayDataInput.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/annotations/GwtIncompatible.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ForwardingListenableFuture.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/MapMakerInternalMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/Service.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/annotations/GwtCompatible.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/escape/ArrayBasedUnicodeEscaper.java
*

/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Cut.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/escape/Escapers.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/net/HostSpecifier.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/cache/LocalCache.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/LineProcessor.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/JdkFutureAdapters.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/AbstractIdleService.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/ByteArrayDataOutput.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/cache/CacheBuilder.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/AbstractService.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/SparseImmutableTable.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/RegularImmutableTable.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/reflect/TypeResolver.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/MapMaker.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/cache/ReferenceEntry.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/net/InternetDomainName.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/html/HtmlEscapers.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/DenseImmutableTable.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/net/UrlEscapers.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/Callables.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/escape/ArrayBasedEscaperMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/SettableFuture.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/SignedBytes.java

* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ForwardingFluentFuture.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/escape/ArrayBasedCharEscaper.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/ByteProcessor.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/xml/XmlEscapers.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/UnsignedBytes.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/escape/Platform.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Platform.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {@code Floats}. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/FloatsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0

*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
  * This method was written by Doug Lea with assistance from members of JCP JSR-166 Expert Group
  * and released to the public domain, as explained at
  * http://creativecommons.org/licenses/publicdomain
  *
  * As of 2010/06/11, this method is identical to the (package private) hash method in OpenJDK 7's
  * java.util.HashMap
 class.
  */

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/Striped.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors.
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*

* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for extra methods of {@code Objects} only in web. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/ExtraObjectsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {@code Doubles}. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/DoublesMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*

* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Preconditions.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/Resources.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/package-info.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/FinalizablePhantomReference.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Suppliers.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Charsets.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/LineReader.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/FinalizableSoftReference.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/CharStreams.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Predicate.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/ByteStreams.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/Primitives.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/LineBuffer.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/FinalizableWeakReference.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Predicates.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Functions.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/AbstractIterator.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/package-info.java

* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/HashBiMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/eventbus/DeadEvent.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/FinalizableReference.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/eventbus/Subscribe.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Objects.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Supplier.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/MultiInputStream.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Interners.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/EnumMultiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ExecutionList.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/package-info.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Defaults.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Function.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/FinalizableReferenceQueue.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/AbstractFuture.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/Closeables.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/eventbus/AllowConcurrentEvents.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/Files.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/eventbus/AsyncEventBus.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/LittleEndianDataOutputStream.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/eventbus/EventBus.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/CountingOutputStream.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-

jar/com/google/common/base/Throwables.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/util/concurrent/ListenableFuture.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/util/concurrent/DirectExecutor.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/io/CountingInputStream.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/io/LittleEndianDataInputStream.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/io/Flushables.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/eventbus/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/AbstractTable.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/reflect/TypeVisitor.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/base/VerifyException.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/util/concurrent/Runnables.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/base/Verify.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/hash/HashingInputStream.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/util/concurrent/WrappingScheduledExecutorService.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-

jar/com/google/common/eventbus/SubscriberExceptionContext.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/thirdparty/publicsuffix/PublicSuffixType.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Utf8.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/FilteredMultimapValues.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/CharSequenceReader.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/eventbus/SubscriberExceptionHandler.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Converter.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/net/PercentEscaper.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/thirdparty/publicsuffix/TrieParser.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/Booleans.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/internal/Finalizer.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/Doubles.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/CharMatcher.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/Floats.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-

jar/com/google/common/util/concurrent/SequentialExecutor.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/Bytes.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ListenableFutureTask.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/FluentIterable.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Stopwatch.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/FileBackedOutputStream.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/Shorts.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/Ints.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/Chars.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/MultiReader.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Joiner.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/escape/UnicodeEscaper.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/Longs.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/net/InetAddresses.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/escape/Escaper.java
No license file was found, but licenses were detected in source scan.

/*
* Written by Doug Lea with assistance from members of JCP JSR-166
* Expert Group and released to the public domain, as explained at
* http://creativecommons.org/publicdomain/zero/1.0/
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/LongAdder.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/cache/Striped64.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/cache/LongAdder.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/Striped64.java

* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/AtomicDoubleArray.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/MultimapBuilder.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the
* License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
* express or implied. See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableSortedMultisetFauxverideShim.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/RangeSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/SortedIterable.java
*

/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/SortedIterables.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/AbstractRangeSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableSortedMultiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/GeneralRange.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/RegularImmutableSortedMultiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingSortedMultiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Count.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
  * Returns an array containing all of the elements in the specified collection. This method
  * returns the elements in the order they are returned by the collection's iterator. The returned
  * array is "safe" in that no references to it are maintained by the collection. The caller is
  * thus free to modify the returned
 array.
  *
  * <p>This method assumes that the collection size doesn't change while the method is running.
  *
  * <p>TODO(kevinb): support concurrently modified collections?
  *
  * @param c the collection for which to return an array of elements
  */
```

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ObjectArrays.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
/**
  * Not supported. <b>You are attempting to create a map that may contain a non-{@code Comparable}
  * key.</b> Proper calls will resolve to the version in {@code ImmutableSortedMap}, not this dummy
  * version.
  *
  * @throws UnsupportedOperationException always
  * @deprecated <b>Pass a key of type {@code Comparable}
 to use {@link
  *     ImmutableSortedMap#of(Comparable, Object)}.</b>
  */
```

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableSortedMapFauxverideShim.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/GwtTransient.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright 2019 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/IgnoreJRERequirement.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2005 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/reflect/Reflection.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2016 The Guava Authors
```

*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/base/JdkPattern.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/base/PatternCompiler.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/base/CommonPattern.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/base/CommonMatcher.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/FilteredEntrySetMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/ForwardingImmutableSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/SortedMultisetBridge.java

*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/DescendingImmutableSortedSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/FilteredKeySetMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableEnumMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/FilteredEntryMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/DescendingMultiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/TreeRangeMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingNavigableSet.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/AllEqualOrdering.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/TreeTraverser.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/RegularImmutableAsList.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/CompactHashMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/TransformedIterator.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ForwardingBlockingDeque.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingDeque.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/AbstractSortedKeySortedSetMultimap.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingNavigableMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/TransformedListIterator.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingBlockingDeque.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingImmutableList.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/AbstractMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/CompactHashSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/EvictingQueue.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-

jar/com/google/common/collect/FilteredSetMultimap.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/CompactLinkedHashMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/CompactLinkedHashSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/AbstractNavigableMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/ForwardingImmutableMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/FilteredKeyListMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/UnmodifiableSortedMultiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/RangeMap.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/FilteredMultimap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/ImmutableEntry.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/thirdparty/publicsuffix/PublicSuffixPatterns.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/CollectPreconditions.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/ImmutableMapEntrySet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-

jar/com/google/common/collect/ImmutableMapKeySet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableCollection.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Table.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableSortedSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/UnmodifiableIterator.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableBiMap.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/TreeBasedTable.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/RegularImmutableBiMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Platform.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableListMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Tables.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Range.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/HashBasedTable.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableMultimap.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/RegularImmutableMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Collections2.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableMultiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/StandardTable.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Serialization.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/StandardRowSortedTable.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/PeekingIterator.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/EmptyImmutableListMultimap.java

* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableMapValues.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/ImmutableSupplier.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ExecutionSequencer.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2015 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/LittleEndianByteArray.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/InterruptibleTask.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/MacHashFunction.java

*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/ReaderInputStream.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/AggregateFutureState.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/Platform.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ConsumingQueueIterator.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/AsyncCallable.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/FarmHashFingerprint64.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/CombinedFuture.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Outer class that exists solely to let us write {@code Partially.GwtIncompatible} instead of plain
* {@code GwtIncompatible}. This is more accurate for {@link Futures#catching}, which is available
* under GWT but with a slightly different signature.
*
* <p>We can't use {@code PartiallyGwtIncompatible} because then the GWT compiler wouldn't recognize
* it as a {@code GwtIncompatible} annotation. And for {@code Futures.catching}, we need the GWT
* compiler to autostrip the normal server method in order to expose the special, inherited GWT
* version.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/Partially.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2012 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
```

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/reflect/ImmutableTypeToInstanceMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/reflect/MutableTypeToInstanceMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/CharSource.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/CartesianList.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableRangeMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/escape/package-info.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/LongAddable.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/BaseEncoding.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/ByteSource.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/math/PairedStatsAccumulator.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/reflect/AbstractInvocationHandler.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/reflect/ClassPath.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/math/StatsAccumulator.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/Closer.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/xml/package-info.java

* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/SmoothRateLimiter.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ListenableScheduledFuture.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/math/PairedStats.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/reflect/Invokable.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/reflect/TypeCapture.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/math/Stats.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/RateLimiter.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/reflect/Parameter.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/reflect/TypeToInstanceMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/reflect/Element.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/cache/LongAddables.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/FileWriteMode.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/LongAddables.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/ByteSink.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ServiceManager.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/StandardSystemProperty.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/CharSink.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/AbstractByteHasher.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/math/LinearTransformation.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/html/package-info.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/SipHashFunction.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/reflect/package-info.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-

jar/com/google/common/hash/ChecksumHashFunction.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableRangeSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/FilteredKeyMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/cache/LongAddable.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ServiceManagerBridge.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Java8Usage.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):

* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/math/BigDecimalMath.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/Java8Compatibility.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/OverflowAvoidingLockSupport.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/Java8Compatibility.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/math/ToDoubleRounder.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ObjectCountLinkedHashMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/Traverser.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/AbstractBaseGraph.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/BaseGraph.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ObjectCountHashMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ClosingFuture.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*

* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/cache/package-info.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/hash/AbstractHasher.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/util/concurrent/AbstractListeningExecutorService.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/primitives/UnsignedInts.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/primitives/UnsignedLongs.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/util/concurrent/ListeningScheduledExecutorService.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/hash/Crc32cHashFunction.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/cache/CacheLoader.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/cache/AbstractCache.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/math/package-info.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/util/concurrent/AsyncFunction.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/hash/AbstractNonStreamingHashFunction.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/primitives/ParseRequest.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/hash/Funnel.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/cache/RemovalNotification.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/TreeRangeSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-

jar/com/google/common/cache/Weigher.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/Uninterruptibles.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/net/HostAndPort.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/AbstractCompositeHashFunction.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/BloomFilterStrategies.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/HashCode.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ExecutionError.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/MessageDigestHashFunction.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/Murmur3_128HashFunction.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/math/IntMath.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/Hashing.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Optional.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/UncheckedExecutionException.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/math/LongMath.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/BoundType.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/UnsignedLong.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/BloomFilter.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/CycleDetectingLockFactory.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/cache/RemovalListeners.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Queues.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/math/BigIntegerMath.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/cache/CacheBuilderSpec.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/net/HttpHeaders.java

* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ForwardingExecutorService.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/DescendingImmutableSortedMultiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Ticker.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/HashingOutputStream.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/reflect/Types.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/Funnels.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/FutureCallback.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/cache/ForwardingLoadingCache.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/PrimitiveSink.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/HashFunction.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/math/DoubleMath.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/cache/RemovalListener.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/Hasher.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Enums.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/cache/LoadingCache.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Present.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/math/MathPreconditions.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/AbstractSortedMultiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/AbstractScheduledService.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/RegularContiguousSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/cache/CacheStats.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/AbstractStreamingHasher.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-

jar/com/google/common/cache/Cache.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/AtomicLongMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/cache/RemovalCause.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/WrappingExecutorService.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/PairwiseEquivalence.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ForwardingListeningExecutorService.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/cache/AbstractLoadingCache.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/Absent.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/cache/ForwardingCache.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/FunctionalEquivalence.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/Murmur3_32HashFunction.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/math/DoubleUtils.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/EmptyContiguousSet.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/reflect/TypeParameter.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/RegularImmutableMultiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/UnsignedInteger.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/net/MediaType.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express

* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/hash/AbstractHashFunction.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ForwardingLock.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ForwardingCondition.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/ImmutableDoubleArray.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/ImmutableIntArray.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/ImmutableLongArray.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/TopKSelector.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/SuccessorsFunction.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/Graph.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/PredecessorsFunction.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/ImmutableGraph.java

* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/Graphs.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/MutableGraph.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/Network.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/MutableNetwork.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/ImmutableNetwork.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingSetMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingSortedSetMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/RowSortedTable.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingImmutableCollection.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/AbstractSequentialIterator.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/UnmodifiableListIterator.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingListMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/MinMaxPriorityQueue.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/SortedMapDifference.java

No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/IndexedImmutableSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/BaseImmutableMultimap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not
* use this file except in compliance with the License. You may obtain a copy of
* the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/SortedMultiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/SortedMultisets.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2009 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/ImmutableClassToInstanceMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/RegularImmutableSortedSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/ArrayTable.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/ImmutableSortedMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/ForwardingTable.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/DiscreteDomain.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/RegularImmutableList.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/ImmutableSetMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/ImmutableAsList.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/SingletonImmutableTable.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/EmptyImmutableSetMultimap.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/ImmutableSortedSetFauxverideShim.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/ComputationException.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/ComparisonChain.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
```

jar/com/google/common/collect/AbstractIndexedListIterator.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableEnumSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableTable.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2019 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/Platform.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/AbstractNetwork.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/UndirectedMultiNetworkConnections.java

* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/LinkedHashMultimapGwtSerializationDependencies.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/UndirectedNetworkConnections.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/DirectedMultiNetworkConnections.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/MapIteratorCache.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/ForwardingNetwork.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/DirectedGraphConnections.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/StandardValueGraph.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/HashMultimapGwtSerializationDependencies.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/NetworkBuilder.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/EdgesConnecting.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableMultisetGwtSerializationDependencies.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/ElementOrder.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/StandardMutableGraph.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/GraphConstants.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/ImmutableValueGraph.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/AbstractGraphBuilder.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/NetworkConnections.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/MutableValueGraph.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/StandardMutableValueGraph.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/AbstractGraph.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/StandardNetwork.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/MapRetrievalCache.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-

jar/com/google/common/graph/AbstractDirectedNetworkConnections.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/graph/UndirectedGraphConnections.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/graph/ValueGraph.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/graph/GraphBuilder.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/graph/EndpointPairIterator.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/Comparators.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/graph/DirectedNetworkConnections.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/graph/ForwardingGraph.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/graph/StandardMutableNetwork.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/graph/GraphConnections.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/graph/EndpointPair.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/RangeGwtSerializationDependencies.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/collect/ArrayListMultimapGwtSerializationDependencies.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/graph/MultiEdgesConnecting.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/graph/AbstractUndirectedNetworkConnections.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/graph/ForwardingValueGraph.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/graph/AbstractValueGraph.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-
jar/com/google/common/graph/ValueGraphBuilder.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2006 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*

* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/TimeoutFuture.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/GwtFuturesCatchingSpecialization.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/annotations/VisibleForTesting.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/GwtFluentFutureCatchingSpecialization.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/FuturesGetChecked.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/SimpleTimeLimiter.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/FakeTimeLimiter.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/AppendableWriter.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/Futures.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/reflect/TypeToken.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/escape/CharEscaperBuilder.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/FluentFuture.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/AbstractTransformFuture.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/AbstractCatchingFuture.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/AggregateFuture.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/escape/CharEscaper.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/CollectionFuture.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/ImmediateFuture.java
*

/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/io/PatternFilenameFilter.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/TimeLimiter.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/util/concurrent/UncheckedTimeoutException.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/CaseFormat.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/*
 * This method was rewritten in Java from an intermediate step of the Murmur hash function in
 * http://code.google.com/p/smhasher/source/browse/trunk/MurmurHash3.cpp, which contained the
 * following header:
 *
 * MurmurHash3 was written by Austin Appleby, and is placed in the public domain. The author
 * hereby disclaims
copyright to this source code.
 */

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/base/SmallCharMatcher.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License

* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {@code Shorts}. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/ShortsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2019 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/graph/IncidentEdgeSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/CompactHashing.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except
* in compliance with the License. You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software distributed under the License
* is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

express
* or implied. See the License for the specific language governing permissions and limitations under
* the License.
*/
/**
* Holder for web specializations of methods of {@code Ints}. Intended to be empty for regular
* version.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/primitives/IntsMethodsForWeb.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/EnumBiMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingSortedSet.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableList.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingCollection.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/AbstractSetMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Multisets.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Ordering.java

* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/NullsLastOrdering.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ImmutableSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingMapEntry.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ClassToInstanceMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/AbstractMapEntry.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/SingletonImmutableSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ListMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/SortedSetMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/TreeMultiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/LinkedListMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Interner.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/UsingToStringOrdering.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/AbstractIterator.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Maps.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingIterator.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ComparatorOrdering.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingSortedMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/SetMultimap.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Lists.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingMultiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-

jar/com/google/common/collect/Multiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/MutableClassToInstanceMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/RegularImmutableSet.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/package-info.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingQueue.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/AbstractBiMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/AbstractMapBasedMultiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ExplicitOrdering.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Iterators.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/LinkedHashMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ConcurrentHashMultiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Multimap.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/LexicographicalOrdering.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingConcurrentMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingObject.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/BiMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Iterables.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/AbstractSortedSetMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/MapDifference.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/AbstractListMultimap.java
*
/opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/HashMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Multimaps.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ReverseNaturalOrdering.java

* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/AbstractMultiset.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Sets.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingListIterator.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/Synchronized.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/CompoundOrdering.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/NaturalOrdering.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ReverseOrdering.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ArrayListMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ByFunctionOrdering.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/NullsFirstOrdering.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/ForwardingList.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/TreeMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/HashMultiset.java
*
 /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/AbstractMapBasedMultimap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/EnumHashBiMap.java
* /opt/cola/permits/1148655839_1618904031.03/0/guava-30-1-1-android-sources-jar/com/google/common/collect/LinkedHashMultiset.java

1.8 json-smart 2.3

1.8.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
*   Copyright 2011-2014 JSON-SMART authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*

* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/writer/DefaultMapper.java
No license file was found, but licenses were detected in source scan.

/*
*   Copyright 2011 JSON-SMART authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/JSONStyle.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/parser/JSONParserInputStream.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/parser/JSONParser.java
*
 /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/parser/JSONParserMemory.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/writer/CompessorMapper.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/JSONUtil.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/JSONStreamAwareEx.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/parser/ParseException.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/JSONValue.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-

jar/net/minidev/json/writer/FakeMapper.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/JSONAwareEx.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/writer/BeansMapper.java
*
/opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/parser/JSONParserByteArray.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/writer/JsonReaderI.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/writer/CollectionMapper.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/JSONStreamAware.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/JSONNavi.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/parser/JSONParserStream.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/JSONAware.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/writer/DefaultMapperOrdered.java
*
/opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/JSONObject.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/writer/JsonReader.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/JSONArray.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/JStylerObj.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/writer/DefaultMapperCollection.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/parser/JSONParserString.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/writer/ArraysMapper.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/parser/JSONParserBase.java
* /opt/cola/permits/1209129301_1633010713.35/0/json-smart-2-3-sources-6-jar/net/minidev/json/parser/JSONParserReader.java

1.9 libcxx 9.0.9svn

1.9.1 Available under license :

==
libc++ License
==

The libc++ library is dual licensed under both the University of Illinois
"BSD-Like" license and the MIT license. As a user of this code you may choose
to use it under either license. As a contributor, you agree to allow your code

to be used under both.

Full text of the relevant licenses is included below.

==

University of Illinois/NCSA
Open Source License

Copyright (c) 2009-2017 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

http://llvm.org

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal with
the Software without restriction, including
 without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimers in the
  documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at
  Urbana-Champaign, nor the names of its contributors may be used to
  endorse or promote products derived from this Software without specific
  prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE

SOFTWARE.

==========

Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial
portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

People who have agreed to one of the CLAs and can contribute patches.
The AUTHORS file lists the copyright holders; this file
lists people. For example, Google employees are listed here
but not in AUTHORS, because Google holds the copyright.
#
Names should be added to this file only after verifying that

```
# the individual or the individual's organization has agreed to
# the appropriate Contributor License Agreement, found here:
#
# https://developers.google.com/open-source/cla/individual
# https://developers.google.com/open-source/cla/corporate
#
# The agreement for individuals can be filled out on the web.
#
# When adding J Random Contributor's name to this file,
# either J's name or J's organization's name should be
# added to the AUTHORS file, depending on whether the
# individual or corporate CLA was used.
#
# Names should be added to this file as:
#     Name <email address>
#
# Please keep the list sorted.

Albert Pretorius <pretoalb@gmail.com>
Arne Beer <arne@twobeer.de>
Billy
 Robert O'Neal III <billy.oneal@gmail.com> <bion@microsoft.com>
Chris Kennelly <ckennelly@google.com> <ckennelly@ckennelly.com>
Christopher Seymour <chris.j.seymour@hotmail.com>
Cyrille Faucheux <cyrille.faucheux@gmail.com>
David Coeurjolly <david.coeurjolly@liris.cnrs.fr>
Deniz Evrenci <denizevrenci@gmail.com>
Dominic Hamon <dma@stripysock.com> <dominic@google.com>
Dominik Czarnota <dominik.b.czarnota@gmail.com>
Eric Fiselier <eric@efcs.ca>
Eugene Zhuk <eugene.zhuk@gmail.com>
Evgeny Safronov <division494@gmail.com>
Federico Ficarelli <federico.ficarelli@gmail.com>
Felix Homann <linuxaudio@showlabor.de>
Ismael Jimenez Martinez <ismael.jimenez.martinez@gmail.com>
Jern-Kuan Leong <jernkuan@gmail.com>
JianXiong Zhou <zhoujianxiong2@gmail.com>
Joao Paulo Magalhaes <joaoppmagalhaes@gmail.com>
John Millikin <jmillikin@stripe.com>
Jussi Knuuttila <jussi.knuuttila@gmail.com>
Kai Wolf <kai.wolf@gmail.com>
Kishan Kumar <kumar.kishan@outlook.com>
Kaito Udagawa <umireon@gmail.com>
Lei Xu <eddyxu@gmail.com>
Matt
 Clarkson <mattyclarkson@gmail.com>
Maxim Vafin <maxvafin@gmail.com>
Nick Hutchinson <nshutchinson@gmail.com>
```

Oleksandr Sochka <sasha.sochka@gmail.com>
Ori Livneh <ori.livneh@gmail.com>
Pascal Leroy <phl@google.com>
Paul Redmond <paul.redmond@gmail.com>
Pierre Phaneuf <pphaneuf@google.com>
Radoslav Yovchev <radoslav.tm@gmail.com>
Raul Marin <rmrodriguez@cartodb.com>
Ray Glover <ray.glover@uk.ibm.com>
Robert Guo <robert.guo@mongodb.com>
Roman Lebedev <lebedev.ri@gmail.com>
Shuo Chen <chenshuo@chenshuo.com>
Tobias Ulvgrd <tobias.ulvgard@dirac.se>
Tom Madams <tom.ej.madams@gmail.com> <tmadams@google.com>
Yixuan Qiu <yixuanq@gmail.com>
Yusuke Suzuki <utatane.tea@gmail.com>
Zbigniew Skowron <zbychs@gmail.com>

1.10 libcxxabi 9.0.9svn

1.10.1 Available under license :

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction,
including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

==========

Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial
portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

==========

libc++abi License

==========

The libc++abi library is dual licensed under both the University of Illinois "BSD-Like" license and the MIT license. As a user of this code you may choose to use it under either license. As a contributor, you agree to allow your code to be used under both.

Full text of the relevant licenses is included below.

==========

University of Illinois/NCSA
Open Source License

Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

http://llvm.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction,
including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at

Urbana-Champaign, nor the names of its contributors may be used to
endorse or promote products derived from this Software without specific
prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

==========

Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial
portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.11 jackson-databind 2.13.1

1.11.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers.

Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0
To find the details that apply to this artifact see the accompanying LICENSE file.

Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

1.12 jackson 2.13.1

1.12.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and
limitations under the License.
Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers.

Licensing

Jackson 2.x core and extension components are licensed under Apache License 2.0
To find the details that apply to this artifact see the accompanying LICENSE file.

Credits

A list of contributors may be found from CREDITS(-2.x) file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

1.13 constraintlayout 2.0.1

1.13.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* Copyright 2019 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/state/helpers/GuidelineReference.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/state/Reference.java
*

/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/state/helpers/BarrierReference.java
No license file was found, but licenses were detected in source scan.

```
apply plugin: 'java'
apply plugin: 'maven'

compileJava {
  targetCompatibility = 1.7
  sourceCompatibility = 1.7
}

dependencies {
  testCompile 'org.testng:testng:6.9.10'
}

test {
  useTestNG()

  testLogging.showStandardStreams = true
  beforeTest { descriptor ->
     logger.lifecycle("Running test: " + descriptor)
  }
  onOutput { descriptor, event ->
     logger.lifecycle("Test: " + descriptor + " output: " + event.message )
  }
}

archivesBaseName = 'constraintlayout-solver'

project.ext.pomName = 'Android ConstraintLayout Solver'
project.ext.pomDesc = 'Solver for ConstraintLayout'

task publishLocal(type: Upload) {
  configuration = configurations.archives
  repositories {
     mavenDeployer {
        repository(url: uri("$rootProject.ext.localRepo"))
        pom.project {
           name project.ext.pomName
           description project.ext.pomDesc


      url 'http://tools.android.com'
           inceptionYear '2007'

           licenses {
              license {
```

```
                name 'The Apache Software License, Version 2.0'
                url 'http://www.apache.org/licenses/LICENSE-2.0.txt'
                distribution 'repo'
            }
        }

        scm {
            url 'https://android.googlesource.com/platform/tools/sherpa'
            connection 'git://android.googlesource.com/platform/tools/sherpa.git'
        }
        developers {
            developer {
                name 'The Android Open Source Project'
            }
        }
    }
  }
}
```

Found in path(s):
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/build.gradle
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2017 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/widget/Placeholder.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/widget/Barrier.java
*

/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/Chain.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/GoalRow.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/utils/HyperSpline.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/widget/Constraints.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/CenterWrapTest.java
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/CircleTest.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/widget/ConstraintAttribute.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/Barrier.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/Guideline.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/OptimizedGoal.java
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/EquationVariableTest.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/ArrayRow.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/ArrayBackedVariables.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-

2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/Amount.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-
2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/OptimizationsTest.java
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-
2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/OriginalGoal.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-
2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/EquationVariable.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-
2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/LinearSystem.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-
2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/LinearSystemTest.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-
2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/Pools.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-
2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/SolverVariable.java
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-
2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/widget/ConstraintLayout.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-
2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/WidgetsPositioningTest.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-
2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/AmountTest.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-
2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/ConstraintWidgetContainer.jav
a
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-
2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/ConstraintAnchor.java
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-
2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/LinearEquation.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-
2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/ConstraintWidget.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-
2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/widget/Guideline.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/ChainTest.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/scout/ScoutWidget.java
*
 /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/WidgetContainer.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/VisibilityTest.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/scout/Direction.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/XmlBasedTest.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/Cache.java
*
 /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/ChainWrapContentTest.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/AdvancedChainTest.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/MatchConstraintTest.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/HistogramCounter.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/scout/Utils.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/Rectangle.java
*
 /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/RandomLayoutTest.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/widget/ConstraintSet.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/scout/ScoutProbabilities.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/scout/Scout.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/LayoutTest.java
*
 /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/RatioTest.java

* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/ArrayLinkedVariables.java
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2019 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/analyzer/HorizontalWidgetRun.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/analyzer/VerticalWidgetRun.java
*
 /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/VirtualLayout.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/KeyCache.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/analyzer/BaselineDimensionDependency.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/analyzer/Dependency.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/analyzer/DependencyGraph.java
*
 /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/analyzer/DependencyNode.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/MotionInterpolator.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/state/helpers/VerticalChainReference.ja

va
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/helper/widget/Flow.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/state/helpers/ChainReference.java
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/Flow.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/analyzer/WidgetRun.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/analyzer/GuidelineReference.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/analyzer/ChainRun.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/state/helpers/AlignHorizontallyReference.java
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/state/ConstraintReference.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/state/helpers/HorizontalChainReference.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/TransitionBuilder.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/analyzer/HelperReferences.java
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/analyzer/RunGroup.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/state/HelperReference.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/analyzer/DimensionDependency.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/state/helpers/AlignVerticallyReference.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/state/Dimension.java
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/analyzer/BasicMeasure.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/state/State.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2020 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/SolverVariableValues.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/PriorityGoalRow.java
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/SolverVariableValuesTest.java
No license file was found, but licenses were detected in source scan.

Copyright (C) 2011 The Android Open Source Project

```
   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE
2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.
```

Found in path(s):
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/resources/check06.xml
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/resources/check00.xml

*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/resources/check03.xml
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/resources/check13.xml
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/resources/check01.xml
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/resources/check10.xml
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/resources/check05.xml
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/resources/check16.xml
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/resources/check08.xml
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/resources/check09.xml
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/resources/check12.xml
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/resources/check07.xml
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/resources/check04.xml
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/resources/check02.xml
No license file was found, but licenses were detected in source scan.

url 'http://www.apache.org/licenses/LICENSE-2.0.txt'

Found in path(s):
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/build.gradle
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/build.gradle
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/widget/ConstraintsChangedListener.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/utils/widget/MotionTelltales.java
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/utils/Oscillator.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/Optimizer.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/utils/MonotonicCurveFit.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/Key.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/Helper.java
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/MotionConstrainedPoint.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/KeyPosition.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/test/java/android/support/constraint/solver/widgets/ChainHeadTest.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/MotionPaths.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/widget/VirtualLayout.java
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/utils/widget/ImageFilterButton.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/widgets/ChainHead.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/utils/VelocityMatrix.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/utils/ArcCurveFit.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/widget/StateSet.java
*

/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/MotionLayout.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/utils/Easing.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/DesignTool.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/KeyAttributes.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/SplineSet.java
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/Debug.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/widget/ConstraintLayoutStates.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/widget/ConstraintProperties.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/TouchResponse.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/KeyPositionBase.java
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/utils/CurveFit.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/KeyTimeCycle.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/solver/src/main/java/androidx/constraintlayout/solver/Metrics.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/KeyTrigger.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/MotionScene.java
*
/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/utils/LinearCurveFit.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/MotionController.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/utils/widget/ImageFilterView.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/CustomFloatAttributes.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/KeyCycle.java
*

/opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/utils/StopLogic.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/TimeCycleSplineSet.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/KeyFrames.java
* /opt/cola/permits/1270630270_1645057508.42/0/constraintlayout-2-0-1-zip/constraintlayout-2.0.1/constraintlayout/constraintlayout/src/main/java/androidx/constraintlayout/motion/widget/KeyCycleOscillator.java

1.14 okio 3.0.0

1.14.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2015 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/ForwardingTimeout.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/SegmentedByteString.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/SegmentedByteString.kt
*
/opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/SegmentSharingTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/SegmentedByteString.kt
No license file was found, but licenses were detected in source scan.

```
/*
*  Licensed to the Apache Software Foundation (ASF) under one or more
*  contributor license agreements.  See the NOTICE file distributed with
*  this work for additional information regarding copyright ownership.
```

* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/-Base64.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2021 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/JvmFileHandle.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmTest/okio/utilNonJvm.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-
jar/mingwX64Main/okio/WindowsFileHandle.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/FixedLengthSourceTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/TimeoutTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-
jar/jvmMain/okio/internal/FixedLengthSource.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/ZipFileSystemTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/unixMain/okio/UnixFileHandle.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nativeMain/okio/FileSystem.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/ZipBuilder.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-

jar/jvmTest/okio/internal/ResourceFileSystemTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-
jar/jvmMain/okio/internal/ResourceFileSystem.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/FileHandle.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/FileSystem.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/utilJvm.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-
jar/jvmTest/okio/FileHandleTestingFileSystem.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/internal/-
FileSystem.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-
jar/jvmTest/okio/ForwardingTimeoutKotlinTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/GzipKotlinTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/ByteStringKotlinTest.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/-DeprecatedUtf8.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/-DeprecatedUpgrade.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/BufferCursorTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/PipeKotlinTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/NioTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/BufferKotlinTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/ThrottlerTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/ThrottlerTakeTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/DeflateKotlinTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/Throttler.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/BufferCursorKotlinTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/PeekSource.kt

* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonTest/okio/CommonOptionsTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/ForwardingTimeoutTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/OkioKotlinTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/-DeprecatedOkio.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/Stopwatch.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonTest/okio/Utf8KotlinTest.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/RealBufferedSource.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/GzipSink.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/RealBufferedSource.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/BufferedSinkTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/TestUtil.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/AsyncTimeout.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/BufferedSink.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/GzipSource.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/OkioTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/BufferedSource.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/Timeout.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/SocketTimeoutTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/Buffer.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/BufferedSinkJavaTest.java
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonTest/okio/MockSink.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/GzipSinkTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/ForwardingSource.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/DeflaterSinkTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/DeflaterSink.kt

* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/AsyncTimeoutTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/ReadUtf8LineTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/ForwardingSink.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/SegmentPool.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/GzipSourceTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/InflaterSource.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/JvmOkio.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/SegmentPool.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/BufferedSourceTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonTest/okio/CommonBufferTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-
jar/jvmTest/okio/BufferedSourceJavaTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/Sink.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/SegmentPool.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/Utf8Test.java
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/InflaterSourceTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/BufferTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/RealBufferedSink.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/Segment.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright 2014 Square Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/ByteString.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/ByteStringJavaTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonTest/okio/HashingTest.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2018 Square, Inc.
*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/internal/-ByteString.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/-CommonPlatform.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonTest/okio/ByteStringFactory.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/internal/-Utf8.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/-JvmPlatform.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonTest/okio/ByteStringTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/ByteString.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/ByteString.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/-Util.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/-NonJvmPlatform.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2019 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/internal/-RealBufferedSource.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/internal/-

RealBufferedSink.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/internal/-SegmentedByteString.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements.  See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License.  You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/internal/zip.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/internal/ZipEntry.kt
*
/opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/ZipFileSystem.kt
No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2016 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/WaitUntilNotifiedTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/HashingSink.kt

* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonTest/okio/HashingSourceTest.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/LargeStreamsTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/Pipe.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/HashingSource.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonTest/okio/HashingSinkTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/Options.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/PipeTest.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nativeMain/okio/-SizetVariant.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2019 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/BufferFactory.kt

* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/Buffer.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/Okio.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/Sink.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/Source.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/Sink.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/RealBufferedSink.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-
jar/commonMain/okio/RealBufferedSource.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/internal/-Buffer.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-
jar/commonTest/okio/CommonRealBufferedSinkTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-
jar/commonTest/okio/AbstractBufferedSourceTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/BufferedSink.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonTest/okio/BufferCommonTest.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/BufferedSource.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/BufferedSink.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-
jar/commonTest/okio/AbstractBufferedSinkTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/RealBufferedSink.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-
jar/commonTest/okio/CommonRealBufferedSourceTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-
jar/commonTest/okio/CommonOkioKotlinTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/Timeout.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-
jar/commonTest/okio/BufferedSinkFactory.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/Timeout.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonTest/okio/util.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/BufferedSource.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-
jar/commonTest/okio/BufferedSourceFactory.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/Buffer.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/internal/Sha512.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/internal/Md5.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/MessageDigestConsistencyTest.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/internal/Sha256.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/internal/Md5.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/internal/Sha1.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/internal/Sha512.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/internal/Sha256.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/internal/Sha1.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/Utf8.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 Square, Inc. and others.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/internal/HashFunction.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/internal/HashFunction.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/CipherSink.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/CipherAlgorithm.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/CipherSinkTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/CipherSourceTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/CipherSource.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/CipherFactory.kt
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonTest/okio/FakeFileSystemTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/HashingSource.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/mingwX64Main/okio/-Windows.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonTest/okio/PathTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/HashingSink.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/FileMetadata.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonTest/okio/ForwardingFileSystemTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/Path.kt

* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/sizet64Main/okio/-Sizet64.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jsMain/okio/-JsPlatform.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/ExperimentalFileSystem.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/sizet32Main/okio/-Sizet32.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nativeMain/okio/-Cinterop.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/internal/-Path.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jsMain/okio/FileSystem.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nativeMain/okio/-PosixVariant.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/ForwardingFileSystem.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nativeTest/okio/NativeSystemFileSystemTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/internal/Hmac.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/Path.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/FileSystemJavaTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/unixMain/okio/-UnixPosixVariant.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nativeMain/okio/FileSource.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/HashingSink.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/linuxX64Main/okio/-LinuxX64PosixVariant.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonTest/okio/UnsafeCursorTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/appleMain/okio/-ApplePosixVariant.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nativeMain/okio/PosixFileSystem.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nonJvmMain/okio/HashingSource.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/internal/Hmac.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/nativeMain/okio/FileSink.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/JvmTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/appleMain/okio/ByteString.kt
*
 /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/Path.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/mingwX64Main/okio/-WindowsPosixVariant.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/internal/HmacTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/appleTest/okio/AppleByteStringTest.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/NioSystemFileSystem.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/ZipFileSystemJavaTest.java
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmMain/okio/JvmSystemFileSystem.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/commonMain/okio/FileSystem.kt
* /opt/cola/permits/1310287373_1650320542.41/0/okio-3-0-0-sources-jar/jvmTest/okio/JvmSystemFileSystemTest.kt

1.15 gson 2.8.9

1.15.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2018 The Gson authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/GsonBuildConfig.java

No license file was found, but licenses were detected in source scan.

```
/*
* Copyright (C) 2011 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Found in path(s):
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/TreeTypeAdapter.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/DateTypeAdapter.java

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/ConstructorConstructor.java
*
 /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/sql/SqlDateTypeAdapter.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/UnsafeAllocator.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/sql/SqlTimeTypeAdapter.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/LazilyParsedNumber.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/ArrayTypeAdapter.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/JsonTreeReader.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/TypeAdapterRuntimeTypeWrapper.java
*
 /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/CollectionTypeAdapterFactory.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/MapTypeAdapterFactory.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/TypeAdapterFactory.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/JsonReaderInternalAccess.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/ReflectiveTypeAdapterFactory.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-

jar/com/google/gson/internal/bind/TypeAdapters.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/ObjectTypeAdapter.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/TypeAdapter.java
*
/opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/JsonTreeWriter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonElement.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonObject.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonArray.java
*
/opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/FieldNamingStrategy.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/annotations/SerializedName.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonDeserializationContext.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/Excluder.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/FieldNamingPolicy.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/$Gson$Preconditions.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/DefaultDateTypeAdapter.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonSerializationContext.java

* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonParseException.java
*
 /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/ObjectConstructor.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/Gson.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonIOException.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/reflect/TypeToken.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonDeserializer.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/annotations/Expose.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/Primitives.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/GsonBuilder.java
*
 /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonSerializer.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/annotations/Since.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonPrimitive.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/ExclusionStrategy.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/annotations/Until.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/InstanceCreator.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonNull.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.

*/

Found in path(s):
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/JsonAdapterAnnotationTypeAdapterFactory.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/annotations/JsonAdapter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 The Android Open Source Project
* Copyright (C) 2012 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/LinkedHashTreeMap.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/LinkedTreeMap.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/stream/JsonReader.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/stream/MalformedJsonException.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/stream/JsonScope.java
*
/opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/stream/JsonToken.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/stream/JsonWriter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 The Gson authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/reflect/PreJava9ReflectionAccessor.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/reflect/ReflectionAccessor.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/reflect/UnsafeReflectionAccessor.java
*
/opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/PreJava9DateFormatProvider.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/JavaVersion.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 Google Inc.
*

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/bind/NumberTypeAdapter.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2021 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/ToNumberPolicy.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/ToNumberStrategy.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*

* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonSyntaxException.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/Streams.java
No license file was found, but licenses were detected in source scan.

/**
* Copyright (C) 2008 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/internal/$Gson$Types.java
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 Google Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Found in path(s):
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonStreamParser.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/LongSerializationPolicy.java
* /opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/FieldAttributes.java
*
/opt/cola/permits/1330613678_1652979131.328877/0/gson-2-8-9-sources-2-jar/com/google/gson/JsonParser.java

1.16 okhttp 4.10.0

1.16.1 Available under license :

Note that publicsuffixes.gz is compiled from The Public Suffix List:
https://publicsuffix.org/list/public_suffix_list.dat

It is subject to the terms of the Mozilla Public License, v. 2.0:
https://mozilla.org/MPL/2.0/
/*
* Copyright (C) 2016 Square, Inc.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.17 protobuf-java 3.17.2

1.17.1 Available under license :

No license file was found, but licenses were detected in source scan.

// Copyright 2008 Google Inc. All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
//   * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
//   * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
//   * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.

Found in path(s):
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/GeneratedMessageLite.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/RopeByteString.java
*
 /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/DynamicMessage.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/CodedInputStreamReader.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/GeneratedMessageInfoFactory.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ExperimentalApi.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/Internal.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/google/protobuf/field_mask.proto
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MessageOrBuilder.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-

jar/com/google/protobuf/NewInstanceSchemaLite.java
*
/opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/google/protobuf/descriptor.proto
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/CodedOutputStream.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MessageInfo.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/google/protobuf/empty.proto
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/StructuralMessageInfo.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/GeneratedMessageV3.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/google/protobuf/duration.proto
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MapEntry.java
*
/opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/LazyStringList.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ExtensionSchemas.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/BinaryWriter.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MessageInfoFactory.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/InvalidProtocolBufferException.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/TextFormatParseInfoTree.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MapFieldSchema.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/DescriptorMessageInfoFactory.java
*
/opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/LazyStringArrayList.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/Message.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/FieldType.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/UnknownFieldSetLiteSchema.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MessageSchema.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/FieldSet.java

* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/Extension.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/google/protobuf/any.proto
*
 /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ListFieldSchema.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ExtensionSchemaLite.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/JavaType.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/RawMessageInfo.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/RepeatedFieldBuilder.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/SmallSortedMap.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/google/protobuf/api.proto
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/google/protobuf/wrappers.proto
*
 /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ArrayDecoders.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/Android.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ProtocolMessageEnum.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/Schema.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MapField.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/UnknownFieldSetLite.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/TextFormatParseLocation.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/google/protobuf/source_context.proto
*
 /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/RpcUtil.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ProtobufArrayList.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/CodedOutputStreamWriter.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MessageSetSchema.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-

jar/com/google/protobuf/ManifestSchemaFactory.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/RpcCallback.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/AbstractMessage.java
*
/opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MapFieldSchemaFull.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/LongArrayList.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/GeneratedMessage.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MapEntryLite.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/AllocatedBuffer.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/UnknownFieldSetSchema.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/RpcController.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/LazyFieldLite.java
*
/opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/UnmodifiableLazyStringList.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/SchemaFactory.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/UnknownFieldSchema.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ByteString.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MutabilityOracle.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/BooleanArrayList.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/AbstractParser.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/BlockingService.java
*
/opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/UninitializedMessageException.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/FieldInfo.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MapFieldSchemas.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/UnsafeByteOperations.java

* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ExtensionRegistry.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/AbstractMessageLite.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/UnknownFieldSet.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ExtensionRegistryFactory.java
*
/opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ProtocolStringList.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/BinaryReader.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/Writer.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/google/protobuf/compiler/plugin.proto
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/SingleFieldBuilder.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MapFieldSchemaLite.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/NewInstanceSchema.java
*
/opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/OneofInfo.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/RpcChannel.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/TextFormatEscaper.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/BlockingRpcChannel.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/IterableByteBufferInputStream.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MessageLiteOrBuilder.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/UnsafeUtil.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/DoubleArrayList.java
*
/opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/Reader.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ByteBufferWriter.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ExtensionLite.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-

jar/com/google/protobuf/SingleFieldBuilderV3.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/Protobuf.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/google/protobuf/timestamp.proto
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ProtobufLists.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/Utf8.java
*
/opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ServiceException.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/RepeatedFieldBuilderV3.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/NewInstanceSchemaFull.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/NewInstanceSchemas.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/IntArrayList.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/Service.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/Parser.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/AbstractProtobufList.java
*
/opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ExtensionSchemaFull.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ProtoSyntax.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/google/protobuf/struct.proto
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/CodedInputStream.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MessageLiteToString.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MessageReflection.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ExtensionSchema.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MessageLite.java
*
/opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/BufferAllocator.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ExtensionRegistryLite.java

* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/TextFormat.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/NioByteString.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/DiscardUnknownFieldsParser.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/SchemaUtil.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/PrimitiveNonBoxingCollection.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/TypeRegistry.java
*
/opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/ByteOutput.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/LazyField.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/Descriptors.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/MapFieldLite.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/FloatArrayList.java
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/google/protobuf/type.proto
* /opt/cola/permits/1444330749_1666012919.2583277/0/protobuf-java-3-17-2-sources-1-jar/com/google/protobuf/WireFormat.java

1.18 safestring 2.0

1.18.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*--
* strstr_s.c
*
* November 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All rights reserved.
*--
*/

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strstr_s.c
No license file was found, but licenses were detected in source scan.

```
/*------------------------------------------------------------------
* safe_string.h
*
* November 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
*
* Permission is hereby granted, free of charge, to any person
* obtaining a copy of this software and associated documentation
* files (the "Software"), to deal in the Software without restriction,
* including without limitation the rights to use, copy, modify, merge,
* publish, distribute, sublicense, and/or sell copies of the Software,
* and to permit persons to whom the Software is furnished to do so,
* subject to the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
* OF MERCHANTABILITY,
 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*------------------------------------------------------------------
*/
```

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/include/safe_string.h
No license file was found, but licenses were detected in source scan.

```
/*------------------------------------------------------------------
* mem_primitives_lib.c - Unguarded Memory Copy Routines
*
* February 2005, Bo Berry
*
* Copyright (c) 2005-2010, 2013 by Cisco Systems, Inc.
*
* Permission is hereby granted, free of charge, to any person
* obtaining a copy of this software and associated documentation
* files (the "Software"), to deal in the Software without
* restriction, including without limitation the rights to use,
* copy, modify, merge, publish, distribute, sublicense, and/or
* sell copies of the Software, and to permit persons to whom the
* Software is furnished to do so, subject to the following
* conditions:
```

```
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
 TO THE WARRANTIES
* OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT.  IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
* HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
* WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
* FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
* OTHER DEALINGS IN THE SOFTWARE.
*----------------------------------------------------------------
*/
```

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/mem_primitives_lib.c
No license file was found, but licenses were detected in source scan.

```
/*----------------------------------------------------------------
* strcat_s.c
*
* October 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All rights Reserved.
*/
```

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strcat_s.c
No license file was found, but licenses were detected in source scan.

```
/*----------------------------------------------------------------
* strpbrk_s.c
*
* November 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All rights Reserved.
*----------------------------------------------------------------
*/
```

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strpbrk_s.c

No license file was found, but licenses were detected in source scan.

```
/*------------------------------------------------------------------
* safe_str_constraint.h
*
* October 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
*
* Permission is hereby granted, free of charge, to any person
* obtaining a copy of this software and associated documentation
* files (the "Software"), to deal in the Software without restriction,
* including without limitation the rights to use, copy, modify, merge,
* publish, distribute, sublicense, and/or sell copies of the Software,
* and to permit persons to whom the Software is furnished to do so,
* subject to the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
* OF MERCHANTABILITY,
 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*------------------------------------------------------------------
*/
```

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/safe_str_constraint.h
No license file was found, but licenses were detected in source scan.

```
/*------------------------------------------------------------------
* strisuppercase_s.c
*
* October 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All rights reserved.
*------------------------------------------------------------------
*/
```

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-

gz/safe_lib_2_0/src/safe_string/strisuppercase_s.c
No license file was found, but licenses were detected in source scan.

```
/*------------------------------------------------------------------
* mem_primitives_lib.h - Unguarded Memory Copy Routines
*
* October 2005, Bo Berry
*
* Copyright (c) 2005-2010, 2013 by Cisco Systems, Inc.
*
* Permission is hereby granted, free of charge, to any person
* obtaining a copy of this software and associated documentation
* files (the "Software"), to deal in the Software without restriction,
* including without limitation the rights to use, copy, modify, merge,
* publish, distribute, sublicense, and/or sell copies of the Software,
* and to permit persons to whom the Software is furnished to do so,
* subject to the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
 WARRANTIES
* OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*------------------------------------------------------------------
*/
```

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/include/mem_primitives_lib.h
No license file was found, but licenses were detected in source scan.

```
/*------------------------------------------------------------------
* safe_mem_constraint.h
*
* October 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
*
* Permission is hereby granted, free of charge, to any person
* obtaining a copy of this software and associated documentation
* files (the "Software"), to deal in the Software without restriction,
* including without limitation the rights to use, copy, modify, merge,
```

* publish, distribute, sublicense, and/or sell copies of the Software,
* and to permit persons to whom the Software is furnished to do so,
* subject to the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
* OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*--
*/

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/safe_mem_constraint.h
No license file was found, but licenses were detected in source scan.

/*--
* safe_limits.h -- limits include for portability
*
* February 2009, Bo Berry
*
* Copyright (c) 2009-2010, 2013 by Cisco Systems, Inc.
*
* Permission is hereby granted, free of charge, to any person
* obtaining a copy of this software and associated documentation
* files (the "Software"), to deal in the Software without restriction,
* including without limitation the rights to use, copy, modify, merge,
* publish, distribute, sublicense, and/or sell copies of the Software,
* and to permit persons to whom the Software is furnished to do so,
* subject to the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES
* OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE

* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*--
*/

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/include/safe_limits.h
No license file was found, but licenses were detected in source scan.

/*--
* safe_types.h -- types include for portability
*
* March 2007, Bo Berry
*
* Copyright (c) 2007-2010, 2013 by Cisco Systems, Inc.
*
* Permission is hereby granted, free of charge, to any person
* obtaining a copy of this software and associated documentation
* files (the "Software"), to deal in the Software without restriction,
* including without limitation the rights to use, copy, modify, merge,
* publish, distribute, sublicense, and/or sell copies of the Software,
* and to permit persons to whom the Software is furnished to do so,
* subject to the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
 * OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*--
*/

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/include/safe_types.h
No license file was found, but licenses were detected in source scan.

/*--
* strzero_s.c
*
* October 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All Rights Reserved.
*

*--
*/

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strzero_s.c
No license file was found, but licenses were detected in source scan.

/*--
* strncat_s.c
*
* October 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All Rights reserved
*--
*/

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strncat_s.c
No license file was found, but licenses were detected in source scan.

/*--
* strcasestr_s.c
*
* November 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2012-2013 by Cisco Systems, Inc.
* All rights Reserved.
*/

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strcasestr_s.c
No license file was found, but licenses were detected in source scan.

/*--
* safe_lib.h -- library include for portability
*
* February 2009, Bo Berry
*
* Copyright (c) 2009-2010, 2013 by Cisco Systems, Inc.
*
* Permission is hereby granted, free of charge, to any person
* obtaining a copy of this software and associated documentation
* files (the "Software"), to deal in the Software without restriction,
* including without limitation the rights to use, copy, modify, merge,

* publish, distribute, sublicense, and/or sell copies of the Software,
* and to permit persons to whom the Software is furnished to do so,
* subject to the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
 * OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*--
*/

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/include/safe_lib.h
No license file was found, but licenses were detected in source scan.

/*--
* safe_str_lib.h -- Safe C Library String APIs
*
* October 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
*
* Permission is hereby granted, free of charge, to any person
* obtaining a copy of this software and associated documentation
* files (the "Software"), to deal in the Software without restriction,
* including without limitation the rights to use, copy, modify, merge,
* publish, distribute, sublicense, and/or sell copies of the Software,
* and to permit persons to whom the Software is furnished to do so,
* subject to the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
 THE WARRANTIES
* OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*--

*/

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/include/safe_str_lib.h
No license file was found, but licenses were detected in source scan.

/*--
* strcpyfldin_s.c
*
* November 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All rights Reserved.
*/

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strcpyfldin_s.c
No license file was found, but licenses were detected in source scan.

/*--
* strljustify_s.c
*
* November 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All rights Reserved
*--
*/

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strljustify_s.c
No license file was found, but licenses were detected in source scan.

/*--
* strncpy_s.c
*
* October 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All rights Reserved.
*--
*/

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strncpy_s.c

No license file was found, but licenses were detected in source scan.

```
/*------------------------------------------------------------------
* strlastchar_s.c
*
* November 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All rights reserved
*/
```

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strlastchar_s.c
No license file was found, but licenses were detected in source scan.

```
/*------------------------------------------------------------------
* safe_mem_lib.h -- Safe C Library Memory APIs
*
* October 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
*
* Permission is hereby granted, free of charge, to any person
* obtaining a copy of this software and associated documentation
* files (the "Software"), to deal in the Software without restriction,
* including without limitation the rights to use, copy, modify, merge,
* publish, distribute, sublicense, and/or sell copies of the Software,
* and to permit persons to whom the Software is furnished to do so,
* subject to the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
 * OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
* CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
* TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
* SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*------------------------------------------------------------------
*/
```

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/include/safe_mem_lib.h
No license file was found, but licenses were detected in source scan.

```
/*------------------------------------------------------------------
* strispassword_s.c
*
* October 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All rights Reserved.
*------------------------------------------------------------------
*/
```

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strispassword_s.c
No license file was found, but licenses were detected in source scan.

```
/*------------------------------------------------------------------
* strtok_s.c
*
* October 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All rights reserved.
*------------------------------------------------------------------
*/
```

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strtok_s.c
No license file was found, but licenses were detected in source scan.

```
/*------------------------------------------------------------------
* strspn_s.c
*
* November 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All rights Reserved.
*------------------------------------------------------------------
*/
```

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strspn_s.c
No license file was found, but licenses were detected in source scan.

```
/*------------------------------------------------------------------
* strislowercase_s.c
*
```

* Februay 2005, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All rights Reserved.
*--
*/

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strislowercase_s.c
No license file was found, but licenses were detected in source scan.

/*--
* strremovews_s.c
*
* November 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All rights reserved.
*--
*/

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strremovews_s.c
No license file was found, but licenses were detected in source scan.

/*--
* strishex_s.c
*
* October 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All Rights reserved.
*/

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strishex_s.c
No license file was found, but licenses were detected in source scan.

/*--
* strprefix_s.c
*
* November 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All rights reserved

```
*-----------------------------------------------------------------
*/
```

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strprefix_s.c
No license file was found, but licenses were detected in source scan.

```
/*-----------------------------------------------------------------
* strlastsame_s.c
*
* November 2008, Bo Berry
*
* Copyright (c) 2008-2010, 2013 by Cisco Systems, Inc.
* All rights reserved
*-----------------------------------------------------------------
*/
```

Found in path(s):
* /opt/cola/permits/1502360938_1670536039.5216763/0/safe-lib-2-0-1-tar-gz/safe_lib_2_0/src/safe_string/strlastsame_s.c

1.19 appcompat 1.4.2

1.19.1 Available under license :

No license file was found, but licenses were detected in source scan.

```
<!--
  Copyright (C) 2016 The Android Open Source Project

  Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

        http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.
-->
```

Found in path(s):
* /opt/cola/permits/1598927406_1686349674.9999151/0/jetified-appcompat-resources-1-4-2-zip/jetified-appcompat-resources-1.4.2/res/drawable/abc_vector_test.xml

No license file was found, but licenses were detected in source scan.

<!--
Copyright (C) 2012 The Android Open Source Project

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.
-->

Found in path(s):
* /opt/cola/permits/1598927406_1686349674.9999151/0/jetified-appcompat-resources-1-4-2-zip/jetified-appcompat-resources-1.4.2/AndroidManifest.xml

1.20 slf4j 2.0.7

1.20.1 Available under license :

No license file was found, but licenses were detected in source scan.

/*
* The contents of this file are subject to the terms of the Common Development and
* Distribution License (the License). You may not use this file except in compliance with the
* License.
*
* You can obtain a copy of the License at legal/CDDLv1.0.txt. See the License for the
* specific language governing permission and limitations under the License.
*
* When distributing Covered Software, include this CDDL Header Notice in each file and include
* the License file at legal/CDDLv1.0.txt. If applicable, add the following below the CDDL
* Header, with the fields enclosed by brackets [] replaced by your own identifying
* information: "Portions Copyrighted [year] [name of copyright owner]".
*
*   Copyright 2011-2014 ForgeRock AS
*/

Found in path(s):
* /opt/cola/permits/1613031661_1680235541.8093038/0/slf4j-2-0-7-sources-jar/org/forgerock/i18n/slf4j/LocalizedLogger.java

No license file was found, but licenses were detected in source scan.

```
<!--
! CDDL HEADER START
!
! The contents of this file are subject to the terms of the
! Common Development and Distribution License, Version 1.0 only
! (the "License").  You may not use this file except in compliance
! with the License.
!
! You can obtain a copy of the license at legal/CDDLv1_0.txt or
! http://forgerock.org/license/CDDLv1.0.html.
! See the License for the specific language governing permissions
! and limitations under the License.
!
! When distributing Covered Code, include this CDDL HEADER in each
! file and include the License file at legal/CDDLv1_0.txt.  If applicable,
! add the following below this CDDL HEADER, with the fields enclosed
! by brackets "[]" replaced with your own identifying information:
!      Portions Copyright [yyyy] [name of copyright owner]
!
! CDDL HEADER END
!
!      Copyright 2011 ForgeRock AS
!
-->
```

Found in path(s):
* /opt/cola/permits/1613031661_1680235541.8093038/0/slf4j-2-0-7-sources-jar/META-INF/maven/org.openidentityplatform.commons.i18n-framework/slf4j/pom.xml
No license file was found, but licenses were detected in source scan.

```
/*
* The contents of this file are subject to the terms of the Common Development and
* Distribution License (the License). You may not use this file except in compliance with the
* License.
*
* You can obtain a copy of the License at legal/CDDLv1.0.txt. See the License for the
* specific language governing permission and limitations under the License.
*
* When distributing Covered Software, include this CDDL Header Notice in each file and include
* the License file at legal/CDDLv1.0.txt. If applicable, add the following below the CDDL
* Header, with the fields enclosed by brackets [] replaced by your own identifying
* information: "Portions Copyrighted [year] [name of copyright owner]".
*
*      Copyright 2011 ForgeRock AS
*/
```

Found in path(s):
* /opt/cola/permits/1613031661_1680235541.8093038/0/slf4j-2-0-7-sources-jar/org/forgerock/i18n/slf4j/LocalizedLoggerFactory.java
* /opt/cola/permits/1613031661_1680235541.8093038/0/slf4j-2-0-7-sources-jar/org/forgerock/i18n/slf4j/package-info.java
No license file was found, but licenses were detected in source scan.

/*
* The contents of this file are subject to the terms of the Common Development and
* Distribution License (the License). You may not use this file except in compliance with the
* License.
*
* You can obtain a copy of the License at legal/CDDLv1.0.txt. See the License for the
* specific language governing permission and limitations under the License.
*
* When distributing Covered Software, include this CDDL Header Notice in each file and include
* the License file at legal/CDDLv1.0.txt. If applicable, add the following below the CDDL
* Header, with the fields enclosed by brackets [] replaced by your own identifying
* information: "Portions Copyrighted [year] [name of copyright owner]".
*
*   Copyright 2014 ForgeRock AS
*/

Found in path(s):
* /opt/cola/permits/1613031661_1680235541.8093038/0/slf4j-2-0-7-sources-jar/org/forgerock/i18n/slf4j/LocalizedMarker.java

1.21 openh264 3.35.0

1.21.1 Available under license :

Copyright (c) 2013, Cisco Systems
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
 list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this
 list of conditions and the following disclaimer in the documentation and/or
 other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR

ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Contributors to the OpenH264 project

Patrick Ai
Sijia Chen
ZhaoZheng Chu
Paley Du
Martin Ettl
Andreas Gal
Xu Guang
Licai Guo
Yi Guo
Horace Huang
Steven Huang
Ethan Hugg
Cullen Jennings
Zhaofeng Jia
Derrick Jin
Jesse Li
Jifei Li
Kai Li
Karina Li
Matt Li
Xiang Li
Bourne Ling
Alex Liu
Wayne Liu
Varun Patil
Eric Rescorla
Adam Roach
Sawyer Shan
Siping Tao
Martin Storsj
Brion Vibber
James Wang
Juanny Wang
Zhiliang Wang
Herv Willems
Gregory J Wolfe
Katherine Wu
Guang Xu
Jeffery Xu

Gang Yang
Li Yao
Jiessie Zhang
Rory Zhang
Volvet Zhang
Ling Zhu
James Zhu
Dong Zhang
Haibo Zhu
Huade Shi

1.22 openssl 1.1.1u

1.22.1 Available under license :

LICENSE ISSUES
==========

The OpenSSL toolkit stays under a double license, i.e. both the conditions of
the OpenSSL License and the original SSLeay license apply to the toolkit.
See below for the actual license texts.

OpenSSL License

/* ==
* Copyright (c) 1998-2018 The OpenSSL Project. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in
*   the documentation and/or other materials provided with the
*   distribution.
*
* 3. All advertising materials mentioning features or use
 of this
*   software must display the following acknowledgment:
*   "This product includes software developed by the OpenSSL Project
*   for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to

* endorse or promote products derived from this software without
* prior written permission. For written permission, please contact
* openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
* nor may "OpenSSL" appear in their names without prior written
* permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
* acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
* ==
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com). This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
* All rights reserved.
*
* This package is an SSL implementation written
* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.
*
* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation

* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
 *
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
*   "This product includes cryptographic software written by
*   Eric Young (eay@cryptsoft.com)"
*   The word 'cryptographic' can be left out if the rouines from the library
*   being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
*   the apps directory (application code) you must
 include an acknowledgement:
*   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e.
 this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price.
Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which
gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we

want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This
License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1

above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices
stating that you changed the files and the date of any change.

b) You must cause any work that
you distribute or publish, that in
whole or in part contains or is derived from the Program or any
part thereof, to be licensed as a whole at no charge to all third
parties under the terms of this License.

c) If the modified program normally reads commands interactively
when run, you must cause it, when started running for such
interactive use in the most ordinary way, to print or display an
announcement including an appropriate copyright notice and a
notice that there is no warranty (or else, saying that you provide
a warranty) and that users may redistribute the program under
these conditions, and telling the user how to view a copy of this
License. (Exception: if the Program itself is interactive but
does not normally print such an announcement, your work based on
the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived
from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may
copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by

modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the
Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if
a patent
license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he
or she is willing
to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number.
If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM,
TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED

TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE
PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free
Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c'
for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public
License instead of this License.

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

b) use the modified Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable

form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided
that you do not advertise this Package as a
product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines
(or comparably compiled subroutines in other
languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always
permitted provided that the use of this Package is embedded; that is,
when no overt attempt is made to make this Package's interfaces visible
to the end user of the commercial distribution. Such use shall not be
construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote
products derived from this software without specific prior written permission.

10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, WITHOUT
 LIMITATION, THE IMPLIED
WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

1.23 nimbus-jose-jwt 7.9

1.23.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this
License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Nimbus JOSE + JWT

Copyright 2012 - 2018, Connect2id Ltd.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use
this file except in compliance with the License. You may obtain a copy of the
License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed
under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, either express or implied. See the License for the
specific language governing permissions and limitations under the License.

1.24 zxing 4.3.0

1.24.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.25 crashlytics 32.0.0

1.25.1 Available under license :

---
nav_order: 1
permalink: /

Contributor documentation

This site is a collection of docs and best practices for contributors to Firebase Android SDKs.
It describes how Firebase works on Android and provides guidance on how to build/maintain a Firebase SDK.

New to Firebase?

- [Development Environment Setup]({{ site.baseurl }}{% link onboarding/env_setup.md %})
- [Creating a new SDK]({{ site.baseurl }}{% link onboarding/new_sdk.md %})
- [How Firebase works]({{ site.baseurl }}{% link how_firebase_works.md %})
---
has_children: true

permalink: /best_practices/
nav_order: 5

Best Practices
---
parent: Best Practices

Dependency Injection

While [Firebase Components]({{ site.baseurl }}{% link components/components.md %}) provides basic Dependency Injection capabilities for interop between Firebase SDKs, it's not ideal as a general purpose DI framework for a few reasons, to name some:

* It's verbose, i.e. requires manually specifying dependencies and constructing instances of components in Component
 definitions.
* It has a runtime cost, i.e. initialization time is linear in the number of Components present in the graph

As a result using [Firebase Components]({{ site.baseurl }}{% link components/components.md %}) is appropriate only
for inter-SDK injection and scoping instances per `FirebaseApp`.

On the other hand, manually instantiating SDKs is often tedious, errorprone, and leads to code smells
that make code less testable and couples it to the implementation rather than the interface. For more context see
[Dependency Injection](https://en.wikipedia.org/wiki/Dependency_injection)
 and [Motivation](https://github.com/google/guice/wiki/Motivation).

{: .important }
It's recommended to use [Dagger](https://dagger.dev) for internal dependency injection within the SDKs and [Components]({{ site.baseurl }}{% link components/components.md %}) to inject inter-sdk dependencies that are available only at
runtime into the [Dagger Graph](https://dagger.dev/dev-guide/#building-the-graph) via
[builder setters](https://dagger.dev/dev-guide/#binding-instances) or [factory
arguments](https://dagger.dev/api/latest/dagger/Component.Factory.html).

See: [Dagger docs](https://dagger.dev)
See: [Dagger tutorial](https://dagger.dev/tutorial/)

{: .highlight }
While Hilt is the recommended way to use dagger in Android applications, it's not suitable for SDK/library development.

How to get started

Since [Dagger](https://dagger.dev) does not strictly follow semver and requires the dagger-compiler version to match the dagger library version,

it's not safe to depend
 on it via a pom level dependency, see [This comment](https://github.com/firebase/firebase-android-sdk/issues/1677#issuecomment-645669608) for context. For this reason in Firebase SDKs we "vendor/repackage" Dagger into the SDK itself under
`com.google.firebase.{sdkname}.dagger`. While it incurs in a size increase, it's usually on the order of a couple of KB and is considered
negligible.

To use Dagger in your SDK use the following in your Gradle build file:

```groovy
plugins {
  id("firebase-vendor")
}

dependencies {
  implementation(libs.javax.inject)
  vendor(libs.dagger.dagger) {
     exclude group: "javax.inject", module: "javax.inject"
  }
  annotationProcessor(libs.dagger.compiler)
}
```

General Dagger setup

As mentioned in [Firebase Components]({{ site.baseurl }}{% link components/components.md %}), all components are scoped per `FirebaseApp`
meaning there is a single instance of the component within a given `FirebaseApp`.

This makes it a natural fit to get all
 inter-sdk dependencies and instatiate the Dagger component inside the `ComponentRegistrar`.

```kotlin
class MyRegistrar : ComponentRegistrar {
 override fun getComponents() = listOf(
  Component.builder(MySdk::class.java)
    .add(Dependency.required(FirebaseOptions::class.java))
    .add(Dependency.optionalProvider(SomeInteropDep::class.java))
    .factory(c -> DaggerMySdkComponent.builder()
      .setFirebaseApp(c.get(FirebaseApp::class.java))
      .setSomeInterop(c.getProvider(SomeInteropDep::class.java))
      .build()
      .getMySdk())
    .build()
}
```

Here's a simple way to define the dagger component:

```kotlin
@Component(modules = MySdkComponent.MainModule::class)
@Singleton
interface MySdkComponent {
 // Informs dagger that this is one of the types we want to be able to create
 // In this example we only care about MySdk
 fun getMySdk() : MySdk

 // Tells Dagger that some types are not available statically and in order to create the component
  // it needs FirebaseApp and Provider<SomeInteropDep>
 @Component.Builder
 interface Builder {
   @BindsInstance fun setFirebaseApp(app: FirebaseApp)
   @BindsInstance fun setSomeInterop(interop: com.google.firebase.inject.Provider<SomeInteropDep>)
   fun build() : MySdkComponent
 }

 @Module
 interface MainModule {
   // define module @Provides and @Binds here
 }
}
```

The only thing left to do is to properly annotate `MySdk`:

```kotlin
@Singleton
class MySdk @Inject constructor(app: FirebaseApp, interopAdapter: MySdkAdapter) {
   fun someMethod() {
      interopAdapter.doInterop()
   }
}

@Singleton
class MySdkInteropAdapter @Inject constructor(private val interop:
com.google.firebase.inject.Provider<SomeInteropDep>) {
   fun doInterop() {
      interop.get().doStuff()
   }
}
```

Scope

Unlike Component, Dagger does not use singleton scope by default and instead injects a new instance of a type at

each injection point,
in the example above we want `MySdk` and
`MySdkInteropAdapter` to be singletons so they are are annotated with `@Singleton`.

See [Scoped bindings](https://dagger.dev/dev-guide/#singletons-and-scoped-bindings) for more details.

Support multiple instances of the SDK per `FirebaseApp`(multi-resource)

As mentioned in [Firebase Components]({{ site.baseurl }}{% link components/components.md %}), some SDKs support multi-resource mode,
which effectively means that there are 2 scopes at play:

1. `@Singleton` scope that the main `MultiResourceComponent` has.
2. Each instance of the sdk will have its own scope.

```mermaid
flowchart LR
 subgraph FirebaseApp
  direction TB
  subgraph FirebaseComponents
   direction BT
   subgraph GlobalComponents[Outside of SDK]
    direction LR

    FirebaseOptions
    SomeInterop
    Executor["@Background Executor"]
   end

   subgraph DatabaseComponent["@Singleton DatabaseMultiDb"]
    direction TB
    subgraph Singleton["@Singleton"]

 SomeImpl -.-> SomeInterop
      SomeImpl -.-> Executor
     end

     subgraph Default["@DbScope SDK(default)"]
      MainClassDefault[FirebaseDatabase] --> SomeImpl
      SomeOtherImplDefault[SomeOtherImpl] -.-> FirebaseOptions
      MainClassDefault --> SomeOtherImplDefault
     end
     subgraph MyDbName["@DbScope SDK(myDbName)"]
      MainClassMyDbName[FirebaseDatabase] --> SomeImpl
      SomeOtherImplMyDbName[SomeOtherImpl] -.-> FirebaseOptions
      MainClassMyDbName --> SomeOtherImplMyDbName
     end
   end
```

```
  end
end

classDef green fill:#4db6ac
classDef blue fill:#1a73e8
class GlobalComponents green
class DatabaseComponent green
class Default blue
class MyDbName blue
```

As you can see above, `DatabaseMultiDb` and `SomeImpl` are singletons, while `FirebaseDatabase` and `SomeOtherImpl` are scoped per `database name`.

It can be easily achieved with the help of [Dagger's subcomponents](https://dagger.dev/dev-guide/subcomponents).

For example:

```kotlin
@Scope
annotation class DbScope

@Component(modules = DatabaseComponent.MainModule::class)
interface DatabaseComponent {
  fun getMultiDb() : DatabaseMultiDb

  @Component.Builder
  interface Builder {
    // usual setters for Firebase component dependencies
    // ...
    fun build() : DatabaseComponent
  }

  @Module(subcomponents = DbInstanceComponent::class)
  interface MainModule {}

  @Subcomponent(modules = DbInstanceComponent.InstanceModule::class)
  @DbScope
  interface DbInstanceComponent {
    fun factory() : Factory

    @Subcomponent.Factory
    interface Factory {
      fun create(@BindsInstance @Named("dbName") dbName: String)
    }
  }
```

```
  @Module
  interface InstanceModule {
      // ...
  }
}
```

Annotating `FirebaseDatabase`:

```kotlin
@DbScope
class FirebaseDatabase @Inject constructor(options: FirebaseOptions, @Named dbName: String) {
 // ...
}
```

Implementing
 `DatabaseMultiDb`:

```kotlin
@Singleton
class DatabaseMultiDb @Inject constructor(private val factory: DbInstanceComponent.Factory) {
 private val instances = mutableMapOf<String, FirebaseDatabase>()

 @Synchronized
 fun get(dbName: String) : FirebaseDatabase {
  if (!instances.containsKey(dbName)) {
    mInstances.put(dbName, factory.create(dbName))
  }
  return mInstances.get(dbName);
 }
}
```
Test license
---
has_children: true
permalink: /components/
nav_order: 4

Firebase Components
{: .no_toc}

1. TOC
{:toc}

Firebase is known for being easy to use and requiring no/minimal configuration at runtime.
Just adding SDKs to the app makes them discover each other to provide additional functionality,

e.g. `Firestore` automatically integrates with `Auth` if present in the app.

* Firebase SDKs have required and optional dependencies on other Firebase SDKs
* SDKs have different initialization requirements, e.g. `Analytics` and `Crashlytics` must be initialized upon application startup, while some are initialized on demand only.

To accommodate these requirements Firebase uses a component model that discovers SDKs present in the app, determines their dependencies and provides them to dependent SDKs via a `Dependency Injection` mechanism.

This page describes the aforementioned Component Model, how it works and why it's needed.

Design Considerations

Transparent/invisible to 3p Developers

To provide good developer experience, we don't want developers to think about how SDKs work and interoperate internally.
Instead we want our SDKs to have a simple API surface that hides all of the internal details.
Most products have an API surface that allows developers to get aninstance of a given SDK via `FirebaseFoo.getInstance()`
and start using it right away.

Simple to use and integrate with for component developers

* The component model is lightweight in terms of integration effort. It is not opinionated on how components are structured.
* The component model should require as little cooperation from components runtime as possible.
* It provides component developers with an API that is easy to use correctly, and hard to use incorrectly.
* Does not sacrifice testability of individual components in isolation

Performant at startup and initialization

The runtime does as little work as possible during initialization.

What is a Component?

A Firebase Component is an entity that:

* Implements one or more interfaces
* Has a list of dependencies(required or optional). See [Dependencies]({{ site.baseurl }}{% link components/dependencies.md %})
* Has initialization requirements(e.g. eager in default app)
* Defines a factory creates an instance of the components interface given it's dependencies. (In other words describes how to create the given component.)

Example:

```java
// Defines a component that is registered as both `FirebaseAuth` and `InternalAuthProvider`.
Component<FirebaseAuth> auth = Component.builder(FirebaseAuth.class, InternalAuthProvider.class)
  // Declares dependencies
  .add(Dependency.required(FirebaseOptions.class))
  // Defines a factory
  .factory(container -> new FirebaseAuth(container.get(FirebaseOptions.class)))
  .eagerInDefaultApp() // alwaysEager() or lazy(), lazy is the default.
  .build()
```

All components are singletons within a Component Container(e.g. one instance per FirebaseApp).
There are however
SDKs that need the ability to expose multiple objects per FirebaseApp,
for example RTBD(as well as Storage and Firestore) has multidb support which allows developers
to access one or more databases within one FirebaseApp. To address this requirement,
SDKs have to register their components in the following form(or similar):

```java
// This is the singleton holder of different instances of FirebaseDatabase.
interface RtdbComponent {
 FirebaseDatabase getDefault();
 FirebaseDatabase get(String databaseName);
}
```

As you can see in the previous section, components are just values and don't have any behavior per se,
essentially they are just blueprints of how to create them and what dependencies they need.

So there needs to be some ComponentRuntime that can discover and wire them together into a dependency graph,
in order to do that, there needs to be an agreed upon location where SDKs can register the components they provide.

The next 2 sections describe how it's done.

Component
 Registration

In order to define the `Components` an SDK provides, it needs to define a class that implements
`ComponentRegistrar`,
this class contains all component definitions the SDK wants to register with the runtime:

```java
public class MyRegistrar implements ComponentRegistrar {
 /// Returns a one or more Components that will be registered in
 /// FirebaseApp and participate in dependency resolution and injection.
 @Override
 public Collection<FirebaseComponent<?>> getComponents() {
```

```
  Arrays.asList(Component.builder(MyType.class)
      /* ... */
      .build());
 }
}
```

Component Discovery

In addition to creating the `ComponentRegistrar` class, SDKs also need to add them to their `AndroidManifest.xml` under `ComponentDiscoveryService`:

```xml
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
 <application>
    <service android:name="com.google.firebase.components.ComponentDiscoveryService"
       android:exported="false">

      <meta-data
          android:name="com.google.firebase.components:com.google.firebase.foo.FirebaseFooRegistrar"
          android:value="com.google.firebase.components.ComponentRegistrar" />
    </service>
 </application>
</manifest>
```

When the final app is built, manifest registrar entries will all end up inside the above `service` as metadata key-value pairs.
At this point `FirebaseApp` will instantiate them and use the `ComponentRuntime` to construct the component graph.

Dependency resolution and initialization

Definitions and constraints

* **Component A depends on Component B** if `B` depends on an `interface` that `A` implements.
* **For any Interface I, only one component is allowed to implement I**(with the exception of
 [Set Dependencies]({{ site.baseurl }}{% link components/dependencies.md %}#set-dependencies)). If this invariant is violated, the container will
 fail to start at runtime.
* **There must not be any dependency cycles** among components.
 See Dependency Cycle Resolution on how this limitation can
 be mitigated
* **Components are initialized lazily by default**(unless a component is declared eager) and are initialized when requested
 by an application either directly or transitively.

The initialization phase of the FirebaseApp will consist of the following steps:

1. Get a list of available FirebaseComponents that were discovered by the Discovery mechanism
2. Topologically sort components based on their declared dependencies - failing if a dependency cycle is detected or multiple implementations are registered for any interface.
3. Store a map of {iface -> ComponentFactory} so that components can be instantiated on demand(Note that component instantiation does not yet happen)
4. Initialize EAGER components or schedule them to initialize on device unlock, if in direct boot mode.

Initialization example

Below is an example illustration of the state of the component graph after initialization:

```mermaid
flowchart
 TD
   Analytics --> Installations
   Auth --> Context
   Auth --> FirebaseOptions
   Context[android.os.Context]
   Crashlytics --> Installations
   Crashlytics --> FirebaseApp
   Crashlytics --> FirebaseOptions
   Crashlytics -.-> Analytics
   Crashlytics --> Context
   Database -.-> Auth
   Database --> Context
   Database --> FirebaseApp
   Database --> FirebaseOptions
   Firestore -.-> Auth
   Messaging --> Installations
   Messaging --> FirebaseOptions
   Messaging --> Context
   RemoteConfig --> FirebaseApp
   RemoteConfig --> Context
   RemoteConfig --> Installations


   classDef eager fill:#4db66e,stroke:#4db6ac,color:#000;
   classDef transitive fill:#4db6ac,stroke:#4db6ac,color:#000;
   classDef always fill:#1a73e8,stroke:#7baaf7,color:#fff;

   class Analytics eager
   class Crashlytics eager
   class Context always
   class FirebaseOptions always
   class FirebaseApp always
   class Installations transitive
```

There are **2
explicitly eager** components in this example: `Crashlytics` and `Analytics`.
These components are initialized when `FirebaseApp` is initialized. `Installations` is initialized eagerly because eager components depends on it(see Prefer Lazy dependencies to avoid this as mush as possible).
`FirebaseApp`, `FirebaseOptions` and `Android Context` are always present in the Component Container and are considered initialized as well.

*The rest of the components are left uninitialized and will remain so until the client application requests them or an eager
component initializes them by using a Lazy dependency.*
For example, if the application calls `FirebaseDatabase.getInstance()`, the container will initialize `Auth` and `Database`
and will return `Database` to the user.

Support multiple instances of the SDK per `FirebaseApp`(multi-resource)

Some SDKs support multi-resource mode of operation, where it's possible to create more than one instance per `FirebaseApp`.

Examples:

* RTDB allows
more than one database in a single Firebase project, so it's possible to instantiate one instance of the sdk per datbase

```kotlin
val rtdbOne = Firebase.database(app) // uses default database
val rtdbTwo = Firebase.database(app, "dbName")
```

* Firestore, functions, and others support the same usage pattern

To allow for that, such SDKs register a singleton "MultiResource" [Firebase component]({{ site.baseurl }}{% link components/components.md %}),
which creates instances per resource(e.g. db name).

Example

```kotlin
class DatabaseComponent(private val app: FirebaseApp, private val tokenProvider: InternalTokenProvider) {
 private val instances: MutableMap<String, FirebaseDatabase> = new HashMap<>();

 @Synchronized
 fun get(dbName: String) : FirebaseDatabase {
  if (!instances.containsKey(dbName)) {
   instances.put(dbName, FirebaseDatabase(app, tokenProvider, dbName))
  }
  return instances.get(dbName);
```

```
 }
}

class FirebaseDatabase(
  app: FirebaseApp,
  tokenProvider:
 InternalTokenProvider,
  private val String dbName)

  companion object {
    fun getInstance(app : FirebaseApp) = getInstance("default")
    fun getInstance(app : FirebaseApp, dbName: String) =
      app.get(DatabaseComponent::class.java).get("default")
  }

```
```
title: Contributor documentation
description: Documentation and best practices for Android SDK development
logo: "https://firebase.google.com/downloads/brand-guidelines/SVG/logo-logomark.svg"

remote_theme: just-the-docs/just-the-docs@v0.4.0.rc3
plugins:
- jekyll-remote-theme

color_scheme: light

# Aux links for the upper right navigation
aux_links:
 "SDK Github Repo":
  - "//github.com/firebase/firebase-android-sdk"

# Enable or disable the site search
# Supports true (default) or false
search_enabled: true
search:
 # Split pages into sections that can be searched individually
 # Supports 1 - 6, default: 2
 heading_level: 6
 # Maximum amount of previews per search result
 # Default: 3
 previews: 2
 # Maximum amount of words to display before a matched word in the preview
 # Default: 5
 preview_words_before: 3
 # Maximum amount of words to display after a matched word in the preview
 # Default: 10
 preview_words_after: 3
 # Set the search token separator
```

```
 # Default: /[\s\-/]+/
  # Example: enable support for hyphenated search words
 tokenizer_separator: /[\s/]+/
 # Display the relative url in search results
 # Supports true (default) or false
 rel_url: true
 # Enable or disable the search button that appears in the bottom right corner of every page
 # Supports true or false (default)
 button: false

mermaid:
 # Version of mermaid library
 # Pick an available version from https://cdn.jsdelivr.net/npm/mermaid/
 version: "9.2.2"
 # Put any additional configuration, such as setting the theme, in _includes/mermaid_config.js

# Enable or disable heading anchors
heading_anchors: true
permalink: pretty

callouts_level: quiet
callouts:
 highlight:
   color: yellow
 important:
   title: Important
   color: blue
 new:
   title: New
   color: green
 note:
   title: Note
   color: purple
 warning:
   title: Warning
   color: red

# Back to top link
back_to_top: true
back_to_top_text: "Back to top"

# Footer last edited timestamp
last_edit_timestamp:
 true # show or hide edit time - page must have `last_modified_date` defined in the frontmatter
last_edit_time_format: "%b %e %Y at %I:%M %p" # uses ruby's time format: https://ruby-doc.org/stdlib-
2.7.0/libdoc/time/rdoc/Time.html
```

```
# Footer "Edit this page on GitHub" link text
gh_edit_link: true # show or hide edit this page link
gh_edit_link_text: "Edit this page on GitHub"
gh_edit_repository: "https://github.com/firebase/firebase-android-sdk" # the github URL for your repo
gh_edit_branch: "master" # the branch that your docs is served from
gh_edit_source: "contributor-docs" # the source that your files originate from
gh_edit_view_mode: "edit" # "tree" or "edit" if you want the user to jump into the editor immediately
name: Copyright check

on: pull_request

concurrency:
 group: ${{ github.workflow }}-${{ github.event.pull_request.number || github.ref }}
 cancel-in-progress: true

jobs:
 copyright-check:
   runs-on: ubuntu-22.04
   steps:
     - uses: actions/checkout@v3.0.2
     - uses: actions/setup-python@v2
       with:
         python-version: '3.9'
     - run: |
         pip install -e "ci/fireci"
     - run: |
         fireci copyright_check \
           -e py                \
           -e gradle             \
           -e java              \
           -e kt               \
           -e groovy             \
           -e sh               \
           -e proto
---
parent: Firebase Components
---

# Dynamic Module Support

TODO
---
parent: Firebase Components
---

# Executors
{: .no_toc}
```

1. TOC
{:toc}

Intro

OS threads are a limited resource that needs to be used with care. In order to minimize the number of threads used by Firebase
as a whole and to increase resource sharing Firebase Common provides a set of standard
[executors](https://developer.android.com/reference/java/util/concurrent/Executor)
and [coroutine dispatchers](https://kotlinlang.org/api/kotlinx.coroutines/kotlinx-coroutines-core/kotlinx.coroutines/-coroutine-dispatcher/)
for use by all Firebase SDKs.

These executors are available as components and can be requested by product SDKs as component dependencies.

Example:

```java
public class MyRegistrar implements ComponentRegistrar {
 public List<Component<?>> getComponents() {
  Qualified<Executor> backgroundExecutor = Qualified.qualified(Background.class, Executor.class);
  Qualified<ExecutorService> liteExecutorService = Qualified.qualified(Lightweight.class, ExecutorService.class);

  return Collections.singletonList(
   Component.builder(MyComponent.class)
    .add(Dependency.required(backgroundExecutor))
    .add(Dependency.required(liteExecutorService))
    .factory(c -> new MyComponent(c.get(backgroundExecutor), c.get(liteExecutorService)))
    .build());
 }
}
```

All executors(with the exception of `@UiThread`) are available as the following interfaces:

* `Executor`
* `ExecutorService`
* `ScheduledExecutorService`
* `CoroutineDispatcher`

`@UiThread` is provided only as a plain `Executor`.

Validation

All SDKs have a custom linter check that detects creation of thread pools and threads,
this is to ensure SDKs use the above executors instead of creating their own.

Choose the right executor

Use the following diagram to pick the right executor for the task you have at hand.

```mermaid
flowchart TD
  Start[Start] --> DoesBlock{Does it block?}
  DoesBlock -->|No| NeedUi{Does it need to run\n on UI thread?}
  NeedUi -->
 |Yes| UiExecutor[[UiThread Executor]]
  NeedUi --> |No| TakesLong{Does it take more than\n 10ms to execute?}
  TakesLong --> |No| LiteExecutor[[Lightweight Executor]]
  TakesLong --> |Yes| BgExecutor[[Background Executor]]
  DoesBlock --> |Yes| DiskIO{Does it block only\n on disk IO?}
  DiskIO --> |Yes| BgExecutor
  DiskIO --> |No| BlockExecutor[[Blocking Executor]]


  classDef start fill:#4db6ac,stroke:#4db6ac,color:#000;
  class Start start

  classDef condition fill:#f8f9fa,stroke:#bdc1c6,color:#000;
  class DoesBlock condition;
  class NeedUi condition;
  class TakesLong condition;
  class DiskIO condition;

  classDef executor fill:#1a73e8,stroke:#7baaf7,color:#fff;
  class UiExecutor executor;
  class LiteExecutor executor;
  class BgExecutor executor;
  class BlockExecutor executor;
```

UiThread

Used to schedule tasks on application's UI thread, internally it uses a Handler to post runnables onto the main looper.

Example:

```java
//
 Java
Qualified<Executor> uiExecutor = qualified(UiThread.class, Executor.class);
```

```kotlin
// Kotlin
```

```
Qualified<CoroutineDispatcher> dispatcher = qualified(UiThread::class.java, CoroutineDispatcher::class.java);
```

Lightweight

Use for tasks that never block and don't take to long to execute. Backed by a thread pool of N threads where N is the amount of parallelism available on the device(number of CPU cores)

Example:

```java
// Java
Qualified<Executor> liteExecutor = qualified(Lightweight.class, Executor.class);
```

```kotlin
// Kotlin
Qualified<CoroutineDispatcher> dispatcher = qualified(Lightweight::class.java, CoroutineDispatcher::class.java);
```

Background

Use for tasks that may block on disk IO(use `@Blocking` for network IO or blocking on other threads). Backed by 4 threads.

Example:

```java
// Java
Qualified<Executor> bgExecutor = qualified(Background.class, Executor.class);
```

```kotlin
// Kotlin
Qualified<CoroutineDispatcher> dispatcher
 = qualified(Background::class.java, CoroutineDispatcher::class.java);
```

Blocking

Use for tasks that can block for arbitrary amounts of time, this includes network IO.

Example:

```java
// Java
Qualified<Executor> blockingExecutor = qualified(Blocking.class, Executor.class);
```

```kotlin
// Kotlin
Qualified<CoroutineDispatcher> dispatcher = qualified(Blocking::class.java, CoroutineDispatcher::class.java);
```

Other executors

Direct executor

{: .warning }
Prefer `@Lightweight` instead of using direct executor as it could cause dead locks and stack overflows.

For any trivial tasks that don't need to run asynchronously

Example:

```kotlin
FirebaseExecutors.directExecutor()
```

Sequential Executor

When you need an executor that runs tasks sequentially and guarantees any memory access is synchronized prefer to use a sequential executor instead of creating a `newSingleThreadedExecutor()`.

Example:

```java
// Pick the appropriate underlying executor using the
 chart above
Qualified<Executor> bgExecutor = qualified(Background.class, Executor.class);
// ...
Executor sequentialExecutor = FirebaseExecutors.newSequentialExecutor(c.get(bgExecutor));
```

Proper Kotlin usage

A `CoroutineContext` should be preferred when possible over an explicit `Executor`
or `CoroutineDispatcher`. You should only use an `Executor` at the highest
(or inversely the lowest) level of your implementations. Most classes should not
be concerned with the existence of an `Executor`.

Keep in mind that you can combine `CoroutineContext` with other `CoroutineScope`
or `CoroutineContext`. And that all `suspend` functions inherent their `coroutineContext`:

```kotlin
suspend fun createSession(): Session {
```

```
 val context = backgroundDispatcher.coroutineContext + coroutineContext
 return Session(context)
}
```

To learn more, you should give the following Kotlin wiki page a read:

[Coroutine context and dispatchers](https://kotlinlang.org/docs/coroutine-context-and-dispatchers.html#dispatchers-and-threads)

Testing

Using Executors in tests

`@Lightweight` and `@Background` executors have StrictMode enabled and throw exceptions on violations. For example trying to do Network IO on either of them will throw.
With that in mind, when it comes to writing tests, prefer to use the common executors as opposed to creating your own thread pools. This will ensure that your code uses the appropriate executor and does not slow down all of Firebase by using the wrong one.

To do that, you should prefer relying on Components to inject the right executor even in tests.
This will ensure your tests are always using the executor that is actually used in your SDK build.
If your SDK uses Dagger, see [Dependency Injection]({{ site.baseurl }}{% link best_practices/dependency_injection.md %})
and [Dagger's testing guide](https://dagger.dev/dev-guide/testing).

When the above is not an option, you can use `TestOnlyExecutors`, but make sure you're testing your code with

the same executor that is used in production code:

```kotlin
dependencies {
 // ...
 testImplementation(project(":integ-testing"))
 // or
 androidTestImplementation(project(":integ-testing"))
}
```

This gives access to

```java
TestOnlyExecutors.ui();
TestOnlyExecutors.background();
TestOnlyExecutors.blocking();
TestOnlyExecutors.lite();
```

Policy violations in tests

Unit tests require [Robolectric](https://github.com/robolectric/robolectric) to
function correctly, and this comes with a major drawback; no policy validation.

Robolectric supports `StrictMode`- but does not provide the backing for its
policy mechanisms to fire on violations. As such, you'll be able to do things
like using `TestOnlyExecutors.background()` to execute blocking actions; usage
that would have otherwise crashed in a real application.

Unfortunately, there is no easy way to fix this for unit tests. You can get
around the issue by moving the tests to an emulator (integration tests)- but
those can
be more expensive than your standard unit test, so you may want to
take that into consideration when planning your testing strategy.

StandardTestDispatcher support

The [kotlin.coroutines.test](https://kotlin.github.io/kotlinx.coroutines/kotlinx-coroutines-test/)
library provides support for a number of different mechanisms in tests. Some of the more
famous features include:

- [advanceUntilIdle](https://kotlin.github.io/kotlinx.coroutines/kotlinx-coroutines-test/kotlinx.coroutines.test/-test-coroutine-scheduler/advance-until-idle.html)
- [advanceTimeBy](https://kotlin.github.io/kotlinx.coroutines/kotlinx-coroutines-test/kotlinx.coroutines.test/-test-coroutine-scheduler/advance-time-by.html)
- [runCurrent](https://kotlin.github.io/kotlinx.coroutines/kotlinx-coroutines-test/kotlinx.coroutines.test/-test-coroutine-scheduler/run-current.html)

These features are all backed by `StandardTestDispatcher`, or more appropriately,
the `TestScope` provided in a `runTest` block.

Unfortunately,
`TestOnlyExecutors` does not natively bind with `TestScope`.
Meaning, should you use `TestOnlyExecutors` in your tests- you won't be able to utilize
the features provided by `TestScope`:

```kotlin
@Test
fun doesStuff() = runTest {
  val scope = CoroutineScope(TestOnlyExecutors.background().asCoroutineDispatcher())
  scope.launch {
    // ... does stuff
  }

  runCurrent() // doesn't invoke scope ??
 }
```

```

To help fix this, we provide an extension method on `TestScope` called
`firebaseExecutors`. It facilitates the binding of `TestOnlyExecutors` with the
current `TestScope`.

For example, here's how you could use this extension method in a test:

```kotlin
@Test
fun doesStuff() = runTest {
  val scope = CoroutineScope(firebaseExecutors.background)
  scope.launch {
    // ... does stuff
  }

  runCurrent()
 }
```
---
nav_order: 3

How Firebase Works

Background

Eager Initialization

One of the biggest strengths for Firebase clients is the ease of integration. In a common case, a developer has very few things to do to integrate with Firebase. There is no need to initialize/configure Firebase at runtime. Firebase automatically initializes at application start and begins providing value to developers. A few notable examples:

* `Analytics` automatically tracks app events
* `Firebase Performance` automatically tracks app startup time, all network requests and screen performance
* `Crashlytics` automatically captures all application crashes, ANRs and non-fatals

This feature makes onboarding and adoption very simple. However, comes with the great responsibility of keeping the application snappy. We shouldn't slow down application startup for 3p developers as it can stand in the way of user adoption of their application.

Automatic Inter-Product Discovery

When present together in an
 application, Firebase products can detect each other and automatically provide additional functionality to the developer, e.g.:

* `Firestore` automatically detects `Auth` and `AppCheck` to protect read/write access to the database
```

* `Crashlytics` integrates with `Analytics`, when available, to provide additional insights into the application behavior and enables safe app rollouts

FirebaseApp at the Core of Firebase

Regardless of what Firebase SDKs are present in the app, the main initialization point of Firebase is `FirebaseApp`. It acts as a container for all SDKs, manages their configuration, initialization and lifecycle.

Initialization

`FirebaseApp` gets initialized with the help of `FirebaseApp#initializeApp()`. This happens [automatically at app startup](https://firebase.blog/posts/2016/12/how-does-firebase-initialize-on-android) or manually by the developer.

During initialization, `FirebaseApp` discovers all Firebase SDKs present in the app, determines the dependency graph between products(for inter-product functionality) and initializes `eager` products that need to start immediately, e.g. `Crashlytics` and `FirebasePerformance`.

Firebase Configuration

`FirebaseApp` contains Firebase configuration for all products to use, namely `FirebaseOptions`, which tells Firebase which `Firebase` project to talk to, which real-time database to use, etc.

Additional Services/Components

In addition to `FirebaseOptions`, `FirebaseApp` registers additional components that product SDKs can request via dependency injection. To name a few:

* `android.content.Context`(Application context)
* [Common Executors]({{ site.baseurl }}{% link components/executors.md %})
* `FirebaseOptions`
* Various internal components

Discovery and Dependency Injection

There are multiple considerations that lead to the current design of how Firebase SDKs initialize.

1. Certain SDKs need to initialize at app startup.
2. SDKs have optional dependencies on other products that
get enabled when the developer adds the dependency to their app.

To enable this functionality, Firebase uses a runtime discovery and dependency injection framework [firebase-components](https://github.com/firebase/firebase-android-sdk/tree/master/firebase-components).

To integrate with this framework SDKs register the components they provide via a `ComponentRegistrar` and declare any dependencies they need to initialize, e.g.

```java
public class MyRegistrar implements ComponentRegistrar {
```

```
 @Override
 public List<Component<?>> getComponents() {
  return Arrays.asList(
     // declare the component
     Component.builder(MyComponent.class)
        // declare dependencies
        .add(Dependency.required(Context.class))
        .add(Dependency.required(FirebaseOptions.class))
        .add(Dependency.optionalProvider(InternalAuthProvider.class))
        // let the runtime know how to create your component.
        .factory(
           diContainer
 ->
              new MyComponent(
                 diContainer.get(Context.class),
                 diContainer.get(FirebaseOptions.class),
                 diContainer.get(InternalAuthProvider.class)))
        .build());
 }
 }
```

This registrar is then registered in `AndroidManifest.xml` of the SDK and is used by `FirebaseApp` to discover all components and construct the dependency graph.

More details in [Firebase Components]({{ site.baseurl }}{% link components/components.md %}).

---
parent: Onboarding

Development Environment Setup

This page describes software and configuration required to work on code in the [Firebase/firebase-android-sdk](https://github.com/firebase/firebase-android-sdk) repository.

{:toc}

JDK

The currently required version of the JDK is `11`. Any other versions are unsupported and using them could result in build failures.

Android Studio

In general, the most recent version of Android Studio should work. The version that is tested at the time of this writing is `Dolphin | 2021.3.1`.

Download it here:
[Download Android Studio](https://developer.android.com/studio)

Emulators

If you plan to run tests on emulators(you should), you should be able to install
them directly from Android Studio's AVD manager.

Github (Googlers Only)

To onboard and get write access to the github repository you need to have a
github account fully linked with go/github.

File a bug using this
[bug template](http://b/issues/new?component=312729&template=1016566)
 and wait
for access to be granted.

After that configure github keys as usual using this
[Github documentation page](https://docs.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh).

Importing the repository

1. Clone the repository with `git clone --recurse-submodules
   git@github.com:firebase/firebase-android-sdk.git`.
1. Open Android Studio and click "Open an existing project".
   ![Open an existing project](as_open_project.png)
1. Find the `firebase-android-sdk` directory and open.
1. To run integration/device tests you will need a `google-services.json` file.

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---
parent: Onboarding

Creating a new Firebase SDK
{: .no_toc}

1. TOC
{:toc}

Want to create a new SDK in
[firebase/firebase-android-sdk](https://github.com/firebase/firebase-android-sdk)?
Read on.

{:toc}

Repository layout and Gradle

[firebase/firebase-android-sdk](https://github.com/firebase/firebase-android-sdk)
uses a multi-project Gradle build to organize the different libraries it hosts.
As a consequence, each project/product within this repo is hosted under its own
subdirectory with its respective build file(s).

```bash
firebase-android-sdk
buildSrc
appcheck
  firebase-appcheck
  firebase-appcheck-playintegrity
firebase-annotations
firebase-common
  firebase-common.gradle # note the name of the build file
  ktx
    ktx.gradle.kts # it can also be kts
build.gradle # root project build file.
```

Most commonly, SDKs are located as immediate child directories of the root
directory, with the directory name being the exact name of the Maven artifact ID
the
 library will have once released. e.g. `firebase-common` directory
hosts code for the `com.google.firebase:firebase-common` SDK.

{: .warning }
Note that the build file name for any given SDK is not `build.gradle` or `build.gradle.kts`
but rather mirrors the name of the sdk, e.g.
`firebase-common/firebase-common.gradle` or `firebase-common/firebase-common.gradle.kts`.

All of the core Gradle build logic lives in `buildSrc` and is used by all
SDKs.

SDKs can be grouped together for convenience by placing them in a directory of
choice.

Creating an SDK

Let's say you want to create an SDK named `firebase-foo`

1. Create a directory called `firebase-foo`.
1. Create a file `firebase-foo/firebase-foo.gradle.kts`.
1. Add `firebase-foo` line to `subprojects.cfg` at the root of the tree.

Update `firebase-foo.gradle.kts` with the following content

<details open markdown="block">
 <summary>
  firebase-foo.gradle.kts
 </summary>
```kotlin
plugins {
 id("firebase-library")
 // Uncomment
 for Kotlin
 // id("kotlin-android")
}

firebaseLibrary {
  // enable this only if you have tests in `androidTest`.
  testLab.enabled = true
  publishSources = true
  publishJavadoc = true
}

android {
 val targetSdkVersion : Int by rootProject
 val minSdkVersion : Int by rootProject

 compileSdk = targetSdkVersion
 defaultConfig {
  namespace = "com.google.firebase.foo"
  // change this if you have custom needs.
  minSdk = minSdkVersion
  targetSdk = targetSdkVersion
  testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
 }
```

```
 testOptions.unitTests.isIncludeAndroidResources = true
}

dependencies {
 implementation("com.google.firebase:firebase-common:20.4.2")
 implementation("com.google.firebase:firebase-components:17.1.5")
}

```
</details>

Create `src/main/AndroidManifest.xml` with the following content:

<details open markdown="block">
 <summary>
  src/main/AndroidManifest.xml
 </summary>
```xml
<?xml version="1.0" encoding="utf-8"?>
<!--
 Copyright {{ 'now' | date: "%Y" }} Google LLC -->
<!-- -->
<!-- Licensed under the Apache License, Version 2.0 (the "License"); -->
<!-- you may not use this file except in compliance with the License. -->
<!-- You may obtain a copy of the License at -->
<!-- -->
<!--      http://www.apache.org/licenses/LICENSE-2.0 -->
<!-- -->
<!-- Unless required by applicable law or agreed to in writing, software -->
<!-- distributed under the License is distributed on an "AS IS" BASIS, -->
<!-- WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. -->
<!-- See the License for the specific language governing permissions and -->
<!-- limitations under the License. -->

<manifest xmlns:android="http://schemas.android.com/apk/res/android">
 <application>
    <service android:name="com.google.firebase.components.ComponentDiscoveryService"
       android:exported="false">
       <meta-data
          android:name="com.google.firebase.components:com.google.firebase.foo.FirebaseFooRegistrar"
           android:value="com.google.firebase.components.ComponentRegistrar" />
    </service>
 </application>
</manifest>
```

</details>

Create `com.google.firebase.foo.FirebaseFoo`

For Kotlin
<details open markdown="block">
 <summary>
  src/main/kotlin/com/google/firebase/foo/FirebaseFoo.kt
 </summary>

```kotlin
class FirebaseFoo {
 companion object {
  @JvmStatic
  val instance: FirebaseFoo
   get() = getInstance(Firebase.app)

  @JvmStatic fun getInstance(app: FirebaseApp): FirebaseFoo = app.get(FirebaseFoo::class.java)
 }
}
```

</details>

For Java
<details markdown="block">
 <summary>
  src/main/java/com/google/firebase/foo/FirebaseFoo.java
 </summary>

```java
public class FirebaseFoo {
 public static FirebaseFoo getInstance() {
  return getInstance(FirebaseApp.getInstance());
 }
 public static FirebaseFoo getInstance(FirebaseApp app) {
  return app.get(FirebaseFoo.class);
 }
}
```

</details>

Create `com.google.firebase.foo.FirebaseFooRegistrar`

For
 Kotlin
<details open markdown="block">
 <summary>
  src/main/kotlin/com/google/firebase/foo/FirebaseFooRegistrar.kt

</summary>

{: .warning }
You should strongly consider using [Dependency Injection]({{ site.baseurl }}{% link best_practices/dependency_injection.md %})
to instantiate your sdk instead of manually constructing its instance in the `factory()` below.

```kotlin
class FirebaseFooRegistrar : ComponentRegistrar {
 override fun getComponents() =
  listOf(
   Component.builder(FirebaseFoo::class.java).factory { container -> FirebaseFoo() }.build(),
   LibraryVersionComponent.create("fire-foo", BuildConfig.VERSION_NAME)
  )
}
```

</details>

For Java
<details markdown="block">
 <summary>
  src/main/java/com/google/firebase/foo/FirebaseFooRegistrar.java
 </summary>

```java
public class FirebaseFooRegistrar implements ComponentRegistrar {
 @Override
 public List<Component<?>> getComponents() {
  return Arrays.asList(
      Component.builder(FirebaseFoo.class).factory(c -> new FirebaseFoo()).build(),
     LibraryVersionComponent.create("fire-foo", BuildConfig.VERSION_NAME));

 }
}
```

</details>

Continue to [How Firebase works]({{ site.baseurl }}{% link how_firebase_works.md %}).
---
has_children: true
permalink: /onboarding/
nav_order: 2

Onboarding

---
parent: Firebase Components

Dependencies
{: .no_toc}

1. TOC
{:toc}

This page gives an overview of the different dependency types supported by the Components Framework.

Background

As discussed in [Firebase Components]({{ site.baseurl }}{% link components/components.md %}), in order for a `Component` to be injected with the things it needs to function, it has to declare its dependencies. These dependencies are then made available and injected into `Components` at runtime.

Firebase Components provide different types of dependencies.

Lazy vs Eager dependencies

When it comes to initialize a component, there are 2 ways of provide its dependencies.

Direct Injection

With this type of injection, the component gets an instance of its dependency directly.

```kotlin
class MyComponent(private val dep : MyDep) {
 fun someMethod() {
   dep.use();
 }
}
```

As you can see above the component's dependency is passed by value directly,
which means that the dependency needs
 to be fully initialized before
it's handed off to the requesting component. As a result `MyComponent` may have to pay the cost of initializing `MyDep` just to be created.

Lazy/Provider Injection

With this type of injection, instead of getting an instance of the dependency directly, the dependency is passed into the `Component` with the help of a `com.google.firebase.inject.Provider`

```java
```

```
public interface Provider<T> { T get(); }
```

```kotlin
class MyComponent(private val dep : Provider<MyDep>) {
 fun someMethod() {
   // Since all components are singletons, each call to
   // get() will return the same instance.
   dep.get().use();
 }
}
```

On the surface this does not look like a big change, but it has an important side effect. In order to create an instance of `MyComponent`, we don't need to initialize `MyDep` anymore. Instead, initialization can be delayed until `MyDep` is actually used.

It is also benefitial to use a `Provider` in the context of [Play's dynamic
 feature delivery](https://developer.android.com/guide/playcore/feature-delivery).
See [Dynamic Module Support]({{ site.baseurl }}{% link components/dynamic_modules.md %}) for more details.

Required dependencies

This type of dependency informs the `ComponentRuntime` that a given `Component` cannot function without a dependency.
When the dependency is missing during initialization, `ComponentRuntime` will throw a `MissingDependencyException`.
This type of dependency is useful for built-in components that are always present like `Context`, `FirebaseApp`, `FirebaseOptions`, [Executors]({{ site.baseurl }}{% link components/executors.md %}).

To declare a required dependency use one of the following in your `ComponentRegistrar`:

```java
 // Required directly injected dependency
 .add(Dependency.required(MyDep.class))
 // Required lazily injected dependency
 .add(Dependency.requiredProvider(MyOtherDep.class))
 .factory( c -> new MyComponent(c.get(MyDep.class), c.getProvider(MyOtherDep.class)))
  .build();
```

Optional Dependencies

This type of dependencies is useful when your `Component` can operate normally when the dependency is not available, but can have enhanced functionality when present. e.g. `Firestore` can work without `Auth` but provides secure database access when `Auth` is present.

To declare an optional dependency use the following in your `ComponentRegistrar`:

```java
 .add(Dependency.optionalProvider(MyDep.class))
 .factory(c -> new MyComponent(c.getProvider(MyDep.class)))
 .build();
```

The provider will return `null` if the dependency is not present in the app.

{: .warning }
When the app uses [Play's dynamic feature delivery](https://developer.android.com/guide/playcore/feature-delivery),
`provider.get()` will return your dependency when it becomes available. To support this use case, don't store references to the result of `provider.get()` calls.

See [Dynamic Module Support]({{ site.baseurl }}{% link components/dynamic_modules.md
 %}) for details

{: .warning }
See Deferred dependencies if you your dependency has a callback based API

Deferred Dependencies

Useful for optional dependencies which have a listener-style API, i.e. the dependent component registers a listener with the dependency and never calls it again (instead the dependency will call the registered listener). A good example is `Firestore`'s use of `Auth`, where `Firestore` registers a token change listener to get notified when a new token is available. The problem is that when `Firestore` initializes, `Auth` may not be present in the app, and is instead part of a dynamic module that can be loaded at runtime on demand.

To solve this problem, Components have a notion of a `Deferred` dependency. A deferred is defined as follows:

```java
public interface Deferred<T> {
 interface DeferredHandler<T> {
   @DeferredApi
   void handle(Provider<T> provider);
 }

 void whenAvailable(DeferredHandler<T> handler);
}
```

To use it a component needs to
 call `Dependency.deferred(SomeType.class)`:

```kotlin
class MyComponent(deferred: Deferred<SomeType>) {
 init {
```

```
  deferred.whenAvailable { someType ->
    someType.registerListener(myListener)
  }
 }
}
```

See [Dynamic Module Support]({{ site.baseurl }}{% link components/dynamic_modules.md %}) for details

Set Dependencies

The Components Framework allows registering components to be part of a set, such components are registered explicitly to be a part of a `Set<T>` as opposed to be a unique value of `T`:

```java
// Sdk 1
Component.intoSet(new SomeTypeImpl(), SomeType.class);
// Sdk 2
Component.intoSetBuilder(SomeType.class)
 .add(Dependency(SomeDep.class))
 .factory(c -> new SomeOtherImpl(c.get(SomeDep.class)))
 .build();
```

With the above setup each SDK contributes a value of `SomeType` into a `Set<SomeType>` which becomes available as a
`Set` dependency.

To consume such a set the interested `Component` needs to declare a special kind of dependency in one of 2 ways:

* `Dependency.setOf(SomeType.class)`, a dependency of type `Set<SomeType>`.
* `Dependency.setOfProvider(SomeType.class)`, a dependency of type `Provider<Set<SomeType>>`. The advantage of this
 is that the `Set` is not initialized until the first call to `provider.get()` at which point all elements of the
 set will get initialized.

{: .warning }
Similar to optional `Provider` dependencies, where an optional dependency can become available at runtime due to [Play's dynamic feature delivery](https://developer.android.com/guide/playcore/feature-delivery),
`Set` dependencies can change at runtime by new elements getting added to the set.
So make sure to hold on to the original `Set` to be able to observe new values in it as they are added.

Example:

```kotlin
class MyClass(private val set1: Set<SomeType>, private val set2: Provider<Set<SomeOtherType>>)
```

```java
Component.builder(MyClass.class)
  .add(Dependency.setOf(SomeType.class))
  .add(Dependency.setOfProvider(SomeOtherType.class))
   .factory(c -> MyClass(c.setOf(SomeType.class), c.setOfProvider(SomeOtherType.class)))
  .build();
```

1.26 kotlin 1.8.10

1.26.1 Available under license :

No license file was found, but licenses were detected in source scan.

{"version":3,"file":"kotlin.js","sources":["wrapper.js","js/arrayUtils.js","js/callableReferenceUtils.js","js/conversions.js","js/core.js","js/long.js","js/markerFunctions.js","js/misc.js","js/polyfills.js","js/rtti.js","runtime/arrayUtils.kt","runtime/Enum.kt","primitiveCompanionObjects.kt","common/src/generated/_Arrays.kt","common/src/generated/_Ranges.kt","unsigned/src/kotlin/UByte.kt","unsigned/src/kotlin/UInt.kt","unsigned/src/kotlin/UShort.kt","src/kotlin/collections/Collections.kt","src/kotlin/collections/Maps.kt","src/kotlin/collections/Sets.kt","src/kotlin/ranges/PrimitiveRanges.kt","src/kotlin/text/StringNumberConversions.kt","src/kotlin/time/Duration.kt","unsigned/src/kotlin/UnsignedUtils.kt","../../core/builtins/src/kotlin/internal/InternalAnnotations.kt","src/kotlin/collections/Iterables.kt","src/kotlin/collections/Sequences.kt","src/kotlin/util/Preconditions.kt","js/src/generated/_ArraysJs.kt","src/kotlin/comparisons/Comparisons.kt","src/kotlin/util/Standard.kt","js/src/generated/_ComparisonsJs.kt","unsigned/src/kotlin/ULong.kt","common/src/generated/_Collections.kt","js/src/kotlin/collections.kt","src/kotlin/collections/Iterators.kt","common/src/generated/_Comparisons.kt","common/src/generated/_Maps.kt","common/src/generated/_OneToManyTitlecaseMappings.kt","js/src/kotlin/text/char.kt","js/src/kotlin/text/string.kt","src/kotlin/text/Char.kt","src/kotlin/CharCode.kt","common/src/generated/_Sequences.kt","common/src/generated/_Sets.kt","common/src/generated/_Strings.kt","src/kotlin/text/Strings.kt","unsigned/src/kotlin/UByteArray.kt","unsigned/src/kotlin/UIntArray.kt","unsigned/src/kotlin/ULongArray.kt","unsigned/src/kotlin/UShortArray.kt","common/src/generated/_UArrays.kt","common/src/generated/_UCollections.kt","common/src/generated/_UComparisons.kt","common/src/generated/_URanges.kt","common/src/generated/_USequences.kt","common/src/kotlin/ExceptionsH.kt","common/src/kotlin/JsAnnotationsH.kt","common/src/kotlin/ioH.kt","builtin-sources/Collections.kt","builtin-sources/Unit.kt","builtin-sources/annotation/Annotations.kt","src/kotlin/builtins.kt","src/kotlin/jsTypeOf.kt","src/kotlin/kotlin.kt","src/kotlin/charCode_js-v1.kt","src/kotlin/coroutines/CoroutineImpl.kt","src/kotlin/util/Result.kt","src/kotlin/coroutines/Continuation.kt","src/kotlin/coroutines/intrinsics/IntrinsicsJs.kt","src/kotlin/currentBeMisc.kt","src/kotlin/exceptions.kt","src/kotlin/jsOperators.kt","src/kotlin/math_js-v1.kt","src/kotlin/numbers_js-v1.kt","src/kotlin/reflection_js-v1.kt","src/kotlin/text/numberConversions_js-v1.kt","js/src/kotlin/js.arrays/fill.kt","js/src/kotlin/js.arrays/sort.kt","js/src/generated/_CharCategories.kt","js/src/generated/_CollectionsJs.kt","js/src/generated/_DigitChars.kt","js/src/generated/_LetterChars.kt","js/src/generated/_OtherLowercaseChars.kt","js/src/generated/_OtherUppercaseChars.kt","js/src/generated/_StringsJs.kt","js/src/generated/_TitlecaseMappings.kt","js/src/generated/_UArraysJs.kt","js/src/generated/_WhitespaceChars.kt","js/src/kotlin/Comparator.kt","js/src/kotlin/annotations.kt","js/src/kotlin/annotationsJVM.kt","js/src/kotlin/collections/AbstractMutableCollection.kt","js/src/kotlin/collections/AbstractMutableList.kt","js/src/kotlin/collections/AbstractMutableMap.kt","js/src/kotlin/collections/AbstractMutableSet.kt","js/src/kotlin/collections/ArrayList.kt","js/src/kotlin/collections/ArraySorting.kt","js/src/kotlin/collections/ArraysJs.kt","js/src/kotlin/collections/EqualityComparator.kt","js/src/kotlin/collections/HashMap.kt","js/src/kotlin/collections/HashSet.kt","js/src/kotlin/collections/InternalHashCodeMap.kt","j

s/src/kotlin/collections/InternalMap.kt","js/src/kotlin/collections/InternalStringMap.kt","js/src/kotlin/collections/LinkedHashMap.kt","js/src/kotlin/collections/LinkedHashSet.kt","js/src/kotlin/concurrent.kt","js/src/kotlin/console.kt","js/src/kotlin/coroutines/SafeContinuationJs.kt","js/src/kotlin/coroutines/cancellation/CancellationException.kt","js/src/kotlin/coroutines/js/internal/EmptyContinuation.kt","js/src/kotlin/date.kt","js/src/kotlin/dom/Builders.kt","js/src/kotlin/dom/Classes.kt","js/src/kotlin/dom/Dom.kt","js/src/kotlin/dom/EventListener.kt","js/src/kotlin/dom/ItemArrayLike.kt","js/src/kotlin/dom/Mutations.kt","js/src/kotlin/dynamic.kt","js/src/kotlin/enums/EnumEntriesSerializationProxy.kt","js/src/kotlin/exceptionUtils.kt","js/src/kotlin/grouping.kt","src/kotlin/collections/Grouping.kt","js/src/kotlin/internalAnnotations.kt","js/src/kotlin/json.kt","js/src/kotlin/math.kt","js/src/kotlin/numbers.kt","js/src/kotlin/promise.kt","js/src/kotlin/random/PlatformRandom.kt","js/src/kotlin/reflect/AssociatedObjects.kt","js/src/kotlin/reflect/JsClass.kt","js/src/kotlin/reflect/KClassImpl.kt","js/src/kotlin/reflect/KClassesImpl.kt","js/src/kotlin/reflect/KTypeHelpers.kt","js/src/kotlin/reflect/KTypeImpl.kt","js/src/kotlin/reflect/KTypeParameterImpl.kt","js/src/kotlin/reflect/primitives.kt","js/src/kotlin/reflect/reflection.kt","js/src/kotlin/regexp.kt","js/src/kotlin/sequence.kt","js/src/kotlin/text/CharCategoryJS.kt","js/src/kotlin/text/CharacterCodingExceptionJs.kt","js/src/kotlin/text/StringBuilderJs.kt","js/src/kotlin/text/numberConversions.kt","js/src/kotlin/text/regex.kt","src/kotlin/text/StringBuilder.kt","js/src/kotlin/text/stringsCode.kt","js/src/kotlin/text/utf8Encoding.kt","js/src/kotlin/throwableExtensions.kt","js/src/kotlin/time/DurationJs.kt","js/src/kotlin/time/DurationUnit.kt","js/src/kotlin/time/MonoTimeSource.kt","js/src/kotlinx/dom/Builders.kt","js/src/kotlinx/dom/Classes.kt","src/kotlin/text/regex/RegexExtensions.kt","js/src/kotlinx/dom/Dom.kt","js/src/kotlinx/dom/Mutations.kt","js/src/org.w3c/deprecated.kt","js/src/org.w3c/org.khronos.webgl.kt","js/src/org.w3c/org.w3c.dom.clipboard.kt","js/src/org.w3c/org.w3c.dom.css.kt","js/src/org.w3c/org.w3c.dom.encryptedmedia.kt","js/src/org.w3c/org.w3c.dom.events.kt","js/src/org.w3c/org.w3c.dom.kt","js/src/org.w3c/org.w3c.fetch.kt","js/src/org.w3c/org.w3c.dom.mediacapture.kt","js/src/org.w3c/org.w3c.dom.mediasource.kt","js/src/org.w3c/org.w3c.dom.pointerevents.kt","js/src/org.w3c/org.w3c.dom.svg.kt","js/src/org.w3c/org.w3c.files.kt","js/src/org.w3c/org.w3c.notifications.kt","js/src/org.w3c/org.w3c.workers.kt","js/src/org.w3c/org.w3c.xhr.kt","src/kotlin/annotations/ExperimentalStdlibApi.kt","src/kotlin/annotations/Inference.kt","src/kotlin/annotations/Multiplatform.kt","src/kotlin/annotations/OptIn.kt","src/kotlin/annotations/WasExperimental.kt","src/kotlin/collections/AbstractCollection.kt","src/kotlin/collections/AbstractIterator.kt","src/kotlin/collections/AbstractList.kt","src/kotlin/collections/AbstractMap.kt","src/kotlin/collections/AbstractSet.kt","src/kotlin/collections/ArrayDeque.kt","src/kotlin/collections/Arrays.kt","src/kotlin/collections/IndexedValue.kt","src/kotlin/collections/MapAccessors.kt","src/kotlin/collections/MapWithDefault.kt","src/kotlin/collections/MutableCollections.kt","src/kotlin/collections/PrimitiveIterators.kt","src/kotlin/collections/ReversedViews.kt","src/kotlin/collections/SequenceBuilder.kt","src/kotlin/collections/SlidingWindow.kt","src/kotlin/collections/UArraySorting.kt","src/kotlin/comparisons/compareTo.kt","src/kotlin/contracts/ContractBuilder.kt","src/kotlin/coroutines/ContinuationInterceptor.kt","src/kotlin/coroutines/CoroutineContext.kt","src/kotlin/coroutines/CoroutineContextImpl.kt","src/kotlin/coroutines/intrinsics/Intrinsics.kt","src/kotlin/enums/EnumEntries.kt","src/kotlin/experimental/ExperimentalObjCName.kt","src/kotlin/experimental/ExperimentalObjCRefinement.kt","src/kotlin/experimental/bitwiseOperations.kt","src/kotlin/experimental/inferenceMarker.kt","src/kotlin/internal/Annotations.kt","src/kotlin/internal/progressionUtil.kt","src/kotlin/properties/Delegates.kt","src/kotlin/properties/Interfaces.kt","src/kotlin/properties/ObservableProperty.kt","src/kotlin/properties/PropertyReferenceDelegates.kt","src/kotlin/random/Random.kt","src/kotlin/random/URandom.kt","src/kotlin/random/XorWowRandom.kt","src/kotlin/ranges/ProgressionIterators.kt","src/kotlin/ranges/Progressions.kt","src/kotlin/ranges/Range.kt","src/kotlin/ranges/Ranges.kt","src/kotlin/reflect/KClasses.kt","src/kotlin/reflect/KTypeProjection.kt","src/kotlin/reflect/KVariance.kt","src/kotlin/reflect/typeOf.kt","src/kotlin/text/Appendable.kt","src/kotlin/text/Indent.kt","src/kotlin/text/Typography.kt","src/kotlin/text/regex/MatchResult.kt","src/kotlin/time/DurationUnit.kt","src/kotlin/time/ExperimentalTime.kt","src/kotlin/time/TimeSource.kt","src/kotlin/time/TimeSources.kt","src/kotlin/time/longSaturatedMath.kt","src/kotlin/time/measureTime.kt","src/kotlin/util/DeepRecursive.kt","src/kotlin/util/FloorDivMod.kt","src/kotlin/util/HashCode.kt","src/kotlin/util/KotlinVersion.kt","src/kotlin/util/Lateinit.kt","src/kotlin/util/Lazy.kt","src/kotlin/util/Numbers.kt","src/kotlin/util/Suspend.kt","src/kotlin/util/Tuples.kt","unsigned/src/kotlin/UIntRange.kt","unsigned/src/kotlin/ULongRange.kt","unsigned/src/kotlin/UMath.kt","unsigned/src/kotlin/UNumbers.kt","unsigned/src/kotlin/UProgressionUtil.kt","unsigned/src/kotlin/UStrings.kt","unsigned/src/kotlin/annotations/Unsigned.kt","common/src/kotlin/MathH.kt","js/src/kotlin/js/js.math.kt"],"sourcesCo

ntent":["(function
(root, factory) {\n if (typeof define === 'function' && define.amd) {\n define('kotlin', ['exports'], factory);\n
}\n else if (typeof exports === 'object') {\n factory(module.exports);\n }\n else {\n root.kotlin = {};\n
  factory(root.kotlin);\n }\n}(this, function (Kotlin) {\n var _ = Kotlin;\n\n insertContent();\n}));\n","/*\n *
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\nKotlin.isBooleanArray = function (a) {\n return (Array.isArray(a) || a instanceof Int8Array) && a.$type$
  === \"BooleanArray\"\n};\n\nKotlin.isByteArray = function (a) {\n return a instanceof Int8Array && a.$type$
!== \"BooleanArray\"\n};\n\nKotlin.isShortArray = function (a) {\n return a instanceof
Int16Array\n};\n\nKotlin.isCharArray = function (a) {\n return a instanceof Uint16Array && a.$type$ ===
\"CharArray\"\n};\n\nKotlin.isIntArray = function (a) {\n return a instanceof
Int32Array\n};\n\nKotlin.isFloatArray = function (a) {\n return a instanceof
Float32Array\n};\n\nKotlin.isDoubleArray = function (a) {\n return a instanceof
Float64Array\n};\n\nKotlin.isLongArray = function (a) {\n return Array.isArray(a) && a.$type$ ===
\"LongArray\"\n};\n\nKotlin.isArray = function (a) {\n return Array.isArray(a) &&
!a.$type$;\n};\n\nKotlin.isArrayish = function (a) {\n return Array.isArray(a) ||
ArrayBuffer.isView(a)\n};\n\nKotlin.arrayToString = function (a) {\n if (a === null) return \"null\"\n var
toString = Kotlin.isCharArray(a) ? String.fromCharCode : Kotlin.toString;\n
  return \"[\" + Array.prototype.map.call(a, function(e) { return toString(e); }).join(\", \") +
\"]\";\n};\n\nKotlin.arrayDeepToString = function (arr) {\n return
Kotlin.kotlin.collections.contentDeepToStringImpl(arr);\n};\n\nKotlin.arrayEquals = function (a, b) {\n if (a ===
b) {\n return true;\n }\n if (a === null || b === null || !Kotlin.isArrayish(b) || a.length !== b.length) {\n
return false;\n }\n\n for (var i = 0, n = a.length; i < n; i++) {\n if (!Kotlin.equals(a[i], b[i])) {\n return
false;\n }\n }\n return true;\n};\n\nKotlin.arrayDeepEquals = function (a, b) {\n return
Kotlin.kotlin.collections.contentDeepEqualsImpl(a, b);\n};\n\nKotlin.arrayHashCode = function (arr) {\n if (arr
=== null) return 0\n var result = 1;\n for (var i = 0, n = arr.length; i < n; i++) {\n result = ((31 * result | 0) +
Kotlin.hashCode(arr[i])) | 0;\n }\n return result;\n};\n\nKotlin.arrayDeepHashCode
  = function (arr) {\n return
Kotlin.kotlin.collections.contentDeepHashCodeImpl(arr);\n};\n\nKotlin.primitiveArraySort = function (array) {\n
array.sort(Kotlin.doubleCompareTo)\n};\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\nKotlin.getCallableRef = function(name, f) {\n f.callableName = name;\n return
f;\n};\n\nKotlin.getPropertyCallableRef = function(name, paramCount, getter, setter) {\n getter.get = getter;\n
getter.set = setter;\n getter.callableName = name;\n return getPropertyRefClass(getter, setter,
propertyRefClassMetadataCache[paramCount]);\n};\n\nfunction getPropertyRefClass(obj, setter, cache) {\n
obj.$metadata$ = getPropertyRefMetadata(typeof setter === \"function\" ? cache.mutable : cache.immutable);\n
obj.constructor = obj;\n return obj;\n}\n\nvar propertyRefClassMetadataCache
  = [\n {\n mutable: { value: null, implementedInterface: function () {\n return
Kotlin.kotlin.reflect.KMutableProperty0 }\n },\n immutable: { value: null, implementedInterface: function
() {\n return Kotlin.kotlin.reflect.KProperty0 }\n }\n },\n {\n mutable: { value: null,
implementedInterface: function () {\n return Kotlin.kotlin.reflect.KMutableProperty1 }\n },\n
immutable: { value: null, implementedInterface: function () {\n return Kotlin.kotlin.reflect.KProperty1 }\n
  }\n }\n];\n\nfunction getPropertyRefMetadata(cache) {\n if (cache.value === null) {\n cache.value = {\n
  interfaces: [cache.implementedInterface()],\n baseClass: null,\n functions: {},\n properties:
{},\n types: {},\n staticMembers: {}\n };\n }\n return cache.value;\n}\n","/*\n * Copyright
2010-2018 JetBrains
  s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\nKotlin.toShort = function (a) {\n return (a &

0xFFFF) << 16 >> 16;\n};\n\nKotlin.toByte = function (a) {\n return (a & 0xFF) << 24 >> 24;\n};\n\nKotlin.toChar = function (a) {\n return a & 0xFFFF;\n};\n\nKotlin.numberToLong = function (a) {\n return a instanceof Kotlin.Long ? a : Kotlin.Long.fromNumber(a);\n};\n\nKotlin.numberToInt = function (a) {\n return a instanceof Kotlin.Long ? a.toInt() : Kotlin.doubleToInt(a);\n};\n\nKotlin.numberToShort = function (a) {\n return Kotlin.toShort(Kotlin.numberToInt(a));\n};\n\nKotlin.numberToByte = function (a) {\n return Kotlin.toByte(Kotlin.numberToInt(a));\n};\n\nKotlin.numberToDouble = function (a) {\n return +a;\n};\n\nKotlin.numberToChar = function (a) {\n return Kotlin.toChar(Kotlin.numberToInt(a));\n};\n\nKotlin.doubleToInt = function(a) {\n if (a > 2147483647) return 2147483647;\n if (a < -2147483648) return -2147483648;\n return a | 0;\n};\n\nKotlin.toBoxedChar = function (a) {\n if (a == null) return a;\n if (a instanceof Kotlin.BoxedChar) return a;\n return new Kotlin.BoxedChar(a);\n};\n\nKotlin.unboxChar = function(a) {\n if (a == null) return a;\n return Kotlin.toChar(a);\n};\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\nKotlin.equals = function (obj1, obj2) {\n if (obj1 == null) {\n return obj2 == null;\n }\n\n if (obj2 == null) {\n return false;\n }\n\n if (obj1 !== obj1) {\n return obj2 !== obj2;\n }\n\n if (typeof obj1 === \"object\" && typeof obj1.equals === \"function\") {\n return obj1.equals(obj2);\n }\n\n if (typeof obj1 === \"number\" && typeof obj2 === \"number\") {\n return obj1 === obj2 && (obj1 !== 0 || 1 / obj1 === 1 / obj2)\n }\n\n return obj1 === obj2;\n};\n\nKotlin.hashCode = function (obj) {\n if (obj == null) {\n return 0;\n }\n var objType = typeof obj;\n if (\"object\" === objType) {\n return \"function\" === typeof obj.hashCode ? obj.hashCode() : getObjectHashCode(obj);\n }\n if (\"function\" === objType) {\n return getObjectHashCode(obj);\n }\n if (\"number\" === objType) {\n return Kotlin.numberHashCode(obj);\n }\n if (\"boolean\" === objType) {\n return Number(obj)\n }\n\n var str = String(obj);\n return getStringHashCode(str);\n};\n\n\nKotlin.toString = function (o) {\n if (o == null) {\n return \"null\";\n }\n else if (Kotlin.isArrayish(o)) {\n return \"[...]\";\n }\n else {\n return o.toString();\n }\n};\n\n/** @const */\nvar POW_2_32 = 4294967296;\n// TODO: consider switching to Symbol type once we are on ES6.\n/** @const */\nvar OBJECT_HASH_CODE_PROPERTY_NAME = \"kotlinHashCodeValue$\";\n\nfunction getObjectHashCode(obj) {\n if (!(OBJECT_HASH_CODE_PROPERTY_NAME in obj)) {\n var hash = (Math.random() * POW_2_32) | 0; // Make 32-bit singed integer.\n Object.defineProperty(obj, OBJECT_HASH_CODE_PROPERTY_NAME, { value: hash, enumerable: false });\n }\n return obj[OBJECT_HASH_CODE_PROPERTY_NAME];\n}\n\nfunction getStringHashCode(str) {\n var hash = 0;\n for (var i = 0; i < str.length; i++) {\n var code = str.charCodeAt(i);\n hash = (hash * 31 + code) | 0; // Keep it 32-bit.\n }\n return hash;\n}\n\nKotlin.identityHashCode = getObjectHashCode;\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Copyright 2009 The Closure Library Authors. All Rights Reserved.\n//\n// Licensed under the Apache License, Version 2.0 (the \"License\");\n// you may not use this file except in compliance with the License.\n// You may obtain a copy of the License at\n//\n// http://www.apache.org/licenses/LICENSE-2.0\n//\n// Unless required by applicable law or agreed to in writing, software\n// distributed under the License is distributed on an \"AS-IS\" BASIS,\n// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.\n\n/**\n * Constructs a 64-bit two's-complement integer, given its low and high 32-bit\n * values as *signed* integers. See the from* functions below for more\n * convenient ways of constructing Longs.\n *\n * The internal representation of a long is the two given signed, 32-bit values.\n * We use 32-bit pieces because these are the size of integers on which\n * Javascript performs bit-operations. For operations like addition and\n * multiplication, we split each number into 16-bit pieces, which can easily be\n * multiplied

within Javascript's floating-point representation without overflow\n * or change in sign.\n *\n * In the algorithms below, we frequently reduce the negative case to the\n * positive case by negating the input(s) and then post-processing the result.\n * Note that we must ALWAYS check specially whether those values are MIN_VALUE\n * (-2^63) because -MIN_VALUE == MIN_VALUE (since 2^63 cannot be represented as\n * a positive number, it overflows back into a negative). Not handling this\n * case would often result in infinite recursion.\n *\n * @param {number} low The low (signed) 32 bits of the long.\n * @param {number} high The high (signed) 32 bits of the long.\n * @constructor\n * @final\n */\nKotlin.Long = function(low, high) {\n /**\n * @type {number}\n * @private\n */\n this.low_ = low | 0; // force into 32 signed bits.\n\n /**\n * @type {number}\n * @private\n */\n this.high_ = high | 0; // force into 32 signed bits.\n};\n\nKotlin.Long.$metadata$ = {\n kind: \"class\",\n simpleName: \"Long\",\n interfaces:[]\n};\n\n\n// NOTE: Common constant values ZERO, ONE, NEG_ONE, etc. are defined below the\n// from* methods on which they depend.\n\n\n/**\n * A cache of the Long representations of small integer values.\n * @type {!Object}\n * @private\n */\nKotlin.Long.IntCache_ = {};\n\n\n/**\n * Returns a Long representing the given (32-bit) integer value.\n * @param {number} value The 32-bit integer in question.\n * @return {!Kotlin.Long} The corresponding Long value.\n */\nKotlin.Long.fromInt = function(value) {\n if (-128 <= value && value < 128) {\n var cachedObj = Kotlin.Long.IntCache_[value];\n if (cachedObj) {\n return cachedObj;\n }\n }\n\n var obj = new Kotlin.Long(value | 0, value < 0 ? -1 : 0);\n if (-128 <= value && value < 128) {\n Kotlin.Long.IntCache_[value] = obj;\n }\n return obj;\n};\n\n\n/**\n * Converts this number value to `Long`.\n * The fractional part, if any, is rounded down towards zero.\n * Returns zero if this `Double` value is `NaN`, `Long.MIN_VALUE` if it's less than `Long.MIN_VALUE`,\n * `Long.MAX_VALUE` if it's bigger than `Long.MAX_VALUE`.\n * @param {number} value The number in question.\n * @return {!Kotlin.Long} The corresponding Long value.\n */\nKotlin.Long.fromNumber = function(value) {\n if (isNaN(value)) {\n return Kotlin.Long.ZERO;\n } else if (value <= -Kotlin.Long.TWO_PWR_63_DBL_) {\n return Kotlin.Long.MIN_VALUE;\n } else if (value + 1 >= Kotlin.Long.TWO_PWR_63_DBL_) {\n return Kotlin.Long.MAX_VALUE;\n } else if (value < 0) {\n return Kotlin.Long.fromNumber(-value).negate();\n } else {\n return new Kotlin.Long(\n (value % Kotlin.Long.TWO_PWR_32_DBL_) | 0,\n (value / Kotlin.Long.TWO_PWR_32_DBL_) | 0);\n }\n};\n\n\n/**\n * Returns a Long representing the 64-bit integer that comes by concatenating\n * the given high and low bits. Each is assumed to use 32 bits.\n * @param {number} lowBits The low 32-bits.\n * @param {number} highBits The high 32-bits.\n * @return {!Kotlin.Long} The corresponding Long value.\n */\nKotlin.Long.fromBits = function(lowBits, highBits) {\n return new Kotlin.Long(lowBits, highBits);\n};\n\n\n/**\n * Returns a Long representation of the given string, written using the given\n * radix.\n * @param {string} str The textual representation of the Long.\n * @param {number=} opt_radix The radix in which the text is written.\n * @return {!Kotlin.Long} The corresponding Long value.\n */\nKotlin.Long.fromString = function(str, opt_radix) {\n if (str.length == 0) {\n throw Error('number format error: empty string');\n }\n\n var radix = opt_radix || 10;\n if (radix < 2 || 36 < radix) {\n throw Error('radix out of range: ' + radix);\n }\n\n if (str.charAt(0) == '-') {\n return Kotlin.Long.fromString(str.substring(1), radix).negate();\n } else if (str.indexOf('-') >= 0) {\n throw Error('number format error: interior \"-\" character: ' + str);\n }\n\n // Do several (8) digits each time through the loop, so as to\n // minimize the calls to the very expensive emulated div.\n var radixToPower = Kotlin.Long.fromNumber(Math.pow(radix, 8));\n\n var result = Kotlin.Long.ZERO;\n for (var i = 0; i < str.length; i += 8) {\n var size = Math.min(8, str.length - i);\n var value = parseInt(str.substring(i, i + size), radix);\n if (size < 8) {\n var power = Kotlin.Long.fromNumber(Math.pow(radix, size));\n result = result.multiply(power).add(Kotlin.Long.fromNumber(value));\n } else {\n result = result.multiply(radixToPower);\n result = result.add(Kotlin.Long.fromNumber(value));\n }\n }\n return result;\n};\n\n\n// NOTE: the compiler should inline these constant values below and then remove\n// these variables, so there should be no runtime penalty for these.\n\n\n/**\n * Number used repeated below in calculations. This must appear before the\n * first call to any from* function below.\n * @type {number}\n * @private\n */\nKotlin.Long.TWO_PWR_16_DBL_

```
 = 1 << 16;\n\n\n/**\n * @type {number}\n * @private\n */\nKotlin.Long.TWO_PWR_24_DBL_ = 1 <<
24;\n\n\n/**\n * @type {number}\n * @private\n */\nKotlin.Long.TWO_PWR_32_DBL_ =\n
Kotlin.Long.TWO_PWR_16_DBL_ * Kotlin.Long.TWO_PWR_16_DBL_;\n\n\n/**\n * @type {number}\n *
@private\n */\nKotlin.Long.TWO_PWR_31_DBL_ =\n    Kotlin.Long.TWO_PWR_32_DBL_ / 2;\n\n\n/**\n *
@type {number}\n * @private\n */\nKotlin.Long.TWO_PWR_48_DBL_ =\n    Kotlin.Long.TWO_PWR_32_DBL_
* Kotlin.Long.TWO_PWR_16_DBL_;\n\n\n/**\n * @type {number}\n * @private\n
*/\nKotlin.Long.TWO_PWR_64_DBL_ =\n    Kotlin.Long.TWO_PWR_32_DBL_ *
Kotlin.Long.TWO_PWR_32_DBL_;\n\n\n/**\n * @type {number}\n * @private\n
*/\nKotlin.Long.TWO_PWR_63_DBL_ =\n    Kotlin.Long.TWO_PWR_64_DBL_ / 2;\n\n\n/** @type
{!Kotlin.Long} */\nKotlin.Long.ZERO = Kotlin.Long.fromInt(0);\n\n\n/** @type {!Kotlin.Long}
*/\nKotlin.Long.ONE = Kotlin.Long.fromInt(1);\n\n\n/** @type {!Kotlin.Long} */\nKotlin.Long.NEG_ONE =
Kotlin.Long.fromInt(-1);\n\n\n/**
 @type {!Kotlin.Long} */\nKotlin.Long.MAX_VALUE =\n    Kotlin.Long.fromBits(0xFFFFFFFF | 0, 0x7FFFFFFF
| 0);\n\n\n/** @type {!Kotlin.Long} */\nKotlin.Long.MIN_VALUE = Kotlin.Long.fromBits(0, 0x80000000 |
0);\n\n\n/**\n * @type {!Kotlin.Long}\n * @private\n */\nKotlin.Long.TWO_PWR_24_ = Kotlin.Long.fromInt(1
<< 24);\n\n\n/** @return {number} The value, assuming it is a 32-bit integer. */\nKotlin.Long.prototype.toInt =
function() {\n  return this.low_;\n};\n\n\n/** @return {number} The closest floating-point representation to this
value. */\nKotlin.Long.prototype.toNumber = function() {\n  return this.high_ *
Kotlin.Long.TWO_PWR_32_DBL_ +\n        this.getLowBitsUnsigned();\n};\n\n/** @return {number} The 32-bit
hashCode of this value. */\nKotlin.Long.prototype.hashCode = function() {\n  return this.high_ ^
this.low_;\n};\n\n/**\n * @param {number=} opt_radix The radix in which the text should be written.\n * @return
{string} The textual representation of this value.\n *
 @override\n */\nKotlin.Long.prototype.toString = function(opt_radix) {\n  var radix = opt_radix || 10;\n  if (radix <
2 || 36 < radix) {\n    throw Error('radix out of range: ' + radix);\n  }\n\n  if (this.isZero()) {\n    return '0';\n  }\n\n  if
(this.isNegative()) {\n    if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n      // We need to change the Long
value before it can be negated, so we remove\n      // the bottom-most digit in this base and then recurse to do the
rest.\n      var radixLong = Kotlin.Long.fromNumber(radix);\n      var div = this.div(radixLong);\n      var rem =
div.multiply(radixLong).subtract(this);\n      return div.toString(radix) + rem.toInt().toString(radix);\n    } else {\n
return '-' + this.negate().toString(radix);\n    }\n  }\n\n  // Do several (5) digits each time through the loop, so as to\n
// minimize the calls to the very expensive emulated div.\n  var radixToPower =
Kotlin.Long.fromNumber(Math.pow(radix, 5));\n\n  var rem = this;\n  var result
 = '';\n  while (true) {\n    var remDiv = rem.div(radixToPower);\n    var intval =
rem.subtract(remDiv.multiply(radixToPower)).toInt();\n    var digits = intval.toString(radix);\n\n    rem = remDiv;\n
 if (rem.isZero()) {\n      return digits + result;\n    } else {\n      while (digits.length < 5) {\n        digits = '0' +
digits;\n      }\n      result = '' + digits + result;\n    }\n  }\n};\n\n\n/** @return {number} The high 32-bits as a signed
value. */\nKotlin.Long.prototype.getHighBits = function() {\n  return this.high_;\n};\n\n\n/** @return {number}
The low 32-bits as a signed value. */\nKotlin.Long.prototype.getLowBits = function() {\n  return
this.low_;\n};\n\n\n/** @return {number} The low 32-bits as an unsigned value.
*/\nKotlin.Long.prototype.getLowBitsUnsigned = function() {\n  return (this.low_ >= 0) ?\n      this.low_ :
Kotlin.Long.TWO_PWR_32_DBL_ + this.low_;\n};\n\n\n/**\n * @return {number} Returns the number of bits
needed to represent the absolute\n *     value
 of this Long.\n */\nKotlin.Long.prototype.getNumBitsAbs = function() {\n  if (this.isNegative()) {\n    if
(this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n      return 64;\n    } else {\n      return
this.negate().getNumBitsAbs();\n    }\n  } else {\n    var val = this.high_ != 0 ? this.high_ : this.low_;\n    for (var bit
= 31; bit > 0; bit--) {\n      if ((val & (1 << bit)) != 0) {\n        break;\n      }\n    }\n    return this.high_ != 0 ? bit + 33
: bit + 1;\n  }\n};\n\n\n/** @return {boolean} Whether this value is zero. */\nKotlin.Long.prototype.isZero =
function() {\n  return this.high_ == 0 && this.low_ == 0;\n};\n\n\n/** @return {boolean} Whether this value is
negative. */\nKotlin.Long.prototype.isNegative = function() {\n  return this.high_ < 0;\n};\n\n\n/** @return
```

{boolean} Whether this value is odd. */\nKotlin.Long.prototype.isOdd = function() {\n return (this.low_ & 1) == 1;\n};\n\n\n/**\n * @param {Kotlin.Long} other Long to compare against.\n * @return {boolean} Whether this Long equals the other.\n */\nKotlin.Long.prototype.equalsLong = function(other) {\n return (this.high_ == other.high_) && (this.low_ == other.low_);\n};\n\n\n/**\n * @param {Kotlin.Long} other Long to compare against.\n * @return {boolean} Whether this Long does not equal the other.\n */\nKotlin.Long.prototype.notEqualsLong = function(other) {\n return (this.high_ != other.high_) || (this.low_ != other.low_);\n};\n\n\n/**\n * @param {Kotlin.Long} other Long to compare against.\n * @return {boolean} Whether this Long is less than the other.\n */\nKotlin.Long.prototype.lessThan = function(other) {\n return this.compare(other) < 0;\n};\n\n\n/**\n * @param {Kotlin.Long} other Long to compare against.\n * @return {boolean} Whether this Long is less than or equal to the other.\n */\nKotlin.Long.prototype.lessThanOrEqual = function(other) {\n return this.compare(other) <= 0;\n};\n\n\n/**\n * @param {Kotlin.Long} other Long to compare against.\n * @return {boolean} Whether this Long is
 greater than the other.\n */\nKotlin.Long.prototype.greaterThan = function(other) {\n return this.compare(other) > 0;\n};\n\n\n/**\n * @param {Kotlin.Long} other Long to compare against.\n * @return {boolean} Whether this Long is greater than or equal to the other.\n */\nKotlin.Long.prototype.greaterThanOrEqual = function(other) {\n return this.compare(other) >= 0;\n};\n\n\n/**\n * Compares this Long with the given one.\n * @param {Kotlin.Long} other Long to compare against.\n * @return {number} 0 if they are the same, 1 if the this is greater, and -1\n *   if the given one is greater.\n */\nKotlin.Long.prototype.compare = function(other) {\n if (this.equalsLong(other)) {\n   return 0;\n }\n\n var thisNeg = this.isNegative();\n var otherNeg = other.isNegative();\n if (thisNeg && !otherNeg) {\n   return -1;\n }\n if (!thisNeg && otherNeg) {\n   return 1;\n }\n\n // at this point, the signs are the same, so subtraction will not overflow\n if (this.subtract(other).isNegative())
 {\n   return -1;\n } else {\n   return 1;\n }\n};\n\n\n/** @return {!Kotlin.Long} The negation of this value. */\nKotlin.Long.prototype.negate = function() {\n if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n   return Kotlin.Long.MIN_VALUE;\n } else {\n   return this.not().add(Kotlin.Long.ONE);\n }\n};\n\n\n/**\n * Returns the sum of this and the given Long.\n * @param {Kotlin.Long} other Long to add to this one.\n * @return {!Kotlin.Long} The sum of this and the given Long.\n */\nKotlin.Long.prototype.add = function(other) {\n // Divide each number into 4 chunks of 16 bits, and then sum the chunks.\n\n var a48 = this.high_ >>> 16;\n var a32 = this.high_ & 0xFFFF;\n var a16 = this.low_ >>> 16;\n var a00 = this.low_ & 0xFFFF;\n\n var b48 = other.high_ >>> 16;\n var b32 = other.high_ & 0xFFFF;\n var b16 = other.low_ >>> 16;\n var b00 = other.low_ & 0xFFFF;\n\n var c48 = 0, c32 = 0, c16 = 0, c00 = 0;\n c00 += a00 + b00;\n c16 += c00 >>> 16;\n c00 &= 0xFFFF;\n
  c16 += a16 + b16;\n c32 += c16 >>> 16;\n c16 &= 0xFFFF;\n c32 += a32 + b32;\n c48 += c32 >>> 16;\n c32 &= 0xFFFF;\n c48 += a48 + b48;\n c48 &= 0xFFFF;\n return Kotlin.Long.fromBits((c16 << 16) | c00, (c48 << 16) | c32);\n};\n\n\n/**\n * Returns the difference of this and the given Long.\n * @param {Kotlin.Long} other Long to subtract from this.\n * @return {!Kotlin.Long} The difference of this and the given Long.\n */\nKotlin.Long.prototype.subtract = function(other) {\n return this.add(other.negate());\n};\n\n\n/**\n * Returns the product of this and the given long.\n * @param {Kotlin.Long} other Long to multiply with this.\n * @return {!Kotlin.Long} The product of this and the other.\n */\nKotlin.Long.prototype.multiply = function(other) {\n if (this.isZero()) {\n   return Kotlin.Long.ZERO;\n } else if (other.isZero()) {\n   return Kotlin.Long.ZERO;\n }\n\n if (this.equalsLong(Kotlin.Long.MIN_VALUE)) {\n   return other.isOdd() ? Kotlin.Long.MIN_VALUE :
 Kotlin.Long.ZERO;\n
  } else if (other.equalsLong(Kotlin.Long.MIN_VALUE)) {\n   return this.isOdd() ? Kotlin.Long.MIN_VALUE :
 Kotlin.Long.ZERO;\n }\n\n if (this.isNegative()) {\n   if (other.isNegative()) {\n   return this.negate().multiply(other.negate());\n   } else {\n   return this.negate().multiply(other).negate();\n   }\n } else if (other.isNegative()) {\n   return this.multiply(other.negate()).negate();\n }\n\n // If both longs are small, use float multiplication\n if (this.lessThan(Kotlin.Long.TWO_PWR_24_) &&\n
 other.lessThan(Kotlin.Long.TWO_PWR_24_)) {\n   return Kotlin.Long.fromNumber(this.toNumber() * other.toNumber());\n }\n\n // Divide each long into 4 chunks of 16 bits, and then add up 4x4 products.\n // We can

skip products that would overflow.\n\n var a48 = this.high_ >>> 16;\n var a32 = this.high_ & 0xFFFF;\n var a16 = this.low_ >>> 16;\n var a00 = this.low_ & 0xFFFF;\n\n var b48 = other.high_ >>> 16;\n var b32 = other.high_ & 0xFFFF;\n var b16 = other.low_
 >>> 16;\n var b00 = other.low_ & 0xFFFF;\n\n var c48 = 0, c32 = 0, c16 = 0, c00 = 0;\n c00 += a00 * b00;\n c16 += c00 >>> 16;\n c00 &= 0xFFFF;\n c16 += a16 * b00;\n c32 += c16 >>> 16;\n c16 &= 0xFFFF;\n c16 += a00 * b16;\n c32 += c16 >>> 16;\n c16 &= 0xFFFF;\n c32 += a32 * b00;\n c48 += c32 >>> 16;\n c32 &= 0xFFFF;\n c32 += a16 * b16;\n c48 += c32 >>> 16;\n c32 &= 0xFFFF;\n c32 += a00 * b32;\n c48 += c32 >>> 16;\n c32 &= 0xFFFF;\n c48 += a48 * b00 + a32 * b16 + a16 * b32 + a00 * b48;\n c48 &= 0xFFFF;\n return Kotlin.Long.fromBits((c16 << 16) | c00, (c48 << 16) | c32);\n};\n\n\n/**\n * Returns this Long divided by the given one.\n * @param {Kotlin.Long} other Long by which to divide.\n * @return {!Kotlin.Long} This Long divided by the given one.\n */\nKotlin.Long.prototype.div = function(other) {\n if (other.isZero()) {\n   throw Error('division by zero');\n } else if (this.isZero()) {\n   return Kotlin.Long.ZERO;\n }\n\n if (this.equalsLong(Kotlin.Long.MIN_VALUE))
 {\n   if (other.equalsLong(Kotlin.Long.ONE) ||\n   other.equalsLong(Kotlin.Long.NEG_ONE)) {\n   return Kotlin.Long.MIN_VALUE; // recall that -MIN_VALUE == MIN_VALUE\n   } else if (other.equalsLong(Kotlin.Long.MIN_VALUE)) {\n   return Kotlin.Long.ONE;\n   } else {\n   // At this point, we have |other| >= 2, so |this/other| < |MIN_VALUE|.\n   var halfThis = this.shiftRight(1);\n   var approx = halfThis.div(other).shiftLeft(1);\n   if (approx.equalsLong(Kotlin.Long.ZERO)) {\n   return other.isNegative() ? Kotlin.Long.ONE : Kotlin.Long.NEG_ONE;\n   } else {\n   var rem = this.subtract(other.multiply(approx));\n   var result = approx.add(rem.div(other));\n   return result;\n   }\n   }\n } else if (other.equalsLong(Kotlin.Long.MIN_VALUE)) {\n   return Kotlin.Long.ZERO;\n }\n\n if (this.isNegative()) {\n   if (other.isNegative()) {\n   return this.negate().div(other.negate());\n   } else {\n   return this.negate().div(other).negate();\n
    }\n } else if (other.isNegative()) {\n   return this.div(other.negate()).negate();\n }\n\n // Repeat the following until the remainder is less than other: find a\n // floating-point that approximates remainder / other *from below*, add this\n // into the result, and subtract it from the remainder. It is critical that\n // the approximate value is less than or equal to the real value so that the\n // remainder never becomes negative.\n var res = Kotlin.Long.ZERO;\n var rem = this;\n while (rem.greaterThanOrEqual(other)) {\n   // Approximate the result of division. This may be a little greater or\n   // smaller than the actual value.\n   var approx = Math.max(1, Math.floor(rem.toNumber() / other.toNumber()));\n\n   // We will tweak the approximate result by changing it in the 48-th digit or\n   // the smallest non-fractional digit, whichever is larger.\n   var log2 = Math.ceil(Math.log(approx) / Math.LN2);\n   var delta = (log2 <= 48) ? 1
 : Math.pow(2, log2 - 48);\n\n   // Decrease the approximation until it is smaller than the remainder. Note\n   // that if it is too large, the product overflows and is negative.\n   var approxRes = Kotlin.Long.fromNumber(approx);\n   var approxRem = approxRes.multiply(other);\n   while (approxRem.isNegative() || approxRem.greaterThan(rem)) {\n   approx -= delta;\n   approxRes = Kotlin.Long.fromNumber(approx);\n   approxRem = approxRes.multiply(other);\n   }\n\n   // We know the answer can't be zero... and actually, zero would cause\n   // infinite recursion since we would make no progress.\n   if (approxRes.isZero()) {\n   approxRes = Kotlin.Long.ONE;\n   }\n\n   res = res.add(approxRes);\n   rem = rem.subtract(approxRem);\n }\n return res;\n};\n\n\n/**\n * Returns this Long modulo the given one.\n * @param {Kotlin.Long} other Long by which to mod.\n * @return {!Kotlin.Long} This Long modulo the given one.\n */\nKotlin.Long.prototype.modulo = function(other)
 {\n return this.subtract(this.div(other).multiply(other));\n};\n\n\n/** @return {!Kotlin.Long} The bitwise-NOT of this value. */\nKotlin.Long.prototype.not = function() {\n return Kotlin.Long.fromBits(~this.low_, ~this.high_);\n};\n\n\n/**\n * Returns the bitwise-AND of this Long and the given one.\n * @param {Kotlin.Long} other The Long with which to AND.\n * @return {!Kotlin.Long} The bitwise-AND of this and the other.\n */\nKotlin.Long.prototype.and = function(other) {\n return Kotlin.Long.fromBits(this.low_ & other.low_,\n   this.high_ & other.high_);\n};\n\n\n/**\n * Returns the bitwise-OR of this Long and the given one.\n *

@param {Kotlin.Long} other The Long with which to OR.\n * @return {!Kotlin.Long} The bitwise-OR of this and the other.\n */\nKotlin.Long.prototype.or = function(other) {\n return Kotlin.Long.fromBits(this.low_ | other.low_,\n   this.high_ | other.high_);\n};\n\n\n/**\n * Returns the bitwise-XOR of this Long and the given one.\n * @param {Kotlin.Long} other The Long with which to XOR.\n * @return {!Kotlin.Long} The bitwise-XOR of this and the other.\n */\nKotlin.Long.prototype.xor = function(other) {\n return Kotlin.Long.fromBits(this.low_ ^ other.low_,\n   this.high_ ^ other.high_);\n};\n\n\n/**\n * Returns this Long with bits shifted to the left by the given amount.\n * @param {number} numBits The number of bits by which to shift.\n * @return {!Kotlin.Long} This shifted to the left by the given amount.\n */\nKotlin.Long.prototype.shiftLeft = function(numBits) {\n numBits &= 63;\n if (numBits == 0) {\n   return this;\n } else {\n   var low = this.low_;\n   if (numBits < 32) {\n   var high = this.high_;\n   return Kotlin.Long.fromBits(\n   low << numBits,\n   (high << numBits) | (low >>> (32 - numBits)));\n   } else {\n   return Kotlin.Long.fromBits(0, low << (numBits - 32));\n   }\n }\n};\n\n\n/**\n * Returns this Long with bits shifted to the right by the given amount.\n * @param {number} numBits The number of bits by which to shift.\n * @return {!Kotlin.Long} This shifted to the right by the given amount.\n */\nKotlin.Long.prototype.shiftRight = function(numBits) {\n numBits &= 63;\n if (numBits == 0) {\n   return this;\n } else {\n   var high = this.high_;\n   if (numBits < 32) {\n   var low = this.low_;\n   return Kotlin.Long.fromBits(\n   (low >>> numBits) | (high << (32 - numBits)),\n   high >> numBits);\n   } else {\n   return Kotlin.Long.fromBits(\n   high >> (numBits - 32),\n   high >= 0 ? 0 : -1);\n   }\n }\n};\n\n\n/**\n * Returns this Long with bits shifted to the right by the given amount, with\n * zeros placed into the new leading bits.\n * @param {number} numBits The number of bits by which to shift.\n * @return {!Kotlin.Long} This shifted to the right by the given amount, with\n *   zeros placed into the new leading bits.\n */\nKotlin.Long.prototype.shiftRightUnsigned
 = function(numBits) {\n numBits &= 63;\n if (numBits == 0) {\n   return this;\n } else {\n   var high = this.high_;\n   if (numBits < 32) {\n   var low = this.low_;\n   return Kotlin.Long.fromBits(\n   (low >>> numBits) | (high << (32 - numBits)),\n   high >>> numBits);\n   } else if (numBits == 32) {\n   return Kotlin.Long.fromBits(high, 0);\n   } else {\n   return Kotlin.Long.fromBits(high >>> (numBits - 32), 0);\n   }\n }\n};\n\n// Support for Kotlin\nKotlin.Long.prototype.equals = function (other) {\n   return other instanceof Kotlin.Long && this.equalsLong(other);\n};\n\nKotlin.Long.prototype.compareTo_11rb$ = Kotlin.Long.prototype.compare;\n\nKotlin.Long.prototype.inc = function() {\n   return this.add(Kotlin.Long.ONE);\n};\n\nKotlin.Long.prototype.dec = function() {\n   return this.add(Kotlin.Long.NEG_ONE);\n};\n\nKotlin.Long.prototype.valueOf = function() {\n   return this.toNumber();\n};\n\nKotlin.Long.prototype.unaryPlus
 = function() {\n   return this;\n};\n\nKotlin.Long.prototype.unaryMinus = Kotlin.Long.prototype.negate;\nKotlin.Long.prototype.inv = Kotlin.Long.prototype.not;\n\nKotlin.Long.prototype.rangeTo = function (other) {\n   return new Kotlin.kotlin.ranges.LongRange(this, other);\n};","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n/**\n * @param {string} id\n * @param {Object} declaration\n */\nKotlin.defineModule = function (id, declaration) {\n};\n\nKotlin.defineInlineFunction = function(tag, fun) {\n return fun;\n};\n\nKotlin.wrapFunction = function(fun) {\n   var f = function() {\n   f = fun();\n   return f.apply(this, arguments);\n   };\n   return function() {\n   return f.apply(this, arguments);\n   };\n};\n\nKotlin.isTypeOf = function(type) {\n   return function (object) {\n   return typeof object === type;\n   }\n};\n\nKotlin.isInstanceOf = function (klass) {\n   return function (object) {\n   return Kotlin.isType(object, klass);\n   }\n};\n\nKotlin.orNull = function (fn) {\n   return function (object) {\n   return object == null || fn(object);\n   }\n};\n\nKotlin.andPredicate = function (a, b) {\n   return function (object) {\n   return a(object) && b(object);\n   }\n};\n\nKotlin.kotlinModuleMetadata = function (abiVersion, moduleName, data) {\n};\n\nKotlin.suspendCall = function(value) {\n   return value;\n};\n\nKotlin.coroutineResult = function(qualifier) {\n   throwMarkerError();\n};\n\nKotlin.coroutineController = function(qualifier) {\n

```
throwMarkerError();\n};\n\nKotlin.coroutineReceiver = function(qualifier) {\n throwMarkerError();\n};\n\nKotlin.setCoroutineResult = function(value, qualifier) {\n throwMarkerError();\n};\n\nKotlin.getReifiedTypeParameterKType = function(typeParameter) {\n throwMarkerError();\n};\n\nfunction
 throwMarkerError() {\n    throw new Error(\n        \"This marker function should never been called. \" +\n \"Looks like compiler did not eliminate it properly. \" +\n        \"Please, report an issue if you caught this exception.\");\n}\n\nKotlin.getFunctionById = function(id, defaultValue) {\n    return function() {\n        return defaultValue;\n    }\n};","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\nKotlin.compareTo = function (a, b) {\n    var typeA = typeof a;\n    if (typeA === \"number\") {\n        if (typeof b === \"number\") {\n            return Kotlin.doubleCompareTo(a, b);\n        }\n return Kotlin.primitiveCompareTo(a, b);\n    }\n    if (typeA === \"string\" || typeA === \"boolean\") {\n        return Kotlin.primitiveCompareTo(a, b);\n    }\n    return a.compareTo_11rb$(b);\n};\n\nKotlin.primitiveCompareTo
 = function (a, b) {\n    return a < b ? -1 : a > b ? 1 : 0;\n};\n\nKotlin.doubleCompareTo = function (a, b) {\n    if (a < b) return -1;\n    if (a > b) return 1;\n\n    if (a === b) {\n        if (a !== 0) return 0;\n\n        var ia = 1 / a;\n        return ia === 1 / b ? 0 : (ia < 0 ? -1 : 1);\n    }\n\n    return a !== a ? (b !== b ? 0 : 1) : -1\n};\n\nKotlin.charInc = function (value) {\n    return Kotlin.toChar(value+1);\n};\n\nKotlin.charDec = function (value) {\n    return Kotlin.toChar(value-1);\n};\n\nKotlin.imul = Math.imul || imul;\n\nKotlin.imulEmulated = imul;\n\nfunction imul(a, b) {\n    return ((a & 0xffff0000) * (b & 0xffff) + (a & 0xffff) * (b | 0)) | 0;\n}\n\n(function() {\n    var buf = new ArrayBuffer(8);\n    var bufFloat64 = new Float64Array(buf);\n    var bufFloat32 = new Float32Array(buf);\n    var bufInt32 = new Int32Array(buf);\n    var lowIndex = 0;\n    var highIndex = 1;\n\n    bufFloat64[0] = -1; // bff00000_00000000\n
    if (bufInt32[lowIndex] !== 0) {\n        lowIndex = 1;\n        highIndex = 0;\n    }\n\n    Kotlin.doubleToBits = function(value) {\n        return Kotlin.doubleToRawBits(isNaN(value) ? NaN : value);\n    };\n\n Kotlin.doubleToRawBits = function(value) {\n        bufFloat64[0] = value;\n        return Kotlin.Long.fromBits(bufInt32[lowIndex], bufInt32[highIndex]);\n    };\n\n    Kotlin.doubleFromBits = function(value) {\n        bufInt32[lowIndex] = value.low_;\n        bufInt32[highIndex] = value.high_;\n        return bufFloat64[0];\n    };\n\n    Kotlin.floatToBits = function(value) {\n        return Kotlin.floatToRawBits(isNaN(value) ? NaN : value);\n    };\n\n    Kotlin.floatToRawBits = function(value) {\n        bufFloat32[0] = value;\n        return bufInt32[0];\n    };\n\n    Kotlin.floatFromBits = function(value) {\n        bufInt32[0] = value;\n        return bufFloat32[0];\n    };\n\n    // returns zero value for number with positive sign bit and non-zero value for
 number with negative sign bit.\n    Kotlin.doubleSignBit = function(value) {\n        bufFloat64[0] = value;\n return bufInt32[highIndex] & 0x80000000;\n    };\n\n    Kotlin.numberHashCode = function(obj) {\n        if ((obj | 0) === obj) {\n            return obj | 0;\n        }\n        else {\n            bufFloat64[0] = obj;\n            return (bufInt32[highIndex] * 31 | 0) + bufInt32[lowIndex] | 0;\n        }\n    }\n})();\n\nKotlin.ensureNotNull = function(x) {\n    return x != null ? x : Kotlin.throwNPE();\n};\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\nif (typeof String.prototype.startsWith === \"undefined\") {\n Object.defineProperty(String.prototype, \"startsWith\", {\n        value: function (searchString, position) {\n position = position || 0;\n            return this.lastIndexOf(searchString,
 position) === position;\n        }\n    });\n}\nif (typeof String.prototype.endsWith === \"undefined\") {\n Object.defineProperty(String.prototype, \"endsWith\", {\n        value: function (searchString, position) {\n            var subjectString = this.toString();\n            if (position === undefined || position > subjectString.length) {\n position = subjectString.length;\n            }\n            position -= searchString.length;\n            var lastIndex = subjectString.indexOf(searchString, position);\n            return lastIndex !== -1 && lastIndex === position;\n        }\n    });\n}\n// ES6 Math polyfills\nif (typeof Math.sign === \"undefined\") {\n    Math.sign = function(x) {\n        x = +x; // convert to a number\n        if (x === 0 || isNaN(x)) {\n            return Number(x);\n        }\n        return x > 0 ? 1 : -1;\n    };\n}\nif (typeof Math.trunc === \"undefined\") {\n    Math.trunc = function(x) {\n        if (isNaN(x)) {\n
```

```
return NaN;\n }\n if (x > 0) {\n return Math.floor(x);\n }\n return Math.ceil(x);\n };\n}\n\n(function() {\n var epsilon = 2.220446049250313E-16;\n var taylor_2_bound = Math.sqrt(epsilon);\n var taylor_n_bound = Math.sqrt(taylor_2_bound);\n var upper_taylor_2_bound = 1/taylor_2_bound;\n var upper_taylor_n_bound = 1/taylor_n_bound;\n\n if (typeof Math.sinh === \"undefined\") {\n Math.sinh = function(x) {\n if (Math.abs(x) < taylor_n_bound) {\n var result = x;\n if (Math.abs(x) > taylor_2_bound) {\n result += (x * x * x) / 6;\n }\n return result;\n } else {\n var y = Math.exp(x);\n var y1 = 1 / y;\n if (!isFinite(y)) return Math.exp(x - Math.LN2);\n if (!isFinite(y1)) return -Math.exp(-x - Math.LN2);\n return (y - y1) / 2;\n }\n };\n }\n if (typeof Math.cosh === \"undefined\") {\n Math.cosh = function(x) {\n var y = Math.exp(x);\n var y1 = 1 / y;\n if (!isFinite(y) || !isFinite(y1)) return Math.exp(Math.abs(x) - Math.LN2);\n return (y + y1) / 2;\n };\n }\n\n if (typeof Math.tanh === \"undefined\") {\n Math.tanh = function(x){\n if (Math.abs(x) < taylor_n_bound) {\n var result = x;\n if (Math.abs(x) > taylor_2_bound) {\n result -= (x * x * x) / 3;\n }\n return result;\n }\n else {\n var a = Math.exp(+x), b = Math.exp(-x);\n return a === Infinity ? 1 : b === Infinity ? -1 : (a - b) / (a + b);\n }\n };\n }\n\n // Inverse hyperbolic function implementations derived from boost special math functions,\n // Copyright Eric Ford & Hubert Holin 2001.\n\n if (typeof Math.asinh === \"undefined\") {\n var asinh = function(x) {\n if (x >= +taylor_n_bound)\n {\n if (x > upper_taylor_n_bound)\n {\n if (x > upper_taylor_2_bound)\n {\n // approximation by laurent series in 1/x at 0+ order from -1 to 0\n return Math.log(x) + Math.LN2;\n }\n else\n {\n // approximation by laurent series in 1/x at 0+ order from -1 to 1\n return Math.log(x * 2 + (1 / (x * 2)));\n }\n }\n else\n {\n return Math.log(x + Math.sqrt(x * x + 1));\n }\n }\n else if (x <= -taylor_n_bound)\n {\n return -asinh(-x);\n }\n else\n {\n // approximation by taylor\n series in x at 0 up to order 2\n var result = x;\n if (Math.abs(x) >= taylor_2_bound)\n {\n var x3 = x * x * x;\n // approximation by taylor series in x at 0 up to order 4\n result -= x3 / 6;\n }\n return result;\n }\n };\n Math.asinh = asinh;\n }\n if (typeof Math.acosh === \"undefined\") {\n Math.acosh = function(x) {\n if (x < 1)\n {\n return NaN;\n }\n else if (x - 1 >= taylor_n_bound)\n {\n if (x > upper_taylor_2_bound)\n {\n // approximation by laurent series in 1/x at 0+ order from -1 to 0\n return Math.log(x) + Math.LN2;\n }\n else\n {\n return Math.log(x + Math.sqrt(x * x - 1));\n }\n }\n else\n {\n var y = Math.sqrt(x - 1);\n // approximation by taylor series in y at 0 up to order 2\n var result = y;\n if (y >= taylor_2_bound)\n {\n var y3 = y * y * y;\n // approximation by taylor series in y at 0 up to order 4\n result -= y3 / 12;\n }\n\n return Math.sqrt(2) * result;\n }\n };\n }\n if (typeof Math.atanh === \"undefined\") {\n Math.atanh = function(x) {\n if (Math.abs(x) < taylor_n_bound) {\n var result = x;\n if (Math.abs(x) > taylor_2_bound) {\n result += (x * x * x) / 3;\n }\n return result;\n }\n return Math.log((1 + x) / (1 - x)) / 2;\n };\n }\n if (typeof Math.log1p === \"undefined\") {\n Math.log1p = function(x) {\n if (Math.abs(x) < taylor_n_bound) {\n var x2 = x * x;\n var x3 = x2 * x;\n var x4 = x3 * x;\n // approximation by taylor series in x at 0 up to order 4\n return (-x4 / 4 + x3 / 3 - x2 / 2 + x);\n }\n return Math.log(x + 1);\n };\n }\n if (typeof Math.expm1 === \"undefined\") {\n Math.expm1 = function(x) {\n if (Math.abs(x) < taylor_n_bound) {\n var x2 = x * x;\n var x3 = x2 * x;\n var x4 = x3 * x;\n // approximation by taylor series in x at 0 up to order 4\n return (x4 / 24 + x3 / 6 + x2 / 2 + x);\n }\n return Math.exp(x) - 1;\n };\n }\n})();\nif (typeof Math.hypot === \"undefined\") {\n Math.hypot = function() {\n var y = 0;\n var length = arguments.length;\n\n for (var i = 0; i < length; i++) {\n if (arguments[i]
```

```
 === Infinity || arguments[i] === -Infinity) {\n                return Infinity;\n            }\n            y += arguments[i] * arguments[i];\n        }\n        return Math.sqrt(y);\n    };\n}\nif (typeof Math.log10 === \"undefined\") {\n Math.log10 = function(x) {\n        return Math.log(x) * Math.LOG10E;\n    };\n}\nif (typeof Math.log2 === \"undefined\") {\n    Math.log2 = function(x) {\n        return Math.log(x) * Math.LOG2E;\n    };\n}\nif (typeof Math.clz32 === \"undefined\") {\n    Math.clz32 = (function(log, LN2) {\n        return function(x) {\n            var asUint = x >>> 0;\n            if (asUint === 0) {\n                return 32;\n            }\n            return 31 - (log(asUint) / LN2 | 0) | 0; // the \"| 0\" acts like math.floor\n        };\n    })(Math.log, Math.LN2);\n}\n\n// For HtmlUnit and PhantomJs\nif (typeof ArrayBuffer.isView === \"undefined\") {\n    ArrayBuffer.isView = function(a) {\n return a != null && a.__proto__ != null && a.__proto__.__proto__
 === Int8Array.prototype.__proto__;\n    };\n}\n\nif (typeof Array.prototype.fill === \"undefined\") {\n    // Polyfill from https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/fill#Polyfill\n Object.defineProperty(Array.prototype, 'fill', {\n        value: function (value) {\n\n            // Steps 1-2.\n            if (this == null) {\n                throw new TypeError('this is null or not defined');\n            }\n\n            var O = Object(this);\n\n            // Steps 3-5.\n            var len = O.length >>> 0;\n\n            // Steps 6-7.\n            var start = arguments[1];\n            var relativeStart = start >> 0;\n\n            // Step 8.\n            var k = relativeStart < 0 ?\n                Math.max(len + relativeStart, 0) :\n                Math.min(relativeStart, len);\n\n            // Steps 9-10.\n            var end = arguments[2];\n            var relativeEnd = end === undefined ?\n                len : end >> 0;\n\n            // Step 11.\n            var finalValue = relativeEnd < 0 ?\n                Math.max(len + relativeEnd, 0) :\n                Math.min(relativeEnd, len);\n\n            // Step 12.\n            while (k < finalValue) {\n                O[k] = value;\n                k++;\n            }\n\n            // Step 13.\n            return O;\n        }\n    });\n}\n\n(function() {\n    function normalizeOffset(offset, length) {\n        if (offset < 0) return Math.max(0, offset + length);\n        return Math.min(offset, length);\n    }\n    function typedArraySlice(begin, end) {\n        if (typeof end === \"undefined\") {\n            end = this.length;\n        }\n        begin = normalizeOffset(begin || 0, this.length);\n        end = Math.max(begin, normalizeOffset(end, this.length));\n        return new this.constructor(this.subarray(begin, end));\n    }\n\n    var arrays = [Int8Array, Int16Array, Uint16Array, Int32Array,
 Float32Array, Float64Array];\n    for (var i = 0; i < arrays.length; ++i) {\n        var TypedArray = arrays[i];\n        if (typeof TypedArray.prototype.fill === \"undefined\") {\n            Object.defineProperty(TypedArray.prototype, 'fill', {\n                value: Array.prototype.fill\n            });\n        }\n        if (typeof TypedArray.prototype.slice === \"undefined\") {\n            Object.defineProperty(TypedArray.prototype, 'slice', {\n                value: typedArraySlice\n            });\n        }\n    }\n\n    // Patch apply to work with TypedArrays if needed.\n    try {\n (function() {}).apply(null, new Int32Array(0))\n    } catch (e) {\n        var apply = Function.prototype.apply;\n Object.defineProperty(Function.prototype, 'apply', {\n            value: function(self, array) {\n                return apply.call(this, self, [].slice.call(array));\n            }\n        });\n    }\n\n\n    // Patch map to work with TypedArrays if needed.\n
    for (var i = 0; i < arrays.length; ++i) {\n        var TypedArray = arrays[i];\n        if (typeof TypedArray.prototype.map === \"undefined\") {\n            Object.defineProperty(TypedArray.prototype, 'map', {\n                value: function(callback, self) {\n                    return [].slice.call(this).map(callback, self);\n                }\n            });\n        }\n    }\n\n    // Patch sort to work with TypedArrays if needed.\n    // TODO: consider to remove following function and replace it with `Kotlin.doubleCompareTo` (see misc.js)\n    var totalOrderComparator = function (a, b) {\n        if (a < b) return -1;\n        if (a > b) return 1;\n\n        if (a === b) {\n            if (a !== 0) return 0;\n\n            var ia = 1 / a;\n            return ia === 1 / b ? 0 : (ia < 0 ? -1 : 1);\n        }\n\n        return a !== a ? (b !== b ? 0 : 1) : -1\n    };\n\n    for (var i = 0; i < arrays.length; ++i) {\n        var TypedArray = arrays[i];\n        if (typeof TypedArray.prototype.sort
 === \"undefined\") {\n            Object.defineProperty(TypedArray.prototype, 'sort', {\n                value: function(compareFunction) {\n                    return Array.prototype.sort.call(this, compareFunction || totalOrderComparator);\n                }\n            });\n        }\n    }\n})();\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors. \n * Use of this source code is governed by the Apache 2.0
```

license that can be found in the license/LICENSE.txt file.\n */\n\nKotlin.Kind = {\n CLASS: \"class\",\n INTERFACE: \"interface\",\n OBJECT: \"object\"\n};\n\nKotlin.callGetter = function (thisObject, klass, propertyName) {\n var propertyDescriptor = Object.getOwnPropertyDescriptor(klass, propertyName);\n if (propertyDescriptor != null && propertyDescriptor.get != null) {\n return propertyDescriptor.get.call(thisObject);\n }\n\n propertyDescriptor = Object.getOwnPropertyDescriptor(thisObject,
propertyName);\n if (propertyDescriptor != null && \"value\" in propertyDescriptor) {\n return thisObject[propertyName];\n }\n\n return Kotlin.callGetter(thisObject, Object.getPrototypeOf(klass), propertyName);\n};\n\nKotlin.callSetter = function (thisObject, klass, propertyName, value) {\n var propertyDescriptor = Object.getOwnPropertyDescriptor(klass, propertyName);\n if (propertyDescriptor != null && propertyDescriptor.set != null) {\n propertyDescriptor.set.call(thisObject, value);\n return;\n }\n\n propertyDescriptor = Object.getOwnPropertyDescriptor(thisObject, propertyName);\n if (propertyDescriptor != null && \"value\" in propertyDescriptor) {\n thisObject[propertyName] = value;\n return\n }\n\n Kotlin.callSetter(thisObject, Object.getPrototypeOf(klass), propertyName, value);\n};\n\nfunction isInheritanceFromInterface(ctor, iface) {\n if (ctor === iface) return true;\n\n var metadata = ctor.$metadata$;\n if (metadata != null) {\n var interfaces = metadata.interfaces;\n for (var i = 0; i < interfaces.length; i++) {\n if (isInheritanceFromInterface(interfaces[i], iface)) {\n return true;\n }\n }\n }\n\n var superPrototype = ctor.prototype != null ? Object.getPrototypeOf(ctor.prototype) : null;\n var superConstructor = superPrototype != null ? superPrototype.constructor : null;\n return superConstructor != null && isInheritanceFromInterface(superConstructor, iface);\n}\n\n/**\n *\n * @param {*} object\n * @param {Function|Object} klass\n * @returns {Boolean}\n */\nKotlin.isType = function (object, klass) {\n if (klass === Object) {\n switch (typeof object) {\n case \"string\":\n case \"number\":\n case \"boolean\":\n case \"function\":\n return true;\n default:\n return object instanceof Object;\n }\n }\n\n
if (object == null || klass == null || (typeof object !== 'object' && typeof object !== 'function')) {\n return false;\n }\n\n if (typeof klass === \"function\" && object instanceof klass) {\n return true;\n }\n\n var proto = Object.getPrototypeOf(klass);\n var constructor = proto != null ? proto.constructor : null;\n if (constructor != null && \"$metadata$\" in constructor) {\n var metadata = constructor.$metadata$;\n if (metadata.kind === Kotlin.Kind.OBJECT) {\n return object === klass;\n }\n }\n\n var klassMetadata = klass.$metadata$;\n\n // In WebKit (JavaScriptCore) for some interfaces from DOM typeof returns \"object\", nevertheless they can be used in RHS of instanceof\n if (klassMetadata == null) {\n return object instanceof klass;\n }\n\n if (klassMetadata.kind === Kotlin.Kind.INTERFACE && object.constructor != null) {\n return isInheritanceFromInterface(object.constructor, klass);\n }\n\n return false;\n};\n\nKotlin.isNumber = function (a) {\n return typeof a == \"number\" || a instanceof Kotlin.Long;\n};\n\nKotlin.isChar = function (value) {\n return value instanceof Kotlin.BoxedChar\n};\n\nKotlin.isComparable = function (value) {\n var type = typeof value;\n\n return type === \"string\" ||\n type === \"boolean\" ||\n Kotlin.isNumber(value) ||\n Kotlin.isType(value, Kotlin.kotlin.Comparable);\n};\n\nKotlin.isCharSequence = function (value) {\n return typeof value === \"string\" || Kotlin.isType(value, Kotlin.kotlin.CharSequence);\n};","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// a package is omitted to get declarations directly under the module\n\n@PublishedApi\nexternal internal fun <T> Array(size: Int): Array<T>\n\n@JsName(\"newArray\")\nfun <T> newArray(size: Int, initValue: T) = fillArrayVal(Array<T>(size), initValue)\n\n@JsName(\"newArrayF\")\ninline fun <T> arrayWithFun(size: Int, init: (Int) -> T) = fillArrayFun(Array<T>(size), init)\n\n@JsName(\"fillArray\")\ninline fun <T> fillArrayFun(array: Array<T>, init: (Int) -> T): Array<T> {\n for (i in 0..array.size - 1) {\n array[i] = init(i)\n }\n return array\n}\n\n@JsName(\"booleanArray\")\nfun booleanArray(size: Int, init: dynamic): Array<Boolean> {\n val result: dynamic = Array<Boolean>(size)\n result.`$type$` = \"BooleanArray\"\n return when (init) {\n null,

true -> fillArrayVal(result, false)\n false -> result\n else -> fillArrayFun<Boolean>(result, init)\n }\n}\n\n@JsName(\"booleanArrayF\")\ninline fun booleanArrayWithFun(size: Int, init: (Int) -> Boolean): Array<Boolean> = fillArrayFun(booleanArray(size, false), init)\n\n@JsName(\"charArray\")\n@Suppress(\"UNUSED_PARAMETER\")\nfun charArray(size: Int, init: dynamic): Array<Char> {\n val result = js(\"new Uint16Array(size)\")\n result.`$type$` = \"CharArray\"\n return when (init) {\n null, true, false -> result // For consistency\n else -> fillArrayFun<Char>(result, init)\n }\n}\n\n@JsName(\"charArrayF\")\ninline fun charArrayWithFun(size: Int, init: (Int) -> Char): Array<Char> {\n val array = charArray(size, null)\n for (i in 0..array.size - 1) {\n @Suppress(\"UNUSED_VARIABLE\") // used in js block\n val value = init(i)\n js(\"array[i] = value;\")\n }\n return array\n}\n\n@JsName(\"untypedCharArrayF\")\ninline fun untypedCharArrayWithFun(size: Int, init: (Int) -> Char): Array<Char> {\n val array = Array<Char>(size)\n for (i in 0..array.size - 1) {\n @Suppress(\"UNUSED_VARIABLE\") // used in js block\n val value = init(i)\n js(\"array[i] = value;\")\n }\n return array\n}\n\n@JsName(\"longArray\")\nfun longArray(size: Int, init: dynamic): Array<Long> {\n val result: dynamic = Array<Long>(size)\n result.`$type$` = \"LongArray\"\n return when (init) {\n null, true -> fillArrayVal(result, 0L)\n false -> result\n else -> fillArrayFun<Long>(result, init)\n }\n}\n\n@JsName(\"longArrayF\")\ninline fun longArrayWithFun(size: Int, init: (Int) -> Long): Array<Long> = fillArrayFun(longArray(size, false), init)\n\nprivate fun <T> fillArrayVal(array: Array<T>, initValue: T): Array<T> {\n for (i in 0..array.size - 1) {\n array[i] = initValue\n }\n return array\n}","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\npublic class Enum<T : Enum<T>> : Comparable<Enum<T>> {\n @JsName(\"name$\") private var _name: String = \"\"\n @JsName(\"ordinal$\") private var _ordinal: Int = 0\n\n val name: String\n get() = _name\n\n val ordinal: Int\n get() = _ordinal\n\n override fun compareTo(other: Enum<T>) = ordinal.compareTo(other.ordinal)\n\n override fun equals(other: Any?) = this === other\n\n override fun hashCode(): Int = js(\"Kotlin.identityHashCode\")(this)\n\n override fun toString() = name\n\n companion object\n}","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js.internal\n\n@JsName(\"DoubleCompanionObject\")\ninternal object DoubleCompanionObject {\n @JsName(\"MIN_VALUE\")\n const val MIN_VALUE: Double = 4.9E-324\n\n @JsName(\"MAX_VALUE\")\n const val MAX_VALUE: Double = 1.7976931348623157E308\n\n @JsName(\"POSITIVE_INFINITY\")\n @Suppress(\"DIVISION_BY_ZERO\")\n const val POSITIVE_INFINITY: Double = 1.0 / 0.0\n\n @JsName(\"NEGATIVE_INFINITY\")\n @Suppress(\"DIVISION_BY_ZERO\")\n const val NEGATIVE_INFINITY: Double = -1.0 / 0.0\n\n @JsName(\"NaN\")\n @Suppress(\"DIVISION_BY_ZERO\")\n const val NaN: Double = -(0.0 / 0.0)\n\n @JsName(\"SIZE_BYTES\")\n const val SIZE_BYTES = 8\n\n @JsName(\"SIZE_BITS\")\n const val SIZE_BITS = 64\n}\n\n@JsName(\"FloatCompanionObject\")\ninternal object FloatCompanionObject {\n @JsName(\"MIN_VALUE\")\n const val MIN_VALUE: Float = 1.4E-45F\n\n @JsName(\"MAX_VALUE\")\n const val MAX_VALUE: Float = 3.4028235E38F\n\n @JsName(\"POSITIVE_INFINITY\")\n @Suppress(\"DIVISION_BY_ZERO\")\n const val POSITIVE_INFINITY: Float = 1.0F / 0.0F\n\n @JsName(\"NEGATIVE_INFINITY\")\n @Suppress(\"DIVISION_BY_ZERO\")\n const val NEGATIVE_INFINITY: Float = -1.0F / 0.0F\n\n @JsName(\"NaN\")\n @Suppress(\"DIVISION_BY_ZERO\")\n const val NaN: Float = -(0.0F / 0.0F)\n\n @JsName(\"SIZE_BYTES\")\n const val SIZE_BYTES = 4\n\n @JsName(\"SIZE_BITS\")\n const val SIZE_BITS = 32\n}\n\n@JsName(\"IntCompanionObject\")\ninternal object IntCompanionObject {\n @JsName(\"MIN_VALUE\")\n val MIN_VALUE: Int = -2147483647 - 1\n\n @JsName(\"MAX_VALUE\")\n

val MAX_VALUE: Int = 2147483647\n\n   @JsName(\"SIZE_BYTES\")\n   const val SIZE_BYTES = 4\n\n   @JsName(\"SIZE_BITS\")\n   const val SIZE_BITS = 32\n}\n\n@JsName(\"LongCcompanionObject\")\ninternal object LongCompanionObject {\n   @JsName(\"MIN_VALUE\")\n   val MIN_VALUE: Long = js(\"Kotlin.Long.MIN_VALUE\")\n\n   @JsName(\"MAX_VALUE\")\n   val MAX_VALUE: Long = js(\"Kotlin.Long.MAX_VALUE\")\n\n   @JsName(\"SIZE_BYTES\")\n   const val SIZE_BYTES = 8\n\n   @JsName(\"SIZE_BITS\")\n   const val SIZE_BITS = 64\n}\n\n@JsName(\"ShortCompanionObject\")\ninternal object ShortCompanionObject {\n   @JsName(\"MIN_VALUE\")\n   val MIN_VALUE: Short = -32768\n\n   @JsName(\"MAX_VALUE\")\n   val MAX_VALUE: Short = 32767\n\n   @JsName(\"SIZE_BYTES\")\n   const val SIZE_BYTES = 2\n\n   @JsName(\"SIZE_BITS\")\n   const val SIZE_BITS = 16\n}\n\n@JsName(\"ByteCompanionObject\")\ninternal object ByteCompanionObject {\n   @JsName(\"MIN_VALUE\")\n   val MIN_VALUE: Byte = -128\n\n   @JsName(\"MAX_VALUE\")\n   val MAX_VALUE: Byte = 127\n\n   @JsName(\"SIZE_BYTES\")\n   const val SIZE_BYTES = 1\n\n   @JsName(\"SIZE_BITS\")\n   const val SIZE_BITS = 8\n}\n\n@JsName(\"CharCompanionObject\")\ninternal object CharCompanionObject {\n   @JsName(\"MIN_VALUE\")\n   public const val MIN_VALUE: Char = '\\u0000'\n\n   @JsName(\"MAX_VALUE\")\n   public const val MAX_VALUE: Char = '\\uFFFF'\n\n   @JsName(\"MIN_HIGH_SURROGATE\")\n   public const val MIN_HIGH_SURROGATE: Char = '\\uD800'\n\n   @JsName(\"MAX_HIGH_SURROGATE\")\n   public const val MAX_HIGH_SURROGATE: Char = '\\uDBFF'\n\n   @JsName(\"MIN_LOW_SURROGATE\")\n   public const val MIN_LOW_SURROGATE: Char = '\\uDC00'\n\n   @JsName(\"MAX_LOW_SURROGATE\")\n   public const val MAX_LOW_SURROGATE: Char = '\\uDFFF'\n\n   @JsName(\"MIN_SURROGATE\")\n   public const val MIN_SURROGATE: Char = MIN_HIGH_SURROGATE\n\n   @JsName(\"MAX_SURROGATE\")\n   public const val MAX_SURROGATE: Char = MAX_LOW_SURROGATE\n\n   @JsName(\"SIZE_BYTES\")\n   const val SIZE_BYTES = 2\n\n   @JsName(\"SIZE_BITS\")\n   const val SIZE_BITS = 16\n}\n\ninternal object StringCompanionObject {}\n\ninternal object BooleanCompanionObject {}\n\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"ArraysKt\")\n\npackage kotlin.collections\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.random.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component1(): T {\n   return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component1(): Byte {\n   return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component1(): Short {\n   return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component1(): Int {\n   return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component1(): Long {\n   return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component1(): Float {\n   return

get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n
* where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component1(): Double {\n return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component1(): Boolean {\n return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component1(): Char {\n return get(0)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic
inline operator fun <T> Array<out T>.component2(): T {\n return get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component2(): Byte {\n return get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component2(): Short {\n return get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component2():
Int {\n return get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component2(): Long {\n return get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component2(): Float {\n return get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component2(): Double {\n return get(1)\n}\n\n/**\n * Returns
2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component2(): Boolean {\n return get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component2(): Char {\n return get(1)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component3(): T {\n return get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size
of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component3(): Byte {\n return get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component3(): Short {\n return get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n

*/\n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component3(): Int {\n return get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an [IndexOutOfBoundsException]
except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component3(): Long {\n return get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component3(): Float {\n return get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component3(): Double {\n return get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*/\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component3(): Boolean {\n return get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component3(): Char {\n return get(2)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component4(): T {\n return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline
operator fun ByteArray.component4(): Byte {\n return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component4(): Short {\n return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component4(): Int {\n return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component4(): Long {\n return get(3)\n}\n\n/**\n
* Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component4(): Float {\n return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component4(): Double {\n return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component4(): Boolean {\n return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n
* \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component4(): Char {\n return get(3)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Array<out T>.component5(): T {\n return get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an

[IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun ByteArray.component5(): Byte {\n return get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun ShortArray.component5(): Short {\n return get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun IntArray.component5(): Int {\n return get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun LongArray.component5(): Long {\n return get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun FloatArray.component5(): Float {\n return get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun DoubleArray.component5(): Double {\n return get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun BooleanArray.component5(): Boolean {\n return get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun CharArray.component5(): Char {\n return get(4)\n}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\npublic operator fun <@kotlin.internal.OnlyInputTypes T> Array<out T>.contains(element: T): Boolean {\n return indexOf(element) >= 0\n}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\npublic operator fun ByteArray.contains(element: Byte): Boolean {\n return indexOf(element) >= 0\n}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\npublic operator fun ShortArray.contains(element: Short): Boolean {\n return indexOf(element) >= 0\n}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\npublic operator fun IntArray.contains(element: Int): Boolean {\n return indexOf(element) >= 0\n}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\npublic operator fun LongArray.contains(element: Long): Boolean {\n return indexOf(element) >= 0\n}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\n@Deprecated(\"The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'any { it == element }' instead to continue using this behavior, or '.asList().contains(element: T)' to get the same search behavior as in a list.\", ReplaceWith(\"any { it == element }\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\", hiddenSince = \"1.7\")\npublic operator fun FloatArray.contains(element: Float): Boolean {\n return any { it == element }\n}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\n@Deprecated(\"The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'any { it == element }' instead to continue using this behavior, or '.asList().contains(element: T)' to get the same search behavior as in a list.\", ReplaceWith(\"any { it == element }\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\", hiddenSince = \"1.7\")\npublic operator fun DoubleArray.contains(element: Double): Boolean {\n return any { it == element }\n}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\npublic operator fun BooleanArray.contains(element: Boolean): Boolean {\n return indexOf(element) >= 0\n}\n\n/**\n * Returns `true` if [element] is found in the array.\n */\npublic operator fun CharArray.contains(element: Char): Boolean {\n return indexOf(element) >= 0\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic expect fun <T> Array<out T>.elementAt(index:

Int): T\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic expect fun ByteArray.elementAt(index: Int):
Byte\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic expect fun ShortArray.elementAt(index: Int): Short\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic expect fun IntArray.elementAt(index: Int): Int\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic expect fun LongArray.elementAt(index: Int): Long\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*/\npublic expect fun FloatArray.elementAt(index: Int): Float\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic expect fun DoubleArray.elementAt(index: Int): Double\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic expect fun BooleanArray.elementAt(index: Int): Boolean\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic expect fun CharArray.elementAt(index: Int): Char\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function
if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\n   return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n */\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Byte): Byte {\n   return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Short): Short {\n   return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Int): Int {\n   return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.elementAtOrElse(index: Int, defaultValue:
(Int) -> Long): Long {\n   return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n */\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Float): Float {\n   return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample

samples.collections.Collections.Elements.elementAtOrElse\n */\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Double): Double {\n   return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n */\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Boolean): Boolean {\n   return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.elementAtOrElse(index: Int, defaultValue: (Int) -> Char): Char {\n   return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this
array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.elementAtOrNull(index: Int): T? {\n   return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.elementAtOrNull(index: Int): Byte? {\n   return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.elementAtOrNull(index: Int): Short? {\n   return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.elementAtOrNull(index: Int): Int? {\n   return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.elementAtOrNull(index: Int): Long? {\n   return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.elementAtOrNull(index: Int): Float? {\n   return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.elementAtOrNull(index: Int): Double? {\n   return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.elementAtOrNull(index: Int): Boolean? {\n   return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.elementAtOrNull(index: Int): Char? {\n   return this.getOrNull(index)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic
inline fun <T> Array<out T>.find(predicate: (T) -> Boolean): T? {\n   return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.find(predicate: (Byte) -> Boolean): Byte? {\n   return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun

ShortArray.find(predicate: (Short) -> Boolean): Short? {\n   return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.find(predicate: (Int) ->
Boolean): Int? {\n   return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.find(predicate: (Long) -> Boolean): Long? {\n   return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.find(predicate: (Float) -> Boolean): Float? {\n   return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.find(predicate: (Double) -> Boolean): Double? {\n   return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.find(predicate: (Boolean) -> Boolean): Boolean? {\n   return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.find(predicate: (Char) -> Boolean): Char? {\n   return firstOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.findLast(predicate: (T) -> Boolean): T? {\n   return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.findLast(predicate: (Byte) -> Boolean): Byte? {\n   return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.findLast(predicate: (Short) -> Boolean): Short? {\n   return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.findLast(predicate: (Int) -> Boolean): Int? {\n   return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate],
or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.findLast(predicate: (Long) -> Boolean): Long? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.findLast(predicate: (Float) -> Boolean): Float? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.findLast(predicate: (Double) -> Boolean): Double? {\n   return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was
found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.findLast(predicate: (Boolean) -> Boolean): Boolean? {\n   return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.findLast(predicate: (Char) -> Boolean): Char? {\n

return lastOrNull(predicate)\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun <T> Array<out T>.first(): T {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun ByteArray.first(): Byte {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun ShortArray.first(): Short {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun IntArray.first(): Int {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun LongArray.first(): Long {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun FloatArray.first(): Float {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun DoubleArray.first(): Double {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun BooleanArray.first(): Boolean {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[0]\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\npublic fun CharArray.first(): Char {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[0]\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */\npublic inline fun <T> Array<out T>.first(predicate: (T) -> Boolean): T {\n for (element in this) if (predicate(element)) return element\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */\npublic inline fun ByteArray.first(predicate: (Byte) -> Boolean): Byte {\n for (element in this) if (predicate(element)) return element\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */\npublic inline fun ShortArray.first(predicate: (Short) -> Boolean): Short {\n for (element in this) if (predicate(element)) return element\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException]
if no such element is found.\n */\npublic inline fun IntArray.first(predicate: (Int) -> Boolean): Int {\n for (element in this) if (predicate(element)) return element\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */\npublic inline fun LongArray.first(predicate: (Long) -> Boolean): Long {\n for (element in this) if (predicate(element)) return element\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */\npublic inline fun FloatArray.first(predicate: (Float) -> Boolean): Float {\n for (element in this) if (predicate(element)) return element\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n
* Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */\npublic inline fun DoubleArray.first(predicate: (Double) -> Boolean): Double {\n for (element in this) if (predicate(element)) return element\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */\npublic inline fun

BooleanArray.first(predicate: (Boolean) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return element\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */\npublic inline fun CharArray.first(predicate: (Char) -> Boolean): Char {\n for (element in this)
if (predicate(element)) return element\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to elements of this array in iteration order,\n * or throws [NoSuchElementException] if no non-null value was produced.\n * \n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any> Array<out T>.firstNotNullOf(transform: (T) -> R?): R {\n return firstNotNullOfOrNull(transform) ?: throw NoSuchElementException(\"No element of the array was transformed to a non-null value.\")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to elements of this array in iteration order,\n * or `null` if no non-null value was produced.\n * \n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic
inline fun <T, R : Any> Array<out T>.firstNotNullOfOrNull(transform: (T) -> R?): R? {\n for (element in this) {\n val result = transform(element)\n if (result != null) {\n return result\n }\n }\n return null\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun <T> Array<out T>.firstOrNull(): T? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun ByteArray.firstOrNull(): Byte? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun ShortArray.firstOrNull(): Short? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun IntArray.firstOrNull(): Int? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the
first element, or `null` if the array is empty.\n */\npublic fun LongArray.firstOrNull(): Long? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun FloatArray.firstOrNull(): Float? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun DoubleArray.firstOrNull(): Double? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun BooleanArray.firstOrNull(): Boolean? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\npublic fun CharArray.firstOrNull(): Char? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun <T> Array<out T>.firstOrNull(predicate:
(T) -> Boolean): T? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun ByteArray.firstOrNull(predicate: (Byte) -> Boolean): Byte? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun ShortArray.firstOrNull(predicate: (Short) -> Boolean): Short? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun IntArray.firstOrNull(predicate: (Int) -> Boolean): Int? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate],
or `null` if element was not found.\n */\npublic inline fun LongArray.firstOrNull(predicate: (Long) -> Boolean): Long? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun FloatArray.firstOrNull(predicate: (Float) -> Boolean): Float? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if

element was not found.\n */\npublic inline fun DoubleArray.firstOrNull(predicate: (Double) -> Boolean): Double? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun BooleanArray.firstOrNull(predicate: (Boolean) -> Boolean): Boolean? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun CharArray.firstOrNull(predicate: (Char) -> Boolean): Char? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.getOrElse(index: Int, defaultValue: (Int) -> T): T {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.getOrElse(index: Int, defaultValue: (Int) -> Byte): Byte {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.getOrElse(index: Int, defaultValue: (Int) -> Short): Short {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.getOrElse(index: Int, defaultValue: (Int) -> Int): Int {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.getOrElse(index: Int, defaultValue: (Int) -> Long): Long {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.getOrElse(index: Int, defaultValue: (Int) -> Float): Float {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.getOrElse(index: Int, defaultValue: (Int) -> Double): Double {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.getOrElse(index: Int, defaultValue: (Int) -> Boolean): Boolean {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.getOrElse(index: Int, defaultValue: (Int) -> Char): Char {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n */\npublic fun <T> Array<out T>.getOrNull(index: Int): T? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n */\npublic fun ByteArray.getOrNull(index: Int): Byte? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n */\npublic fun ShortArray.getOrNull(index: Int): Short? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n */\npublic fun IntArray.getOrNull(index: Int): Int? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given

[index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n */\npublic fun LongArray.getOrNull(index: Int): Long? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n */\npublic fun FloatArray.getOrNull(index: Int): Float? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n */\npublic fun DoubleArray.getOrNull(index: Int): Double? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n */\npublic fun BooleanArray.getOrNull(index: Int): Boolean? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n */\npublic fun CharArray.getOrNull(index: Int): Char? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\npublic fun <@kotlin.internal.OnlyInputTypes T> Array<out T>.indexOf(element: T): Int {\n if (element == null) {\n for (index in indices) {\n if (this[index] == null) {\n return index\n }\n }\n } else {\n for (index in indices) {\n if (element == this[index]) {\n return index\n }\n }\n }\n return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\npublic fun ByteArray.indexOf(element: Byte): Int {\n for (index in indices) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\npublic fun ShortArray.indexOf(element: Short): Int {\n for (index in indices) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\npublic fun IntArray.indexOf(element: Int): Int {\n for (index in indices) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\npublic fun LongArray.indexOf(element: Long): Int {\n for (index in indices) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\n@Deprecated(\"The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfFirst { it == element }' instead to continue using this behavior, or '.asList().indexOf(element: T)' to get the same search behavior as in a list.\", ReplaceWith(\"indexOfFirst { it == element }\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\", hiddenSince = \"1.7\")\npublic fun FloatArray.indexOf(element: Float): Int {\n for (index in indices) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\n@Deprecated(\"The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfFirst { it == element }' instead to continue using this behavior, or '.asList().indexOf(element: T)' to get the same search behavior as in a list.\", ReplaceWith(\"indexOfFirst { it == element }\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\", hiddenSince = \"1.7\")\npublic fun DoubleArray.indexOf(element: Double): Int {\n for (index in indices) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\npublic fun BooleanArray.indexOf(element: Boolean): Int {\n for (index in indices) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\npublic fun CharArray.indexOf(element: Char): Int {\n for (index in indices) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns

index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun <T> Array<out T>.indexOfFirst(predicate: (T) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun ByteArray.indexOfFirst(predicate: (Byte) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun ShortArray.indexOfFirst(predicate: (Short) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun IntArray.indexOfFirst(predicate: (Int) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun LongArray.indexOfFirst(predicate: (Long) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun FloatArray.indexOfFirst(predicate: (Float) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun DoubleArray.indexOfFirst(predicate: (Double) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun BooleanArray.indexOfFirst(predicate: (Boolean) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun CharArray.indexOfFirst(predicate: (Char) -> Boolean): Int {\n for (index in indices) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun <T> Array<out T>.indexOfLast(predicate: (T) -> Boolean): Int {\n for (index in indices.reversed()) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun ByteArray.indexOfLast(predicate: (Byte) -> Boolean): Int {\n for (index in indices.reversed()) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun ShortArray.indexOfLast(predicate: (Short) -> Boolean): Int {\n for (index in indices.reversed()) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun IntArray.indexOfLast(predicate: (Int) -> Boolean): Int {\n for (index in indices.reversed()) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun LongArray.indexOfLast(predicate: (Long) -> Boolean): Int {\n for (index in indices.reversed()) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun FloatArray.indexOfLast(predicate: (Float) -> Boolean): Int {\n for (index in indices.reversed()) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun

DoubleArray.indexOfLast(predicate: (Double) -> Boolean): Int {\n for (index in indices.reversed()) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.\n */\npublic inline fun BooleanArray.indexOfLast(predicate: (Boolean) -> Boolean): Int {\n for (index in indices.reversed()) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or
-1 if the array does not contain such element.\n */\npublic inline fun CharArray.indexOfLast(predicate: (Char) -> Boolean): Int {\n for (index in indices.reversed()) {\n if (predicate(this[index])) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun <T> Array<out T>.last(): T {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun ByteArray.last(): Byte {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is
empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun ShortArray.last(): Short {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun IntArray.last(): Int {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun LongArray.last(): Long {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
*/\npublic fun FloatArray.last(): Float {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun DoubleArray.last(): Double {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun BooleanArray.last(): Boolean {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[lastIndex]\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun CharArray.last():
Char {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return this[lastIndex]\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun <T> Array<out T>.last(predicate: (T) -> Boolean): T {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun ByteArray.last(predicate: (Byte) -> Boolean): Byte {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if
(predicate(element)) return element\n }\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample

samples.collections.Collections.Elements.last\n */\npublic inline fun ShortArray.last(predicate: (Short) -> Boolean): Short {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun IntArray.last(predicate: (Int) -> Boolean): Int {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun LongArray.last(predicate: (Long) -> Boolean): Long {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun FloatArray.last(predicate: (Float) -> Boolean): Float {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun DoubleArray.last(predicate: (Double) -> Boolean): Double {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun BooleanArray.last(predicate: (Boolean) -> Boolean): Boolean {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun CharArray.last(predicate: (Char) -> Boolean): Char {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\npublic fun <@kotlin.internal.OnlyInputTypes T> Array<out T>.lastIndexOf(element: T): Int {\n if (element == null) {\n for (index in indices.reversed()) {\n if (this[index] == null) {\n return index\n }\n }\n } else {\n for (index in indices.reversed()) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\npublic fun ByteArray.lastIndexOf(element: Byte): Int {\n for (index in indices.reversed()) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\npublic fun ShortArray.lastIndexOf(element: Short): Int {\n for (index in indices.reversed()) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\npublic fun IntArray.lastIndexOf(element: Int): Int {\n for (index in indices.reversed()) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\npublic fun LongArray.lastIndexOf(element: Long): Int {\n for (index in indices.reversed()) {\n if (element == this[index]) {\n return index\n

}\n }\n return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\n@Deprecated(\"The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfLast { it == element }' instead to continue using this behavior, or '.asList().lastIndexOf(element: T)' to get the same search behavior as in a list.\", ReplaceWith(\"indexOfLast { it == element }\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\", hiddenSince = \"1.7\")\npublic fun FloatArray.lastIndexOf(element: Float): Int {\n for (index in indices.reversed())
{\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\n@Deprecated(\"The function has unclear behavior when searching for NaN or zero values and will be removed soon. Use 'indexOfLast { it == element }' instead to continue using this behavior, or '.asList().lastIndexOf(element: T)' to get the same search behavior as in a list.\", ReplaceWith(\"indexOfLast { it == element }\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\", hiddenSince = \"1.7\")\npublic fun DoubleArray.lastIndexOf(element: Double): Int {\n for (index in indices.reversed()) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\npublic fun BooleanArray.lastIndexOf(element: Boolean): Int {\n for (index in indices.reversed())
{\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\npublic fun CharArray.lastIndexOf(element: Char): Int {\n for (index in indices.reversed()) {\n if (element == this[index]) {\n return index\n }\n }\n return -1\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun <T> Array<out T>.lastOrNull(): T? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun ByteArray.lastOrNull(): Byte? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
*/\npublic fun ShortArray.lastOrNull(): Short? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun IntArray.lastOrNull(): Int? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun LongArray.lastOrNull(): Long? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun FloatArray.lastOrNull(): Float? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun DoubleArray.lastOrNull():
Double? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun BooleanArray.lastOrNull(): Boolean? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun CharArray.lastOrNull(): Char? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun <T> Array<out T>.lastOrNull(predicate: (T) -> Boolean): T? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last
element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun ByteArray.lastOrNull(predicate: (Byte) -> Boolean): Byte? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun ShortArray.lastOrNull(predicate: (Short) ->

Boolean): Short? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun IntArray.lastOrNull(predicate: (Int) -> Boolean): Int? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun LongArray.lastOrNull(predicate: (Long) -> Boolean): Long? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun FloatArray.lastOrNull(predicate: (Float) -> Boolean): Float? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun DoubleArray.lastOrNull(predicate: (Double) -> Boolean): Double? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun BooleanArray.lastOrNull(predicate: (Boolean) -> Boolean): Boolean? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun CharArray.lastOrNull(predicate: (Char) -> Boolean): Char? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.random(): T {\n return random(Random)\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.random(): Byte {\n return random(Random)\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.random(): Short {\n return random(Random)\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.random(): Int {\n return random(Random)\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.random(): Long {\n return random(Random)\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.random(): Float {\n return random(Random)\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.random(): Double {\n return random(Random)\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.random(): Boolean {\n return random(Random)\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n

*/\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.random(): Char {\n return random(Random)\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\npublic fun <T> Array<out T>.random(random: Random): T {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\npublic fun ByteArray.random(random: Random): Byte {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\npublic fun ShortArray.random(random: Random): Short {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\npublic fun IntArray.random(random: Random): Int {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns
a random element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\npublic fun LongArray.random(random: Random): Long {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\npublic fun FloatArray.random(random: Random): Float {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\npublic fun DoubleArray.random(random: Random): Double {\n if (isEmpty())\n throw NoSuchElementException(\"Array
is empty.\")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\npublic fun BooleanArray.random(random: Random): Boolean {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\npublic fun CharArray.random(random: Random): Char {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline
fun <T> Array<out T>.randomOrNull(): T? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.randomOrNull(): Byte? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.randomOrNull(): Short? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.randomOrNull(): Int? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is

empty.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun LongArray.randomOrNull(): Long? {\n   return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun FloatArray.randomOrNull(): Float? {\n   return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun DoubleArray.randomOrNull(): Double? {\n   return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array, or `null` if this array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun BooleanArray.randomOrNull():
Boolean? {\n   return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if
this array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun CharArray.randomOrNull(): Char? {\n   return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this array using the specified source of randomness, or `null` if this array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T> Array<out
T>.randomOrNull(random: Random): T? {\n   if (isEmpty())\n   return null\n   return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
ByteArray.randomOrNull(random: Random): Byte? {\n   if (isEmpty())\n   return null\n   return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
ShortArray.randomOrNull(random: Random): Short? {\n   if (isEmpty())\n   return null\n   return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
IntArray.randomOrNull(random: Random): Int? {\n   if (isEmpty())\n   return null\n   return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
LongArray.randomOrNull(random: Random): Long? {\n   if (isEmpty())\n
        return null\n   return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using
the specified source of randomness, or `null` if this array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
FloatArray.randomOrNull(random: Random): Float? {\n   if (isEmpty())\n   return null\n   return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
DoubleArray.randomOrNull(random: Random): Double? {\n   if (isEmpty())\n   return null\n   return
get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of
randomness, or `null` if this array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun
BooleanArray.randomOrNull(random:

Random): Boolean? {\n if (isEmpty())\n return null\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun CharArray.randomOrNull(random: Random): Char? {\n if (isEmpty())\n return null\n return get(random.nextInt(size))\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or has more than one element.\n */\npublic fun <T> Array<out T>.single(): T {\n return when (size) {\n 0 -> throw NoSuchElementException(\"Array is empty.\")\n 1 -> this[0]\n else -> throw IllegalArgumentException(\"Array has more than one element.\")\n }\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or has more than one element.\n */\npublic fun ByteArray.single(): Byte {\n
return when (size) {\n 0 -> throw NoSuchElementException(\"Array is empty.\")\n 1 -> this[0]\n else -> throw IllegalArgumentException(\"Array has more than one element.\")\n }\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or has more than one element.\n */\npublic fun ShortArray.single(): Short {\n return when (size) {\n 0 -> throw NoSuchElementException(\"Array is empty.\")\n 1 -> this[0]\n else -> throw IllegalArgumentException(\"Array has more than one element.\")\n }\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or has more than one element.\n */\npublic fun IntArray.single(): Int {\n return when (size) {\n 0 -> throw NoSuchElementException(\"Array is empty.\")\n 1 -> this[0]\n else -> throw IllegalArgumentException(\"Array has more than one element.\")\n }\n}\n\n/**\n * Returns the single element, or throws an exception if
the array is empty or has more than one element.\n */\npublic fun LongArray.single(): Long {\n return when (size) {\n 0 -> throw NoSuchElementException(\"Array is empty.\")\n 1 -> this[0]\n else -> throw IllegalArgumentException(\"Array has more than one element.\")\n }\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or has more than one element.\n */\npublic fun FloatArray.single(): Float {\n return when (size) {\n 0 -> throw NoSuchElementException(\"Array is empty.\")\n 1 -> this[0]\n else -> throw IllegalArgumentException(\"Array has more than one element.\")\n }\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or has more than one element.\n */\npublic fun DoubleArray.single(): Double {\n return when (size) {\n 0 -> throw NoSuchElementException(\"Array is empty.\")\n 1 -> this[0]\n else -> throw IllegalArgumentException(\"Array
has more than one element.\")\n }\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or has more than one element.\n */\npublic fun BooleanArray.single(): Boolean {\n return when (size) {\n 0 -> throw NoSuchElementException(\"Array is empty.\")\n 1 -> this[0]\n else -> throw IllegalArgumentException(\"Array has more than one element.\")\n }\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or has more than one element.\n */\npublic fun CharArray.single(): Char {\n return when (size) {\n 0 -> throw NoSuchElementException(\"Array is empty.\")\n 1 -> this[0]\n else -> throw IllegalArgumentException(\"Array has more than one element.\")\n }\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\npublic inline fun <T> Array<out T>.single(predicate: (T) -> Boolean): T {\n
var single: T? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if (found) throw IllegalArgumentException(\"Array contains more than one matching element.\")\n single = element\n found = true\n }\n }\n if (!found) throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n @Suppress(\"UNCHECKED_CAST\")\n return single as T\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\npublic inline fun ByteArray.single(predicate: (Byte) -> Boolean): Byte {\n var single: Byte? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if (found) throw IllegalArgumentException(\"Array contains more than one matching element.\")\n single = element\n found = true\n }\n }\n if (!found) throw NoSuchElementException(\"Array

```
contains no element matching the predicate.\")\n    @Suppress(\"UNCHECKED_CAST\")\n    return single as Byte\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\npublic inline fun ShortArray.single(predicate: (Short) -> Boolean): Short {\n    var single: Short? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException(\"Array contains more than one matching element.\")\n            single = element\n            found = true\n        }\n    }\n    if (!found) throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n    @Suppress(\"UNCHECKED_CAST\")\n    return single as Short\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\npublic inline fun IntArray.single(predicate:
    (Int) -> Boolean): Int {\n    var single: Int? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException(\"Array contains more than one matching element.\")\n            single = element\n            found = true\n        }\n    }\n    if (!found) throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n    @Suppress(\"UNCHECKED_CAST\")\n    return single as Int\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\npublic inline fun LongArray.single(predicate: (Long) -> Boolean): Long {\n    var single: Long? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException(\"Array contains more than one matching element.\")\n            single = element\n            found = true\n        }\n    }\n    if (!found) throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n    @Suppress(\"UNCHECKED_CAST\")\n    return single as Long\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\npublic inline fun FloatArray.single(predicate: (Float) -> Boolean): Float {\n    var single: Float? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException(\"Array contains more than one matching element.\")\n            single = element\n            found = true\n        }\n    }\n    if (!found) throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n    @Suppress(\"UNCHECKED_CAST\")\n    return single as Float\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching
    element.\n */\npublic inline fun DoubleArray.single(predicate: (Double) -> Boolean): Double {\n    var single: Double? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException(\"Array contains more than one matching element.\")\n            single = element\n            found = true\n        }\n    }\n    if (!found) throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n    @Suppress(\"UNCHECKED_CAST\")\n    return single as Double\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\npublic inline fun BooleanArray.single(predicate: (Boolean) -> Boolean): Boolean {\n    var single: Boolean? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException(\"Array contains more than one
    matching element.\")\n            single = element\n            found = true\n        }\n    }\n    if (!found) throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n    @Suppress(\"UNCHECKED_CAST\")\n    return single as Boolean\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\npublic inline fun CharArray.single(predicate: (Char) -> Boolean): Char {\n    var single: Char? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException(\"Array contains more than one matching element.\")\n            single = element\n            found = true\n        }\n    }\n    if (!found) throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n    @Suppress(\"UNCHECKED_CAST\")\n    return single as Char\n}\n\n/**\n * Returns single element, or `null` if the
    array is empty or has more than one element.\n */\npublic fun <T> Array<out T>.singleOrNull(): T? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one
```

```
element.\n */\npublic fun ByteArray.singleOrNull(): Byte? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n */\npublic fun ShortArray.singleOrNull(): Short? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n */\npublic fun IntArray.singleOrNull(): Int? {\n return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n */\npublic fun LongArray.singleOrNull(): Long? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty
 or has more than one element.\n */\npublic fun FloatArray.singleOrNull(): Float? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n */\npublic fun DoubleArray.singleOrNull(): Double? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n */\npublic fun BooleanArray.singleOrNull(): Boolean? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n */\npublic fun CharArray.singleOrNull(): Char? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic inline fun <T> Array<out T>.singleOrNull(predicate: (T) -> Boolean): T? {\n    var single: T? = null\n    var found
 = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic inline fun ByteArray.singleOrNull(predicate: (Byte) -> Boolean): Byte? {\n    var single: Byte? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic inline fun ShortArray.singleOrNull(predicate: (Short) -> Boolean): Short? {\n    var single: Short? = null\n
    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic inline fun IntArray.singleOrNull(predicate: (Int) -> Boolean): Int? {\n    var single: Int? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic inline fun LongArray.singleOrNull(predicate: (Long) -> Boolean): Long? {\n    var single: Long? =
 null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic inline fun FloatArray.singleOrNull(predicate: (Float) -> Boolean): Float? {\n    var single: Float? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic inline fun DoubleArray.singleOrNull(predicate: (Double) ->
 Boolean): Double? {\n
 var single: Double? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic inline fun BooleanArray.singleOrNull(predicate: (Boolean) -> Boolean): Boolean? {\n    var single: Boolean? = null\n    var found = false\n    for (element in this) {\n        if
```

(predicate(element)) {\n if (found) return null\n single = element\n found = true\n }\n }\n if (!found) return null\n return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic inline fun CharArray.singleOrNull(predicate: (Char) -> Boolean): Char? {\n var single: Char? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if (found) return null\n single = element\n found = true\n }\n }\n if (!found) return null\n return single\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun <T> Array<out T>.drop(n: Int): List<T> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun ByteArray.drop(n: Int): List<Byte> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun ShortArray.drop(n: Int): List<Short> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun IntArray.drop(n: Int): List<Int> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun LongArray.drop(n: Int): List<Long> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun FloatArray.drop(n: Int): List<Float> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun DoubleArray.drop(n: Int): List<Double> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun BooleanArray.drop(n: Int): List<Boolean> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun CharArray.drop(n: Int): List<Char> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun <T> Array<out T>.dropLast(n: Int): List<T> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample

samples.collections.Collections.Transformations.drop\n */\npublic fun ByteArray.dropLast(n: Int): List<Byte> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun ShortArray.dropLast(n: Int): List<Short> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun IntArray.dropLast(n: Int): List<Int> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun LongArray.dropLast(n: Int): List<Long> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun FloatArray.dropLast(n: Int): List<Float> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun DoubleArray.dropLast(n: Int): List<Double> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun BooleanArray.dropLast(n: Int): List<Boolean> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun CharArray.dropLast(n: Int): List<Char> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun <T> Array<out T>.dropLastWhile(predicate: (T) -> Boolean): List<T> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun ByteArray.dropLastWhile(predicate: (Byte) -> Boolean): List<Byte> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun ShortArray.dropLastWhile(predicate: (Short) -> Boolean): List<Short> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun IntArray.dropLastWhile(predicate: (Int) -> Boolean): List<Int> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun LongArray.dropLastWhile(predicate:

(Long) -> Boolean): List<Long> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun FloatArray.dropLastWhile(predicate: (Float) -> Boolean): List<Float> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except
last elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun DoubleArray.dropLastWhile(predicate: (Double) -> Boolean): List<Double> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun BooleanArray.dropLastWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun CharArray.dropLastWhile(predicate: (Char) -> Boolean): List<Char> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return take(index + 1)\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun <T> Array<out T>.dropWhile(predicate: (T) -> Boolean): List<T> {\n var yielding = false\n val list = ArrayList<T>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun ByteArray.dropWhile(predicate:
(Byte) -> Boolean): List<Byte> {\n var yielding = false\n val list = ArrayList<Byte>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun ShortArray.dropWhile(predicate: (Short) -> Boolean): List<Short> {\n var yielding = false\n val list = ArrayList<Short>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun IntArray.dropWhile(predicate: (Int) -> Boolean): List<Int> {\n var yielding = false\n val list = ArrayList<Int>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun LongArray.dropWhile(predicate: (Long) -> Boolean): List<Long> {\n var yielding = false\n val list = ArrayList<Long>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n *
\n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun FloatArray.dropWhile(predicate: (Float) -> Boolean): List<Float> {\n var yielding = false\n val list = ArrayList<Float>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing all

elements except first elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun DoubleArray.dropWhile(predicate: (Double) -> Boolean): List<Double> {\n var yielding = false\n val list = ArrayList<Double>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun BooleanArray.dropWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n var yielding = false\n val list = ArrayList<Boolean>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun CharArray.dropWhile(predicate: (Char) -> Boolean): List<Char> {\n var yielding = false\n val list = ArrayList<Char>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun <T> Array<out T>.filter(predicate: (T) -> Boolean): List<T> {\n return filterTo(ArrayList<T>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun ByteArray.filter(predicate: (Byte) -> Boolean): List<Byte> {\n return filterTo(ArrayList<Byte>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun ShortArray.filter(predicate: (Short) -> Boolean): List<Short> {\n return filterTo(ArrayList<Short>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun IntArray.filter(predicate: (Int) -> Boolean): List<Int> {\n return filterTo(ArrayList<Int>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun LongArray.filter(predicate: (Long) -> Boolean): List<Long> {\n return filterTo(ArrayList<Long>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun FloatArray.filter(predicate: (Float) -> Boolean): List<Float> {\n return filterTo(ArrayList<Float>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun DoubleArray.filter(predicate: (Double) -> Boolean): List<Double> {\n return filterTo(ArrayList<Double>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun BooleanArray.filter(predicate: (Boolean) -> Boolean): List<Boolean> {\n return filterTo(ArrayList<Boolean>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun CharArray.filter(predicate: (Char) -> Boolean): List<Char> {\n return filterTo(ArrayList<Char>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun <T> Array<out T>.filterIndexed(predicate:
(index: Int, T) -> Boolean): List<T> {\n return filterIndexedTo(ArrayList<T>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun ByteArray.filterIndexed(predicate: (index: Int, Byte) -> Boolean): List<Byte> {\n return

filterIndexedTo(ArrayList<Byte>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun ShortArray.filterIndexed(predicate: (index: Int, Short) -> Boolean): List<Short>
{\n return filterIndexedTo(ArrayList<Short>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun IntArray.filterIndexed(predicate: (index: Int, Int) -> Boolean): List<Int> {\n return filterIndexedTo(ArrayList<Int>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun LongArray.filterIndexed(predicate: (index: Int, Long) -> Boolean): List<Long> {\n return filterIndexedTo(ArrayList<Long>(),
predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun FloatArray.filterIndexed(predicate: (index: Int, Float) -> Boolean): List<Float> {\n return filterIndexedTo(ArrayList<Float>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun DoubleArray.filterIndexed(predicate: (index: Int, Double) -> Boolean): List<Double> {\n return filterIndexedTo(ArrayList<Double>(), predicate)\n}\n\n/**\n
* Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun BooleanArray.filterIndexed(predicate: (index: Int, Boolean) -> Boolean): List<Boolean> {\n return filterIndexedTo(ArrayList<Boolean>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun CharArray.filterIndexed(predicate: (index: Int, Char) -> Boolean): List<Char> {\n return filterIndexedTo(ArrayList<Char>(), predicate)\n}\n\n/**\n * Appends all elements matching
the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <T, C : MutableCollection<in T>> Array<out T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C {\n forEachIndexed { index, element ->\n if (predicate(index, element)) destination.add(element)\n }\n return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Byte>> ByteArray.filterIndexedTo(destination: C, predicate:
(index: Int, Byte) -> Boolean): C {\n forEachIndexed { index, element ->\n if (predicate(index, element)) destination.add(element)\n }\n return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample

samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Short>> ShortArray.filterIndexedTo(destination: C, predicate: (index: Int, Short) -> Boolean): C {\n   forEachIndexed { index, element ->\n   if (predicate(index, element)) destination.add(element)\n   }\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate
 evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Int>> IntArray.filterIndexedTo(destination: C, predicate: (index: Int, Int) -> Boolean): C {\n   forEachIndexed { index, element ->\n   if (predicate(index, element)) destination.add(element)\n   }\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Long>> LongArray.filterIndexedTo(destination: C, predicate: (index: Int, Long) -> Boolean): C {\n   forEachIndexed { index, element ->\n   if (predicate(index, element)) destination.add(element)\n   }\n   return destination\n}\n\n/**\n
 * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Float>> FloatArray.filterIndexedTo(destination: C, predicate: (index: Int, Float) -> Boolean): C {\n   forEachIndexed { index, element ->\n   if (predicate(index, element)) destination.add(element)\n   }\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Double>> DoubleArray.filterIndexedTo(destination:
 C, predicate: (index: Int, Double) -> Boolean): C {\n   forEachIndexed { index, element ->\n   if (predicate(index, element)) destination.add(element)\n   }\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Boolean>> BooleanArray.filterIndexedTo(destination: C, predicate: (index: Int, Boolean) -> Boolean): C {\n   forEachIndexed { index, element ->\n   if (predicate(index, element)) destination.add(element)\n   }\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and
 the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C : MutableCollection<in Char>> CharArray.filterIndexedTo(destination: C, predicate: (index: Int, Char) -> Boolean): C {\n   forEachIndexed { index, element ->\n   if (predicate(index, element)) destination.add(element)\n   }\n   return destination\n}\n\n/**\n * Returns a list containing all elements that are instances of specified type parameter R.\n * \n * @sample samples.collections.Collections.Filtering.filterIsInstance\n */\npublic inline fun <reified R> Array<*>.filterIsInstance(): List<@kotlin.internal.NoInfer R> {\n   return filterIsInstanceTo(ArrayList<R>())\n}\n\n/**\n * Appends all elements that are instances of specified type parameter R to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterIsInstanceTo\n */\npublic inline fun <reified R, C
 : MutableCollection<in R>> Array<*>.filterIsInstanceTo(destination: C): C {\n   for (element in this) if (element is R) destination.add(element)\n   return destination\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun <T> Array<out T>.filterNot(predicate: (T) -> Boolean): List<T> {\n   return filterNotTo(ArrayList<T>(),

predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun ByteArray.filterNot(predicate: (Byte) -> Boolean): List<Byte> {\n return filterNotTo(ArrayList<Byte>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun ShortArray.filterNot(predicate: (Short) -> Boolean): List<Short>
{\n return filterNotTo(ArrayList<Short>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun IntArray.filterNot(predicate: (Int) -> Boolean): List<Int> {\n return filterNotTo(ArrayList<Int>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun LongArray.filterNot(predicate: (Long) -> Boolean): List<Long> {\n return filterNotTo(ArrayList<Long>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun FloatArray.filterNot(predicate: (Float) -> Boolean): List<Float> {\n return filterNotTo(ArrayList<Float>(), predicate)\n}\n\n/**\n * Returns a list containing all elements
not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun DoubleArray.filterNot(predicate: (Double) -> Boolean): List<Double> {\n return filterNotTo(ArrayList<Double>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun BooleanArray.filterNot(predicate: (Boolean) -> Boolean): List<Boolean> {\n return filterNotTo(ArrayList<Boolean>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun CharArray.filterNot(predicate: (Char) -> Boolean): List<Char> {\n return filterNotTo(ArrayList<Char>(), predicate)\n}\n\n/**\n * Returns a list containing all elements that are not `null`.\n * \n * @sample samples.collections.Collections.Filtering.filterNotNull\n
*/\npublic fun <T : Any> Array<out T?>.filterNotNull(): List<T> {\n return filterNotNullTo(ArrayList<T>())\n}\n\n/**\n * Appends all elements that are not `null` to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterNotNullTo\n */\npublic fun <C : MutableCollection<in T>, T : Any> Array<out T?>.filterNotNullTo(destination: C): C {\n for (element in this) if (element != null) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <T, C : MutableCollection<in T>> Array<out T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {\n for (element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*/\npublic inline fun <C : MutableCollection<in Byte>> ByteArray.filterNotTo(destination: C, predicate: (Byte) -> Boolean): C {\n for (element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Short>> ShortArray.filterNotTo(destination: C, predicate: (Short) -> Boolean): C {\n for (element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Int>> IntArray.filterNotTo(destination: C, predicate: (Int) -> Boolean): C {\n for (element in this)
if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Long>> LongArray.filterNotTo(destination: C, predicate: (Long) -> Boolean): C {\n for (element in this) if

(!predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Float>> FloatArray.filterNotTo(destination: C, predicate: (Float) -> Boolean): C {\n   for (element in this) if (!predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Double>> DoubleArray.filterNotTo(destination: C, predicate: (Double) -> Boolean): C {\n   for (element in this) if (!predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Boolean>> BooleanArray.filterNotTo(destination: C, predicate: (Boolean) -> Boolean): C {\n   for (element in this) if (!predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Char>> CharArray.filterNotTo(destination: C, predicate: (Char) -> Boolean): C {\n   for (element in this) if (!predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <T, C : MutableCollection<in T>> Array<out T>.filterTo(destination: C, predicate: (T) -> Boolean): C {\n   for (element in this) if (predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Byte>> ByteArray.filterTo(destination: C, predicate: (Byte) -> Boolean): C {\n   for (element in this) if (predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Short>> ShortArray.filterTo(destination: C, predicate: (Short) -> Boolean): C {\n   for (element in this) if (predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Int>> IntArray.filterTo(destination: C, predicate: (Int) -> Boolean): C {\n   for (element in this) if (predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Long>> LongArray.filterTo(destination: C, predicate: (Long) -> Boolean): C {\n   for (element in this) if (predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Float>> FloatArray.filterTo(destination: C, predicate: (Float) -> Boolean): C {\n   for (element in this) if (predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Double>> DoubleArray.filterTo(destination: C, predicate: (Double) -> Boolean): C {\n   for (element in this) if (predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Boolean>> BooleanArray.filterTo(destination: C, predicate: (Boolean) -> Boolean): C {\n   for (element in this) if (predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : MutableCollection<in Char>>

CharArray.filterTo(destination: C, predicate: (Char) -> Boolean): C {\n for (element in this) if (predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n */\npublic fun <T> Array<out T>.slice(indices: IntRange): List<T> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n */\npublic fun ByteArray.slice(indices:
IntRange): List<Byte> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n */\npublic fun ShortArray.slice(indices: IntRange): List<Short> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n */\npublic fun IntArray.slice(indices: IntRange): List<Int> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n */\npublic fun LongArray.slice(indices: IntRange): List<Long> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n
* Returns a list containing elements at indices in the specified [indices] range.\n */\npublic fun FloatArray.slice(indices: IntRange): List<Float> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n */\npublic fun DoubleArray.slice(indices: IntRange): List<Double> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n */\npublic fun BooleanArray.slice(indices: IntRange): List<Boolean> {\n if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n */\npublic fun CharArray.slice(indices: IntRange): List<Char> {\n
if (indices.isEmpty()) return listOf()\n return copyOfRange(indices.start, indices.endInclusive + 1).asList()\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun <T> Array<out T>.slice(indices: Iterable<Int>): List<T> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list = ArrayList<T>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun ByteArray.slice(indices: Iterable<Int>): List<Byte> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list = ArrayList<Byte>(size)\n for (index in indices) {\n list.add(get(index))\n
}\n return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun ShortArray.slice(indices: Iterable<Int>): List<Short> {\n val size = indices.collectionSizeOrDefault(10)\n
if (size == 0) return emptyList()\n val list = ArrayList<Short>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun IntArray.slice(indices: Iterable<Int>): List<Int> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list = ArrayList<Int>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun LongArray.slice(indices: Iterable<Int>): List<Long> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list = ArrayList<Long>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun FloatArray.slice(indices:
Iterable<Int>): List<Float> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list = ArrayList<Float>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun DoubleArray.slice(indices: Iterable<Int>): List<Double> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list = ArrayList<Double>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n

*/\npublic fun BooleanArray.slice(indices: Iterable<Int>): List<Boolean> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list = ArrayList<Boolean>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n */\npublic fun CharArray.slice(indices: Iterable<Int>): List<Char> {\n val size = indices.collectionSizeOrDefault(10)\n if (size == 0) return emptyList()\n val list = ArrayList<Char>(size)\n for (index in indices) {\n list.add(get(index))\n }\n return list\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun <T> Array<T>.sliceArray(indices: Collection<Int>): Array<T> {\n val result = arrayOfNulls(this, indices.size)\n var targetIndex = 0\n for (sourceIndex in indices) {\n result[targetIndex++] = this[sourceIndex]\n }\n return result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun ByteArray.sliceArray(indices: Collection<Int>): ByteArray {\n val result = ByteArray(indices.size)\n var targetIndex = 0\n for (sourceIndex in indices) {\n result[targetIndex++]
= this[sourceIndex]\n }\n return result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun ShortArray.sliceArray(indices: Collection<Int>): ShortArray {\n val result = ShortArray(indices.size)\n var targetIndex = 0\n for (sourceIndex in indices) {\n result[targetIndex++] = this[sourceIndex]\n }\n return result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun IntArray.sliceArray(indices: Collection<Int>): IntArray {\n val result = IntArray(indices.size)\n var targetIndex = 0\n for (sourceIndex in indices) {\n result[targetIndex++] = this[sourceIndex]\n }\n return result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun LongArray.sliceArray(indices: Collection<Int>): LongArray {\n val result = LongArray(indices.size)\n var targetIndex = 0\n for (sourceIndex in indices)
{\n result[targetIndex++] = this[sourceIndex]\n }\n return result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun FloatArray.sliceArray(indices: Collection<Int>): FloatArray {\n val result = FloatArray(indices.size)\n var targetIndex = 0\n for (sourceIndex in indices) {\n result[targetIndex++] = this[sourceIndex]\n }\n return result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun DoubleArray.sliceArray(indices: Collection<Int>): DoubleArray {\n val result = DoubleArray(indices.size)\n var targetIndex = 0\n for (sourceIndex in indices) {\n result[targetIndex++] = this[sourceIndex]\n }\n return result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun BooleanArray.sliceArray(indices: Collection<Int>): BooleanArray {\n val result = BooleanArray(indices.size)\n
var targetIndex = 0\n for (sourceIndex in indices) {\n result[targetIndex++] = this[sourceIndex]\n }\n return result\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\npublic fun CharArray.sliceArray(indices: Collection<Int>): CharArray {\n val result = CharArray(indices.size)\n var targetIndex = 0\n for (sourceIndex in indices) {\n result[targetIndex++] = this[sourceIndex]\n }\n return result\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\npublic fun <T> Array<T>.sliceArray(indices: IntRange): Array<T> {\n if (indices.isEmpty()) return copyOfRange(0, 0)\n return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\npublic fun ByteArray.sliceArray(indices: IntRange): ByteArray {\n if (indices.isEmpty()) return ByteArray(0)\n return
copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\npublic fun ShortArray.sliceArray(indices: IntRange): ShortArray {\n if (indices.isEmpty()) return ShortArray(0)\n return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\npublic fun IntArray.sliceArray(indices: IntRange): IntArray {\n if (indices.isEmpty()) return IntArray(0)\n return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\npublic fun LongArray.sliceArray(indices: IntRange): LongArray {\n if (indices.isEmpty()) return LongArray(0)\n return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\npublic

```
fun FloatArray.sliceArray(indices: IntRange): FloatArray {\n    if (indices.isEmpty()) return FloatArray(0)\n    return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\npublic fun DoubleArray.sliceArray(indices: IntRange): DoubleArray {\n    if (indices.isEmpty()) return DoubleArray(0)\n    return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\npublic fun BooleanArray.sliceArray(indices: IntRange): BooleanArray {\n    if (indices.isEmpty()) return BooleanArray(0)\n    return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\npublic fun CharArray.sliceArray(indices: IntRange): CharArray {\n    if (indices.isEmpty()) return CharArray(0)\n    return copyOfRange(indices.start, indices.endInclusive + 1)\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun <T> Array<out T>.take(n: Int): List<T> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<T>(n)\n    for (item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun ByteArray.take(n: Int): List<Byte> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<Byte>(n)\n    for (item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun ShortArray.take(n: Int): List<Short> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<Short>(n)\n    for (item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun IntArray.take(n: Int): List<Int> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<Int>(n)\n    for (item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun LongArray.take(n: Int): List<Long> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<Long>(n)\n    for (item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun FloatArray.take(n: Int): List<Float> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<Float>(n)\n    for (item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun DoubleArray.take(n: Int): List<Double> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<Double>(n)\n    for (item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws
```

```
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun BooleanArray.take(n: Int): List<Boolean> {\n
  require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    if (n >=
size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n    val list = ArrayList<Boolean>(n)\n
for (item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n    }\n    return list\n}\n\n/**\n *
Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException
 if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun
CharArray.take(n: Int): List<Char> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if
(n == 0) return emptyList()\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[0])\n    var count = 0\n
val list = ArrayList<Char>(n)\n    for (item in this) {\n        list.add(item)\n        if (++count == n)\n            break\n
}\n    return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun <T> Array<out T>.takeLast(n: Int): List<T>
{\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    val
size = size\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[size - 1])\n    val list =
ArrayList<T>(n)\n
    for (index in size - n until size)\n        list.add(this[index])\n    return list\n}\n\n/**\n * Returns a list containing
last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun ByteArray.takeLast(n: Int): List<Byte> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    val size =
size\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[size - 1])\n    val list = ArrayList<Byte>(n)\n
for (index in size - n until size)\n        list.add(this[index])\n    return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun ShortArray.takeLast(n: Int): List<Short> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\"
 }\n    if (n == 0) return emptyList()\n    val size = size\n    if (n >= size) return toList()\n    if (n == 1) return
listOf(this[size - 1])\n    val list = ArrayList<Short>(n)\n    for (index in size - n until size)\n
list.add(this[index])\n    return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun IntArray.takeLast(n: Int): List<Int> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    val size =
size\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[size - 1])\n    val list = ArrayList<Int>(n)\n
for (index in size - n until size)\n        list.add(this[index])\n    return list\n}\n\n/**\n * Returns a list containing last
[n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n
 */\npublic fun LongArray.takeLast(n: Int): List<Long> {\n    require(n >= 0) { \"Requested element count $n is less
than zero.\" }\n    if (n == 0) return emptyList()\n    val size = size\n    if (n >= size) return toList()\n    if (n == 1)
return listOf(this[size - 1])\n    val list = ArrayList<Long>(n)\n    for (index in size - n until size)\n
list.add(this[index])\n    return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun FloatArray.takeLast(n: Int): List<Float> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return emptyList()\n    val size =
size\n    if (n >= size) return toList()\n    if (n == 1) return listOf(this[size - 1])\n    val list = ArrayList<Float>(n)\n
for (index in size - n until size)\n        list.add(this[index])\n
    return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if
[n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun
DoubleArray.takeLast(n: Int): List<Double> {\n    require(n >= 0) { \"Requested element count $n is less than
zero.\" }\n    if (n == 0) return emptyList()\n    val size = size\n    if (n >= size) return toList()\n    if (n == 1) return
```

listOf(this[size - 1])\n   val list = ArrayList<Double>(n)\n   for (index in size - n until size)\n list.add(this[index])\n   return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun BooleanArray.takeLast(n: Int): List<Boolean> {\n   require(n >= 0) { \"Requested element count $n is less than zero.\" }\n   if (n == 0) return emptyList()\n   val
 size = size\n   if (n >= size) return toList()\n   if (n == 1) return listOf(this[size - 1])\n   val list = ArrayList<Boolean>(n)\n   for (index in size - n until size)\n   list.add(this[index])\n   return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun CharArray.takeLast(n: Int): List<Char> {\n   require(n >= 0) { \"Requested element count $n is less than zero.\" }\n   if (n == 0) return emptyList()\n   val size = size\n   if (n >= size) return toList()\n   if (n == 1) return listOf(this[size - 1])\n   val list = ArrayList<Char>(n)\n   for (index in size - n until size)\n   list.add(this[index])\n   return list\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun <T> Array<out
 T>.takeLastWhile(predicate: (T) -> Boolean): List<T> {\n   for (index in lastIndex downTo 0) {\n   if (!predicate(this[index])) {\n   return drop(index + 1)\n   }\n   }\n   return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun ByteArray.takeLastWhile(predicate: (Byte) -> Boolean): List<Byte> {\n   for (index in lastIndex downTo 0) {\n   if (!predicate(this[index])) {\n
 return drop(index + 1)\n   }\n   }\n   return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun ShortArray.takeLastWhile(predicate: (Short) -> Boolean): List<Short> {\n   for (index in lastIndex downTo 0) {\n   if (!predicate(this[index])) {\n   return drop(index + 1)\n
  }\n   }\n   return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun IntArray.takeLastWhile(predicate: (Int) -> Boolean): List<Int> {\n   for (index in lastIndex downTo 0) {\n   if (!predicate(this[index])) {\n   return drop(index + 1)\n   }\n   }\n   return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun LongArray.takeLastWhile(predicate: (Long) -> Boolean): List<Long> {\n   for (index in lastIndex downTo 0) {\n   if (!predicate(this[index])) {\n
  return drop(index + 1)\n   }\n   }\n   return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun FloatArray.takeLastWhile(predicate: (Float) -> Boolean): List<Float> {\n   for (index in lastIndex downTo 0) {\n   if (!predicate(this[index])) {\n   return drop(index + 1)\n   }\n   }\n   return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun DoubleArray.takeLastWhile(predicate: (Double) -> Boolean): List<Double> {\n   for (index in lastIndex downTo 0) {\n   if (!predicate(this[index])) {\n
        return drop(index + 1)\n   }\n   }\n   return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun BooleanArray.takeLastWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n   for (index in lastIndex downTo 0) {\n   if (!predicate(this[index])) {\n
          return drop(index + 1)\n   }\n   }\n   return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun CharArray.takeLastWhile(predicate: (Char) -> Boolean): List<Char> {\n   for (index in lastIndex downTo 0) {\n   if (!predicate(this[index])) {\n   return drop(index + 1)\n   }\n   }\n   return toList()\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun <T> Array<out

T>.takeWhile(predicate: (T) -> Boolean): List<T> {\n val list = ArrayList<T>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun ByteArray.takeWhile(predicate: (Byte) -> Boolean): List<Byte> {\n val list = ArrayList<Byte>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun ShortArray.takeWhile(predicate: (Short) -> Boolean): List<Short> {\n val list = ArrayList<Short>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun IntArray.takeWhile(predicate: (Int) -> Boolean): List<Int> {\n val list = ArrayList<Int>()\n for (item in this)
{\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun LongArray.takeWhile(predicate: (Long) -> Boolean): List<Long> {\n val list = ArrayList<Long>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun FloatArray.takeWhile(predicate: (Float) -> Boolean): List<Float> {\n val list = ArrayList<Float>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n
* \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun DoubleArray.takeWhile(predicate: (Double) -> Boolean): List<Double> {\n val list = ArrayList<Double>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun BooleanArray.takeWhile(predicate: (Boolean) -> Boolean): List<Boolean> {\n val list = ArrayList<Boolean>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic inline fun CharArray.takeWhile(predicate: (Char) -> Boolean): List<Char> {\n val
list = ArrayList<Char>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n
}\n return list\n}\n\n/**\n * Reverses elements in the array in-place.\n */\npublic fun <T> Array<T>.reverse(): Unit {\n val midPoint = (size / 2) - 1\n if (midPoint < 0) return\n var reverseIndex = lastIndex\n for (index in 0..midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n reverseIndex--\n }\n}\n\n/**\n * Reverses elements in the array in-place.\n */\npublic fun ByteArray.reverse(): Unit {\n val midPoint = (size / 2) - 1\n if (midPoint < 0) return\n var reverseIndex = lastIndex\n for (index in 0..midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n reverseIndex--\n }\n}\n\n/**\n * Reverses elements in the array in-place.\n */\npublic fun ShortArray.reverse(): Unit {\n
val midPoint = (size / 2) - 1\n if (midPoint < 0) return\n var reverseIndex = lastIndex\n for (index in 0..midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n reverseIndex--\n }\n}\n\n/**\n * Reverses elements in the array in-place.\n */\npublic fun IntArray.reverse(): Unit {\n val midPoint = (size / 2) - 1\n if (midPoint < 0) return\n var reverseIndex = lastIndex\n for (index in 0..midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n reverseIndex--\n }\n}\n\n/**\n * Reverses elements in the array in-place.\n */\npublic fun LongArray.reverse(): Unit {\n val midPoint = (size / 2) - 1\n if (midPoint < 0) return\n var reverseIndex = lastIndex\n for (index in 0..midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n reverseIndex--\n

```
}\n}\n\n/**\n * Reverses elements in the array in-place.\n */\npublic fun FloatArray.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements in the array in-place.\n */\npublic fun DoubleArray.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements in the array in-place.\n */\npublic fun BooleanArray.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements in the array in-place.\n */\npublic fun CharArray.reverse(): Unit {\n    val midPoint = (size / 2) - 1\n    if (midPoint < 0) return\n    var reverseIndex = lastIndex\n    for (index in 0..midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun <T> Array<T>.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex = toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun ByteArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex = toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun ShortArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex = toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun IntArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n    AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n    val midPoint = (fromIndex + toIndex) / 2\n    if (fromIndex == midPoint) return\n    var reverseIndex = toIndex - 1\n    for (index in fromIndex until midPoint) {\n        val tmp = this[index]\n        this[index] = this[reverseIndex]\n        this[reverseIndex] = tmp\n        reverseIndex--\n    }\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex]
```

is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun LongArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val midPoint = (fromIndex + toIndex) / 2\n if (fromIndex == midPoint) return\n var reverseIndex = toIndex - 1\n for (index in fromIndex until midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n reverseIndex--\n }\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun FloatArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val midPoint = (fromIndex + toIndex) / 2\n if (fromIndex == midPoint) return\n var reverseIndex = toIndex - 1\n for (index in fromIndex until midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n reverseIndex--\n }\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end
of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun DoubleArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val midPoint = (fromIndex + toIndex) / 2\n if (fromIndex == midPoint) return\n var reverseIndex = toIndex - 1\n for (index in fromIndex until midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n reverseIndex--\n }\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun BooleanArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val midPoint = (fromIndex + toIndex) / 2\n if (fromIndex == midPoint) return\n var reverseIndex = toIndex - 1\n for (index in fromIndex until midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n reverseIndex--\n }\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is
greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun CharArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val midPoint = (fromIndex + toIndex) / 2\n if (fromIndex == midPoint) return\n var reverseIndex = toIndex - 1\n for (index in fromIndex until midPoint) {\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n reverseIndex--\n }\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\npublic fun <T> Array<out T>.reversed(): List<T> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\npublic fun ByteArray.reversed(): List<Byte> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n
*/\npublic fun ShortArray.reversed(): List<Short> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\npublic fun IntArray.reversed(): List<Int> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\npublic fun LongArray.reversed(): List<Long> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\npublic fun

FloatArray.reversed(): List<Float> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\npublic fun DoubleArray.reversed(): List<Double> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\npublic fun BooleanArray.reversed(): List<Boolean> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\npublic fun CharArray.reversed(): List<Char> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic fun <T> Array<T>.reversedArray(): Array<T> {\n if (isEmpty()) return this\n val result = arrayOfNulls(this, size)\n val lastIndex = lastIndex\n for (i in 0..lastIndex)\n result[lastIndex - i] = this[i]\n return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic fun ByteArray.reversedArray(): ByteArray {\n if (isEmpty()) return this\n val result = ByteArray(size)\n val lastIndex = lastIndex\n for (i in 0..lastIndex)\n result[lastIndex - i] = this[i]\n return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic fun ShortArray.reversedArray(): ShortArray {\n if (isEmpty()) return this\n val result = ShortArray(size)\n val lastIndex = lastIndex\n for (i in 0..lastIndex)\n result[lastIndex - i] = this[i]\n return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic fun IntArray.reversedArray(): IntArray {\n if (isEmpty()) return this\n val result = IntArray(size)\n val lastIndex = lastIndex\n for (i in 0..lastIndex)\n result[lastIndex - i] = this[i]\n return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic fun LongArray.reversedArray(): LongArray {\n if (isEmpty()) return this\n val result = LongArray(size)\n val lastIndex = lastIndex\n for (i in 0..lastIndex)\n result[lastIndex - i] = this[i]\n return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic fun FloatArray.reversedArray(): FloatArray {\n if (isEmpty()) return this\n val result = FloatArray(size)\n val lastIndex = lastIndex\n for (i in 0..lastIndex)\n result[lastIndex - i] = this[i]\n return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic fun DoubleArray.reversedArray(): DoubleArray {\n if (isEmpty()) return this\n val result = DoubleArray(size)\n val lastIndex = lastIndex\n for (i in 0..lastIndex)\n result[lastIndex - i] = this[i]\n return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic fun BooleanArray.reversedArray(): BooleanArray {\n if (isEmpty()) return this\n val result = BooleanArray(size)\n val lastIndex = lastIndex\n for (i in 0..lastIndex)\n result[lastIndex - i] = this[i]\n return result\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\npublic fun CharArray.reversedArray(): CharArray {\n if (isEmpty()) return this\n val result = CharArray(size)\n val lastIndex = lastIndex\n for (i in 0..lastIndex)\n result[lastIndex - i] = this[i]\n return result\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T> Array<T>.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin(\"1.4\")\npublic fun ByteArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin(\"1.4\")\npublic fun ShortArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin(\"1.4\")\npublic fun IntArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin(\"1.4\")\npublic fun LongArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin(\"1.4\")\npublic fun FloatArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin(\"1.4\")\npublic fun DoubleArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin(\"1.4\")\npublic fun BooleanArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin(\"1.4\")\npublic fun CharArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n *

See: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm\n
*/\n@SinceKotlin(\"1.4\")\npublic fun <T> Array<T>.shuffle(random: Random): Unit {\n for (i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm\n
*/\n@SinceKotlin(\"1.4\")\npublic fun ByteArray.shuffle(random: Random): Unit {\n for (i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm\n
*/\n@SinceKotlin(\"1.4\")\npublic
fun ShortArray.shuffle(random: Random): Unit {\n for (i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm\n
*/\n@SinceKotlin(\"1.4\")\npublic fun IntArray.shuffle(random: Random): Unit {\n for (i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm\n
*/\n@SinceKotlin(\"1.4\")\npublic fun LongArray.shuffle(random: Random): Unit {\n for
(i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm\n
*/\n@SinceKotlin(\"1.4\")\npublic fun FloatArray.shuffle(random: Random): Unit {\n for (i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm\n
*/\n@SinceKotlin(\"1.4\")\npublic fun DoubleArray.shuffle(random: Random): Unit {\n for (i in lastIndex downTo 1) {\n val j = random.nextInt(i
+ 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm\n
*/\n@SinceKotlin(\"1.4\")\npublic fun BooleanArray.shuffle(random: Random): Unit {\n for (i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n this[j] = copy\n }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm\n
*/\n@SinceKotlin(\"1.4\")\npublic fun CharArray.shuffle(random: Random): Unit {\n for (i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] = this[j]\n
this[j] = copy\n }\n}\n\n/**\n * Sorts elements in the array in-place according to natural sort order of the value returned by specified [selector] function.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic inline fun <T, R : Comparable<R>> Array<out T>.sortBy(crossinline selector: (T) -> R?): Unit {\n if (size > 1) sortWith(compareBy(selector))\n}\n\n/**\n * Sorts elements in the array in-place descending according to natural sort order of the value returned by specified

[selector] function.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic inline fun <T, R : Comparable<R>> Array<out T>.sortByDescending(crossinline selector: (T) -> R?): Unit {\n if (size > 1) sortWith(compareByDescending(selector))\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> Array<out T>.sortDescending(): Unit {\n sortWith(reverseOrder())\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort order.\n */\npublic fun ByteArray.sortDescending(): Unit {\n if (size > 1) {\n sort()\n reverse()\n }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort order.\n */\npublic fun ShortArray.sortDescending(): Unit {\n if (size > 1) {\n sort()\n reverse()\n }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort order.\n */\npublic fun IntArray.sortDescending(): Unit {\n if (size > 1) {\n sort()\n reverse()\n }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort order.\n */\npublic fun LongArray.sortDescending(): Unit {\n if (size > 1) {\n sort()\n reverse()\n }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort order.\n */\npublic fun FloatArray.sortDescending(): Unit {\n if (size > 1) {\n sort()\n reverse()\n }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort order.\n */\npublic fun DoubleArray.sortDescending(): Unit {\n if (size > 1) {\n sort()\n reverse()\n }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort order.\n */\npublic fun CharArray.sortDescending(): Unit {\n if (size > 1) {\n sort()\n reverse()\n }\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> Array<out T>.sorted():
List<T> {\n return sortedArray().asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\npublic fun ByteArray.sorted(): List<Byte> {\n return toTypedArray().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\npublic fun ShortArray.sorted(): List<Short> {\n return toTypedArray().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\npublic fun IntArray.sorted(): List<Int> {\n return toTypedArray().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\npublic fun LongArray.sorted(): List<Long> {\n return toTypedArray().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\npublic fun FloatArray.sorted(): List<Float> {\n return toTypedArray().apply
{ sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\npublic fun DoubleArray.sorted(): List<Double> {\n return toTypedArray().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\npublic fun CharArray.sorted(): List<Char> {\n return toTypedArray().apply { sort() }.asList()\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> Array<T>.sortedArray(): Array<T> {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun ByteArray.sortedArray(): ByteArray {\n if (isEmpty()) return this\n return this.copyOf().apply
{ sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun ShortArray.sortedArray(): ShortArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun IntArray.sortedArray(): IntArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun LongArray.sortedArray(): LongArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun FloatArray.sortedArray(): FloatArray {\n if (isEmpty()) return this\n return this.copyOf().apply {

sort() }\n}\n\n/**\n * Returns
an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun DoubleArray.sortedArray(): DoubleArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\npublic fun CharArray.sortedArray(): CharArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> Array<T>.sortedArrayDescending(): Array<T> {\n if (isEmpty()) return this\n return this.copyOf().apply { sortWith(reverseOrder()) }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n
*/\npublic fun ByteArray.sortedArrayDescending(): ByteArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun ShortArray.sortedArrayDescending(): ShortArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun IntArray.sortedArrayDescending(): IntArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun LongArray.sortedArrayDescending(): LongArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements
of this array sorted descending according to their natural sort order.\n */\npublic fun FloatArray.sortedArrayDescending(): FloatArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun DoubleArray.sortedArrayDescending(): DoubleArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\npublic fun CharArray.sortedArrayDescending(): CharArray {\n if (isEmpty()) return this\n return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according the specified [comparator].\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic
fun <T> Array<out T>.sortedArrayWith(comparator: Comparator<in T>): Array<out T> {\n if (isEmpty()) return this\n return this.copyOf().apply { sortWith(comparator) }\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @sample samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <T, R : Comparable<R>> Array<out T>.sortedBy(crossinline selector: (T) -> R?): List<T> {\n return sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n * \n * @sample samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <R : Comparable<R>> ByteArray.sortedBy(crossinline selector: (Byte) -> R?): List<Byte> {\n return sortedWith(compareBy(selector))\n}\n\n/**\n
* Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n * \n * @sample samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <R : Comparable<R>> ShortArray.sortedBy(crossinline selector: (Short) -> R?): List<Short> {\n return sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n * \n * @sample samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <R : Comparable<R>> IntArray.sortedBy(crossinline selector: (Int) -> R?): List<Int> {\n return sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n * \n * @sample

samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <R : Comparable<R>> LongArray.sortedBy(crossinline selector: (Long) -> R?): List<Long> {\n return sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n * \n * @sample samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <R : Comparable<R>> FloatArray.sortedBy(crossinline selector: (Float) -> R?): List<Float> {\n return sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n * \n * @sample samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <R : Comparable<R>> DoubleArray.sortedBy(crossinline selector: (Double) -> R?): List<Double> {\n return sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n * \n * @sample samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <R : Comparable<R>> BooleanArray.sortedBy(crossinline selector: (Boolean) -> R?): List<Boolean> {\n return sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n * \n * @sample samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <R : Comparable<R>> CharArray.sortedBy(crossinline selector: (Char) -> R?): List<Char> {\n return sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic inline fun <T, R : Comparable<R>> Array<out T>.sortedByDescending(crossinline selector: (T) -> R?): List<T> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> ByteArray.sortedByDescending(crossinline selector: (Byte) -> R?): List<Byte> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> ShortArray.sortedByDescending(crossinline selector: (Short) -> R?): List<Short> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> IntArray.sortedByDescending(crossinline selector: (Int) -> R?): List<Int> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> LongArray.sortedByDescending(crossinline selector: (Long) -> R?): List<Long> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> FloatArray.sortedByDescending(crossinline selector: (Float) -> R?): List<Float> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> DoubleArray.sortedByDescending(crossinline selector: (Double) -> R?): List<Double> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> BooleanArray.sortedByDescending(crossinline selector: (Boolean) -> R?): List<Boolean> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n */\npublic inline fun <R : Comparable<R>> CharArray.sortedByDescending(crossinline selector: (Char) -> R?): List<Char> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n * \n * The sort is _stable_. It means

that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> Array<out T>.sortedDescending(): List<T> {\n   return sortedWith(reverseOrder())\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun ByteArray.sortedDescending(): List<Byte> {\n   return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun ShortArray.sortedDescending(): List<Short> {\n   return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun IntArray.sortedDescending(): List<Int> {\n   return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun LongArray.sortedDescending(): List<Long> {\n   return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun FloatArray.sortedDescending():
List<Float> {\n   return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun DoubleArray.sortedDescending(): List<Double> {\n   return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n */\npublic fun CharArray.sortedDescending(): List<Char> {\n   return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T> Array<out T>.sortedWith(comparator: Comparator<in T>): List<T> {\n   return sortedArrayWith(comparator).asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun ByteArray.sortedWith(comparator: Comparator<in Byte>):
List<Byte> {\n   return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun ShortArray.sortedWith(comparator: Comparator<in Short>): List<Short> {\n   return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun IntArray.sortedWith(comparator: Comparator<in Int>): List<Int> {\n   return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun LongArray.sortedWith(comparator: Comparator<in Long>): List<Long> {\n   return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun FloatArray.sortedWith(comparator: Comparator<in Float>): List<Float>
{\n   return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun DoubleArray.sortedWith(comparator: Comparator<in Double>): List<Double> {\n   return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun BooleanArray.sortedWith(comparator: Comparator<in Boolean>): List<Boolean> {\n   return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n */\npublic fun CharArray.sortedWith(comparator: Comparator<in Char>): List<Char> {\n   return toTypedArray().apply { sortWith(comparator) }.asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun <T> Array<out T>.asList(): List<T>\n\n/**\n * Returns a [List] that wraps the original array.\n
*/\npublic expect fun ByteArray.asList(): List<Byte>\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun ShortArray.asList(): List<Short>\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun IntArray.asList(): List<Int>\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun LongArray.asList(): List<Long>\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun FloatArray.asList(): List<Float>\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun DoubleArray.asList(): List<Double>\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic expect fun BooleanArray.asList(): List<Boolean>\n\n/**\n * Returns a [List] that wraps the

original array.\n */\npublic expect fun CharArray.asList(): List<Char>\n\n/**\n * Returns `true` if the two specified arrays are *deeply* equal to one another,\n * i.e. contain the same number of the same elements
in the same order.\n * \n * If two corresponding elements are nested arrays, they are also compared deeply.\n * If any of arrays contains itself on any nesting level the behavior is undefined.\n * \n * The elements of other types are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic expect infix fun <T> Array<out T>.contentDeepEquals(other: Array<out T>): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *deeply* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The specified arrays are also considered deeply equal if both are `null`.\n * \n * If two corresponding elements are nested arrays, they are also compared deeply.\n * If any of arrays contains itself on any nesting level the behavior is undefined.\n * \n
* The elements of other types are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\npublic expect infix fun <T> Array<out T>?.contentDeepEquals(other: Array<out T>?): Boolean\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n * Nested arrays are treated as lists too.\n * \n * If any of arrays contains itself on any nesting level the behavior is undefined.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic expect fun <T> Array<out T>.contentDeepHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n * Nested arrays are treated as lists too.\n * \n * If any of arrays contains itself on any nesting level the behavior is undefined.\n */\n@SinceKotlin(\"1.4\")\npublic expect fun <T> Array<out T>?.contentDeepHashCode(): Int\n\n/**\n
* Returns a string representation of the contents of this array as if it is a [List].\n * Nested arrays are treated as lists too.\n * \n * If any of arrays contains itself on any nesting level that reference\n * is rendered as `\"[...]\"` to prevent recursion.\n * \n * @sample samples.collections.Arrays.ContentOperations.contentDeepToString\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic expect fun <T> Array<out T>.contentDeepToString(): String\n\n/**\n * Returns a string representation of the contents of this array as if it is a [List].\n * Nested arrays are treated as lists too.\n * \n * If any of arrays contains itself on any nesting level that reference\n * is rendered as `\"[...]\"` to prevent recursion.\n * \n * @sample samples.collections.Arrays.ContentOperations.contentDeepToString\n */\n@SinceKotlin(\"1.4\")\npublic expect fun <T> Array<out T>?.contentDeepToString(): String\n\n/**\n * Returns `true` if the two specified arrays are *structurally*
equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect infix fun <T> Array<out T>.contentEquals(other: Array<out T>): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince
= \"1.4\")\npublic expect infix fun ByteArray.contentEquals(other: ByteArray): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect infix fun ShortArray.contentEquals(other: ShortArray): Boolean\n\n/**\n * Returns `true` if the two specified arrays are

structurally equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that
`NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect infix fun IntArray.contentEquals(other: IntArray): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect infix fun LongArray.contentEquals(other: LongArray): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the
same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect infix fun FloatArray.contentEquals(other: FloatArray): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect infix fun DoubleArray.contentEquals(other:
DoubleArray): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect infix fun BooleanArray.contentEquals(other: BooleanArray): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@Deprecated(\"Use
Kotlin compiler 1.4 to avoid deprecation
warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect infix fun CharArray.contentEquals(other: CharArray): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\npublic expect infix fun <T> Array<out T>?.contentEquals(other: Array<out T>?): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal
to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\npublic expect infix fun ByteArray?.contentEquals(other: ByteArray?): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\npublic expect infix fun ShortArray?.contentEquals(other: ShortArray?): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n

* The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\npublic expect infix fun IntArray?.contentEquals(other: IntArray?): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\npublic expect infix fun LongArray?.contentEquals(other: LongArray?): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\npublic expect infix fun FloatArray?.contentEquals(other: FloatArray?): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\npublic expect infix fun DoubleArray?.contentEquals(other: DoubleArray?): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\npublic expect infix fun BooleanArray?.contentEquals(other: BooleanArray?): Boolean\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\npublic expect infix fun CharArray?.contentEquals(other: CharArray?): Boolean\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun <T> Array<out T>.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun ByteArray.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun ShortArray.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun IntArray.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun LongArray.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun FloatArray.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun DoubleArray.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun

BooleanArray.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun CharArray.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\npublic expect fun <T> Array<out T>?.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\npublic expect fun ByteArray?.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\npublic expect fun ShortArray?.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\npublic expect fun IntArray?.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\npublic expect fun LongArray?.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\npublic expect fun FloatArray?.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\npublic expect fun DoubleArray?.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\npublic expect fun BooleanArray?.contentHashCode(): Int\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\npublic expect fun CharArray?.contentHashCode(): Int\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun <T> Array<out T>.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun ByteArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun ShortArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun IntArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun LongArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun FloatArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun DoubleArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified

array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun BooleanArray.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic expect fun CharArray.contentToString(): String\n\n/**\n * Returns
a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\npublic expect fun <T> Array<out T>?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\npublic expect fun ByteArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\npublic expect fun ShortArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\npublic
expect fun IntArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\npublic expect fun LongArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\npublic expect fun FloatArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\npublic expect fun DoubleArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\npublic expect fun BooleanArray?.contentToString(): String\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\npublic expect fun CharArray?.contentToString(): String\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException]
when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\npublic expect fun <T> Array<out T>.copyInto(destination: Array<T>, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): Array<T>\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the

subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\npublic expect fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ByteArray\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination
the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\npublic expect fun ShortArray.copyInto(destination: ShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ShortArray\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that
array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\npublic expect fun IntArray.copyInto(destination: IntArray,
destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is
out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\npublic expect fun LongArray.copyInto(destination: LongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): LongArray\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n

* @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\npublic expect fun FloatArray.copyInto(destination: FloatArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): FloatArray\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by
default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\npublic expect fun DoubleArray.copyInto(destination: DoubleArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): DoubleArray\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by
default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\npublic expect fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange
so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\npublic expect fun CharArray.copyInto(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): CharArray\n\n/**\n * Returns
new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\n@Suppress(\"NO_ACTUAL_FOR_EXPECT\")\npublic expect fun <T> Array<T>.copyOf(): Array<T>\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\npublic expect fun ByteArray.copyOf(): ByteArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\npublic expect fun ShortArray.copyOf(): ShortArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\npublic expect fun IntArray.copyOf(): IntArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample

samples.collections.Arrays.CopyOfOperations.copyOf\n */\npublic expect fun LongArray.copyOf(): LongArray\n\n/**\n * Returns new array
which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\npublic expect fun FloatArray.copyOf(): FloatArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\npublic expect fun DoubleArray.copyOf(): DoubleArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\npublic expect fun BooleanArray.copyOf(): BooleanArray\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\npublic expect fun CharArray.copyOf(): CharArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the
copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n */\npublic expect fun ByteArray.copyOf(newSize: Int): ByteArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n */\npublic expect fun ShortArray.copyOf(newSize: Int): ShortArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given
[newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n */\npublic expect fun IntArray.copyOf(newSize: Int): IntArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n
*/\npublic expect fun LongArray.copyOf(newSize: Int): LongArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n */\npublic expect fun FloatArray.copyOf(newSize: Int): FloatArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the
size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n */\npublic expect fun DoubleArray.copyOf(newSize: Int): DoubleArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with `false` values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `false` values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n */\npublic expect fun

BooleanArray.copyOf(newSize: Int): BooleanArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with null char (`\\u0000`) values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with null char (`\\u0000`) values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n */\npublic expect fun CharArray.copyOf(newSize: Int): CharArray\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with `null` values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `null` values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizingCopyOf\n */\n@Suppress(\"NO_ACTUAL_FOR_EXPECT\")\npublic expect fun <T> Array<T>.copyOf(newSize: Int): Array<T?>\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@Suppress(\"NO_ACTUAL_FOR_EXPECT\")\npublic expect fun <T> Array<T>.copyOfRange(fromIndex: Int, toIndex: Int): Array<T>\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic expect fun ByteArray.copyOfRange(fromIndex: Int, toIndex: Int): ByteArray\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic expect fun ShortArray.copyOfRange(fromIndex: Int, toIndex: Int): ShortArray\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic expect fun IntArray.copyOfRange(fromIndex: Int, toIndex: Int): IntArray\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic expect fun LongArray.copyOfRange(fromIndex: Int, toIndex: Int): LongArray\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic expect fun FloatArray.copyOfRange(fromIndex: Int, toIndex: Int): FloatArray\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic

expect fun DoubleArray.copyOfRange(fromIndex: Int, toIndex: Int): DoubleArray\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException
if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic expect fun BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic expect fun CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the
range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\npublic expect fun <T> Array<T>.fill(element: T, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\npublic expect fun ByteArray.fill(element: Byte, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this
array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\npublic expect fun ShortArray.fill(element: Short, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex]
is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\npublic expect fun IntArray.fill(element: Int, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\npublic expect fun LongArray.fill(element: Long, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n
* \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\npublic expect fun FloatArray.fill(element: Float, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n

*/\n@SinceKotlin(\"1.3\")\npublic expect fun DoubleArray.fill(element: Double, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\npublic expect fun BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\npublic expect fun CharArray.fill(element: Char, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Returns the range of valid indices for the array.\n */\npublic val <T> Array<out T>.indices: IntRange\n get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n */\npublic val ByteArray.indices: IntRange\n get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n */\npublic val ShortArray.indices: IntRange\n get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n */\npublic val IntArray.indices: IntRange\n get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n */\npublic val LongArray.indices: IntRange\n get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n */\npublic val FloatArray.indices: IntRange\n get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid
indices for the array.\n */\npublic val DoubleArray.indices: IntRange\n get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n */\npublic val BooleanArray.indices: IntRange\n get() = IntRange(0, lastIndex)\n\n/**\n * Returns the range of valid indices for the array.\n */\npublic val CharArray.indices: IntRange\n get() = IntRange(0, lastIndex)\n\n/**\n * Returns `true` if the array is empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.isEmpty(): Boolean {\n return size == 0\n}\n\n/**\n * Returns `true` if the array is empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.isEmpty(): Boolean {\n return size == 0\n}\n\n/**\n * Returns `true` if the array is empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.isEmpty(): Boolean {\n return size == 0\n}\n\n/**\n * Returns `true` if the array is empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.isEmpty(): Boolean {\n return size == 0\n}\n\n/**\n * Returns `true` if the array is empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.isEmpty(): Boolean {\n return size == 0\n}\n\n/**\n * Returns `true` if the array is empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.isEmpty(): Boolean {\n return size == 0\n}\n\n/**\n * Returns `true` if the array is empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.isEmpty(): Boolean {\n return size == 0\n}\n\n/**\n * Returns `true` if the array is empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.isEmpty(): Boolean {\n return size == 0\n}\n\n/**\n * Returns `true` if the array is empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.isEmpty(): Boolean {\n return size == 0\n}\n\n/**\n * Returns `true` if the array is not empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.isNotEmpty(): Boolean {\n return !isEmpty()\n}\n\n/**\n
* Returns `true` if the array is not empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.isNotEmpty(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if the array is not empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.isNotEmpty(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if the array is not empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.isNotEmpty(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if the array is not empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.isNotEmpty(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if the array is not empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.isNotEmpty(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if the array is not empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.isNotEmpty(): Boolean {\n return

!isEmpty()\n}\n\n/**\n
* Returns `true` if the array is not empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.isNotEmpty(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if the array is not empty.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.isNotEmpty(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns the last valid index for the array.\n */\npublic val <T> Array<out T>.lastIndex: Int\n get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\npublic val ByteArray.lastIndex: Int\n get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\npublic val ShortArray.lastIndex: Int\n get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\npublic val IntArray.lastIndex: Int\n get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\npublic val LongArray.lastIndex: Int\n get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\npublic
val FloatArray.lastIndex: Int\n get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\npublic val DoubleArray.lastIndex: Int\n get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\npublic val BooleanArray.lastIndex: Int\n get() = size - 1\n\n/**\n * Returns the last valid index for the array.\n */\npublic val CharArray.lastIndex: Int\n get() = size - 1\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@Suppress(\"NO_ACTUAL_FOR_EXPECT\")\npublic expect operator fun <T> Array<T>.plus(element: T): Array<T>\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\npublic expect operator fun ByteArray.plus(element: Byte): ByteArray\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\npublic expect operator fun ShortArray.plus(element: Short): ShortArray\n\n/**\n
* Returns an array containing all elements of the original array and then the given [element].\n */\npublic expect operator fun IntArray.plus(element: Int): IntArray\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\npublic expect operator fun LongArray.plus(element: Long): LongArray\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\npublic expect operator fun FloatArray.plus(element: Float): FloatArray\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\npublic expect operator fun DoubleArray.plus(element: Double): DoubleArray\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\npublic expect operator fun BooleanArray.plus(element: Boolean): BooleanArray\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n
*/\npublic expect operator fun CharArray.plus(element: Char): CharArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\n@Suppress(\"NO_ACTUAL_FOR_EXPECT\")\npublic expect operator fun <T> Array<T>.plus(elements: Collection<T>): Array<T>\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun ByteArray.plus(elements: Collection<Byte>): ByteArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun ShortArray.plus(elements: Collection<Short>): ShortArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun IntArray.plus(elements: Collection<Int>): IntArray\n\n/**\n
* Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun LongArray.plus(elements: Collection<Long>): LongArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun FloatArray.plus(elements: Collection<Float>): FloatArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun DoubleArray.plus(elements: Collection<Double>): DoubleArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic expect operator fun BooleanArray.plus(elements: Collection<Boolean>): BooleanArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the

given [elements]
collection.\n */\npublic expect operator fun CharArray.plus(elements: Collection<Char>): CharArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress(\"NO_ACTUAL_FOR_EXPECT\")\npublic expect operator fun <T> Array<T>.plus(elements: Array<out T>): Array<T>\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun ByteArray.plus(elements: ByteArray): ByteArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun ShortArray.plus(elements: ShortArray): ShortArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun IntArray.plus(elements: IntArray): IntArray\n\n/**\n * Returns an array
containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun LongArray.plus(elements: LongArray): LongArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun FloatArray.plus(elements: FloatArray): FloatArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\npublic expect operator fun CharArray.plus(elements:
CharArray): CharArray\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@Suppress(\"NO_ACTUAL_FOR_EXPECT\")\npublic expect fun <T> Array<T>.plusElement(element: T): Array<T>\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun IntArray.sort(): Unit\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun LongArray.sort(): Unit\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun ByteArray.sort(): Unit\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun ShortArray.sort(): Unit\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun DoubleArray.sort(): Unit\n\n/**\n * Sorts the array in-place.\n
* \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun FloatArray.sort(): Unit\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic expect fun CharArray.sort(): Unit\n\n/**\n * Sorts the array in-place according to the natural order of its elements.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @sample samples.collections.Arrays.Sorting.sortArrayOfComparable\n */\npublic expect fun <T : Comparable<T>> Array<out T>.sort(): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if
[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArrayOfComparable\n */\n@SinceKotlin(\"1.4\")\npublic expect fun <T : Comparable<T>> Array<out T>.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample

samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\npublic expect fun ByteArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts
a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\npublic expect fun ShortArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex]
is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\npublic expect fun IntArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\npublic expect fun LongArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to
sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\npublic expect fun FloatArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\npublic expect fun DoubleArray.sort(fromIndex:
Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\npublic expect fun CharArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex
the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun <T : Comparable<T>> Array<out T>.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n   sortWith(reverseOrder(), fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if

[fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun ByteArray.sortDescending(fromIndex:
Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun ShortArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of
the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun IntArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun LongArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n
sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun FloatArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to
sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun DoubleArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\npublic fun CharArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n

reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts the array in-place according to the order specified by the given [comparator].\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic expect fun <T> Array<out T>.sortWith(comparator: Comparator<in T>): Unit\n\n/**\n * Sorts a range in the array in-place with the given [comparator].\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic

expect fun <T> Array<out T>.sortWith(comparator: Comparator<in
T>, fromIndex: Int = 0, toIndex: Int = size): Unit\n\n/**\n * Returns an array of Boolean containing all of the elements of this generic array.\n */\npublic fun Array<out Boolean>.toBooleanArray(): BooleanArray {\n return BooleanArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Byte containing all of the elements of this generic array.\n */\npublic fun Array<out Byte>.toByteArray(): ByteArray {\n return ByteArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Char containing all of the elements of this generic array.\n */\npublic fun Array<out Char>.toCharArray(): CharArray {\n return CharArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Double containing all of the elements of this generic array.\n */\npublic fun Array<out Double>.toDoubleArray(): DoubleArray {\n return DoubleArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Float containing all of the elements of this generic array.\n */\npublic fun Array<out Float>.toFloatArray(): FloatArray {\n return FloatArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Int containing all of the elements of this generic array.\n */\npublic fun Array<out Int>.toIntArray(): IntArray {\n return IntArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Long containing all of the elements of this generic array.\n */\npublic fun Array<out Long>.toLongArray(): LongArray {\n return LongArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of Short containing all of the elements of this generic array.\n */\npublic fun Array<out Short>.toShortArray(): ShortArray {\n return ShortArray(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic expect fun ByteArray.toTypedArray(): Array<Byte>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic
expect fun ShortArray.toTypedArray(): Array<Short>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic expect fun IntArray.toTypedArray(): Array<Int>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic expect fun LongArray.toTypedArray(): Array<Long>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic expect fun FloatArray.toTypedArray(): Array<Float>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic expect fun DoubleArray.toTypedArray(): Array<Double>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic expect fun BooleanArray.toTypedArray(): Array<Boolean>\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic expect fun CharArray.toTypedArray():
Array<Char>\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n */\npublic inline fun <T, K, V> Array<out T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {\n val capacity = mapCapacity(size).coerceAtLeast(16)\n return associateTo(LinkedHashMap<K, V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n
*/\npublic inline fun <K, V> ByteArray.associate(transform: (Byte) -> Pair<K, V>): Map<K, V> {\n val capacity = mapCapacity(size).coerceAtLeast(16)\n return associateTo(LinkedHashMap<K, V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n */\npublic inline fun <K, V> ShortArray.associate(transform: (Short) -> Pair<K, V>): Map<K, V> {\n val capacity = mapCapacity(size).coerceAtLeast(16)\n return associateTo(LinkedHashMap<K, V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to

elements of the given array.\n
 * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n */\npublic inline fun <K, V> IntArray.associate(transform: (Int) -> Pair<K, V>): Map<K, V> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateTo(LinkedHashMap<K, V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n */\npublic inline fun <K, V> LongArray.associate(transform: (Long) -> Pair<K, V>): Map<K, V> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n
    return associateTo(LinkedHashMap<K, V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n */\npublic inline fun <K, V> FloatArray.associate(transform: (Float) -> Pair<K, V>): Map<K, V> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateTo(LinkedHashMap<K, V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original
 array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n */\npublic inline fun <K, V> DoubleArray.associate(transform: (Double) -> Pair<K, V>): Map<K, V> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateTo(LinkedHashMap<K, V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n */\npublic inline fun <K, V> BooleanArray.associate(transform: (Boolean) -> Pair<K, V>): Map<K, V> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateTo(LinkedHashMap<K, V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing key-value pairs
 provided by [transform] function\n * applied to elements of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitives\n */\npublic inline fun <K, V> CharArray.associate(transform: (Char) -> Pair<K, V>): Map<K, V> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateTo(LinkedHashMap<K, V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n * returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n
 */\npublic inline fun <T, K> Array<out T>.associateBy(keySelector: (T) -> K): Map<K, T> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, T>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n * returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample

samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K> ByteArray.associateBy(keySelector: (Byte) -> K): Map<K, Byte> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, Byte>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n * returned
 from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K> ShortArray.associateBy(keySelector: (Short) -> K): Map<K, Short> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, Short>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n * returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
 samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K> IntArray.associateBy(keySelector: (Int) -> K): Map<K, Int> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, Int>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n * returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
 samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K> LongArray.associateBy(keySelector: (Long) -> K): Map<K, Long> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, Long>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n
 * returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
 samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K> FloatArray.associateBy(keySelector: (Float) -> K): Map<K, Float> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, Float>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n * returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
 samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K> DoubleArray.associateBy(keySelector: (Double) -> K): Map<K, Double> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, Double>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the given array indexed by the key\n * returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample
 samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K> BooleanArray.associateBy(keySelector: (Boolean) -> K): Map<K, Boolean> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, Boolean>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the elements from the
 given array indexed by the key\n * returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The

returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesBy\n */\npublic inline fun <K> CharArray.associateBy(keySelector: (Char) -> K): Map<K, Char> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, Char>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n */\npublic inline fun <T, K, V> Array<out T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n */\npublic inline fun <K, V> ByteArray.associateBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V): Map<K, V> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n */\npublic inline fun <K, V> ShortArray.associateBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, V> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n */\npublic inline fun <K, V> IntArray.associateBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, V> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n */\npublic inline fun <K, V> LongArray.associateBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, V> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n */\npublic inline fun <K, V> FloatArray.associateBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, V> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, V>(capacity),

keySelector, valueTransform)\n}\n\n/**\n
 * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n */\npublic inline fun <K, V> DoubleArray.associateBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, V> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n
 * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n */\npublic inline fun <K, V> BooleanArray.associateBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, V> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByWithValueTransform\n */\npublic inline fun <K, V> CharArray.associateBy(keySelector:
 (Char) -> K, valueTransform: (Char) -> V): Map<K, V> {\n   val capacity = mapCapacity(size).coerceAtLeast(16)\n   return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline fun <T, K, M : MutableMap<in K, in T>> Array<out T>.associateByTo(destination: M, keySelector: (T) -> K): M {\n   for (element in this) {\n   destination.put(keySelector(element), element)\n   }\n   return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector]
 function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline fun <K, M : MutableMap<in K, in Byte>> ByteArray.associateByTo(destination: M, keySelector: (Byte) -> K): M {\n   for (element in this) {\n   destination.put(keySelector(element), element)\n   }\n   return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline
 fun <K, M : MutableMap<in K, in Short>> ShortArray.associateByTo(destination: M, keySelector: (Short) -> K): M {\n   for (element in this) {\n   destination.put(keySelector(element), element)\n   }\n   return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline fun <K, M : MutableMap<in K, in Int>> IntArray.associateByTo(destination: M, keySelector: (Int) -> K): M {\n   for (element in this) {\n   destination.put(keySelector(element), element)\n   }\n   return

destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline fun <K, M : MutableMap<in K, in Long>> LongArray.associateByTo(destination: M, keySelector: (Long) -> K): M {\n for (element in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline fun <K, M : MutableMap<in K, in Float>> FloatArray.associateByTo(destination: M, keySelector: (Float) -> K): M {\n for (element in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline fun <K, M : MutableMap<in K, in Double>> DoubleArray.associateByTo(destination: M, keySelector: (Double) -> K): M {\n for (element in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline fun <K, M : MutableMap<in K, in Boolean>> BooleanArray.associateByTo(destination: M, keySelector: (Boolean) -> K): M {\n for (element in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given array\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByTo\n */\npublic inline fun <K, M : MutableMap<in K, in Char>> CharArray.associateByTo(destination: M, keySelector: (Char) -> K): M {\n for (element in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n */\npublic inline fun <T, K, V, M : MutableMap<in K, in V>> Array<out T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> ByteArray.associateByTo(destination: M, keySelector: (Byte) -> K, valueTransform: (Byte) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by

the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> ShortArray.associateByTo(destination: M, keySelector: (Short) -> K, valueTransform: (Short) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> IntArray.associateByTo(destination: M, keySelector: (Int) -> K, valueTransform: (Int) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> LongArray.associateByTo(destination: M, keySelector: (Long) -> K, valueTransform: (Long) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> FloatArray.associateByTo(destination: M, keySelector: (Float) -> K, valueTransform: (Float) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> DoubleArray.associateByTo(destination: M, keySelector: (Double) -> K, valueTransform: (Double) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> BooleanArray.associateByTo(destination: M, keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given array.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesByToWithValueTransform\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> CharArray.associateByTo(destination: M, keySelector:

(Char)
-> K, valueTransform: (Char) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <T, K, V, M : MutableMap<in K, in V>> Array<out T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key the last one gets
added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> ByteArray.associateTo(destination: M, transform: (Byte) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> ShortArray.associateTo(destination: M, transform: (Short) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with
key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> IntArray.associateTo(destination: M, transform: (Int) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> LongArray.associateTo(destination: M, transform: (Long) -> Pair<K, V>): M {\n
for (element in this) {\n destination += transform(element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> FloatArray.associateTo(destination: M, transform: (Float) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n
*/\npublic inline fun <K, V, M : MutableMap<in K, in V>> DoubleArray.associateTo(destination: M, transform: (Double) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> BooleanArray.associateTo(destination: M, transform: (Boolean) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n }\n return destination\n}\n\n/**\n * Populates

and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given array.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample samples.collections.Arrays.Transformations.associateArrayOfPrimitivesTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> CharArray.associateTo(destination: M, transform: (Char) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n }\n return destination\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <K, V> Array<out K>.associateWith(valueSelector: (K) -> V): Map<K, V> {\n val result = LinkedHashMap<K, V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <V> ByteArray.associateWith(valueSelector: (Byte) -> V): Map<Byte, V> {\n val result = LinkedHashMap<Byte, V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <V> ShortArray.associateWith(valueSelector: (Short) -> V): Map<Short, V> {\n val result = LinkedHashMap<Short, V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <V> IntArray.associateWith(valueSelector: (Int) -> V): Map<Int, V> {\n val result = LinkedHashMap<Int, V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <V> LongArray.associateWith(valueSelector: (Long) -> V): Map<Long, V> {\n val result = LinkedHashMap<Long, V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal,\n the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <V> FloatArray.associateWith(valueSelector: (Float) -> V): Map<Float, V> {\n val result = LinkedHashMap<Float, V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function

applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <V> DoubleArray.associateWith(valueSelector: (Double) -> V): Map<Double, V> {\n val result = LinkedHashMap<Double, V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <V> BooleanArray.associateWith(valueSelector: (Boolean) -> V): Map<Boolean, V> {\n val result = LinkedHashMap<Boolean, V>(mapCapacity(size).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <V> CharArray.associateWith(valueSelector: (Char) -> V): Map<Char, V> {\n val result = LinkedHashMap<Char, V>(mapCapacity(size.coerceAtMost(128)).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n * @sample samples.collections.Collections.Transformations.associateWithTo\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <K, V, M : MutableMap<in K, in V>> Array<out K>.associateWithTo(destination: M, valueSelector: (K) -> V): M {\n for (element in this) {\n destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n * @sample samples.collections.Collections.Transformations.associateWithTo\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Byte, in V>> ByteArray.associateWithTo(destination: M, valueSelector: (Byte) -> V): M {\n for (element in this) {\n destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n * @sample samples.collections.Collections.Transformations.associateWithTo\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Short, in V>> ShortArray.associateWithTo(destination: M, valueSelector: (Short) -> V): M {\n for (element in this) {\n destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n *

@sample samples.collections.Collections.Transformations.associateWithTo\n
*/\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Int, in V>>
IntArray.associateWithTo(destination: M, valueSelector: (Int) -> V): M {\n for (element in this) {\n
destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are
equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*/\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Long, in V>>
LongArray.associateWithTo(destination: M, valueSelector: (Long) -> V): M {\n for (element in this) {\n
destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n/**\n * Populates and
returns the [destination] mutable map with key-value pairs for each element of the given array,\n * where key is the
element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements
are equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*/\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Float, in V>>
FloatArray.associateWithTo(destination: M, valueSelector: (Float) -> V): M {\n for (element in this) {\n
destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector]
function applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the
map.\n * \n * @sample samples.collections.Collections.Transformations.associateWithTo\n
*/\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Double, in V>>
DoubleArray.associateWithTo(destination: M, valueSelector: (Double) -> V): M {\n for (element in this) {\n
destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element
itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are
equal, the last one overwrites the former value in the map.\n * \n * @sample
samples.collections.Collections.Transformations.associateWithTo\n
*/\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline
fun <V, M : MutableMap<in Boolean, in V>> BooleanArray.associateWithTo(destination: M, valueSelector:
(Boolean) -> V): M {\n for (element in this) {\n destination.put(element, valueSelector(element))\n }\n
return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each
element of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function
applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n
* @sample samples.collections.Collections.Transformations.associateWithTo\n
*/\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in Char, in V>>
CharArray.associateWithTo(destination: M, valueSelector: (Char) -> V): M {\n for (element in this) {\n
destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n/**\n * Appends all elements to
the given
[destination] collection.\n */\npublic fun <T, C : MutableCollection<in T>> Array<out T>.toCollection(destination:
C): C {\n for (item in this) {\n destination.add(item)\n }\n return destination\n}\n\n/**\n * Appends all
elements to the given [destination] collection.\n */\npublic fun <C : MutableCollection<in Byte>>
ByteArray.toCollection(destination: C): C {\n for (item in this) {\n destination.add(item)\n }\n return
destination\n}\n\n/**\n * Appends all elements to the given [destination] collection.\n */\npublic fun <C :
MutableCollection<in Short>> ShortArray.toCollection(destination: C): C {\n for (item in this) {\n
destination.add(item)\n }\n return destination\n}\n\n/**\n * Appends all elements to the given [destination]
collection.\n */\npublic fun <C : MutableCollection<in Int>> IntArray.toCollection(destination: C): C {\n for

(item in this) {\n destination.add(item)\n }\n return destination\n}\n\n/**\n * Appends all elements to the given [destination] collection.\n */\npublic fun <C : MutableCollection<in Long>> LongArray.toCollection(destination: C): C {\n for (item in this) {\n destination.add(item)\n }\n return destination\n}\n\n/**\n * Appends all elements to the given [destination] collection.\n */\npublic fun <C : MutableCollection<in Float>> FloatArray.toCollection(destination: C): C {\n for (item in this) {\n destination.add(item)\n }\n return destination\n}\n\n/**\n * Appends all elements to the given [destination] collection.\n */\npublic fun <C : MutableCollection<in Double>> DoubleArray.toCollection(destination: C): C {\n for (item in this) {\n destination.add(item)\n }\n return destination\n}\n\n/**\n * Appends all elements to the given [destination] collection.\n */\npublic fun <C : MutableCollection<in Boolean>> BooleanArray.toCollection(destination: C): C {\n for (item in this) {\n destination.add(item)\n }\n return destination\n}\n\n/**\n * Appends all elements to the given [destination] collection.\n */\npublic fun <C : MutableCollection<in Char>> CharArray.toCollection(destination: C): C {\n for (item in this) {\n destination.add(item)\n }\n return destination\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n */\npublic fun <T> Array<out T>.toHashSet(): HashSet<T> {\n return toCollection(HashSet<T>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n */\npublic fun ByteArray.toHashSet(): HashSet<Byte> {\n return toCollection(HashSet<Byte>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n */\npublic fun ShortArray.toHashSet(): HashSet<Short> {\n return toCollection(HashSet<Short>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n */\npublic fun IntArray.toHashSet(): HashSet<Int> {\n return toCollection(HashSet<Int>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n */\npublic fun LongArray.toHashSet(): HashSet<Long> {\n return toCollection(HashSet<Long>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n */\npublic fun FloatArray.toHashSet(): HashSet<Float> {\n return toCollection(HashSet<Float>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n */\npublic fun DoubleArray.toHashSet(): HashSet<Double> {\n return toCollection(HashSet<Double>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n */\npublic fun BooleanArray.toHashSet(): HashSet<Boolean> {\n return toCollection(HashSet<Boolean>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n */\npublic fun CharArray.toHashSet(): HashSet<Char> {\n return toCollection(HashSet<Char>(mapCapacity(size.coerceAtMost(128))))\n}\n\n/**\n * Returns a [List] containing all elements.\n */\npublic fun <T> Array<out T>.toList(): List<T> {\n return when (size) {\n 0 -> emptyList()\n 1 -> listOf(this[0])\n else -> this.toMutableList()\n }\n}\n\n/**\n * Returns a [List] containing all elements.\n */\npublic fun ByteArray.toList(): List<Byte> {\n return when (size) {\n 0 -> emptyList()\n 1 -> listOf(this[0])\n else -> this.toMutableList()\n }\n}\n\n/**\n * Returns a [List] containing all elements.\n */\npublic fun ShortArray.toList(): List<Short> {\n return when (size) {\n 0 -> emptyList()\n 1 -> listOf(this[0])\n else -> this.toMutableList()\n }\n}\n\n/**\n * Returns a [List] containing all elements.\n */\npublic fun IntArray.toList(): List<Int> {\n return when (size) {\n 0 -> emptyList()\n 1 -> listOf(this[0])\n else -> this.toMutableList()\n }\n}\n\n/**\n * Returns a [List] containing all elements.\n */\npublic fun LongArray.toList(): List<Long> {\n return when (size) {\n 0 -> emptyList()\n 1 -> listOf(this[0])\n else -> this.toMutableList()\n }\n}\n\n/**\n * Returns a [List] containing all elements.\n */\npublic fun FloatArray.toList(): List<Float> {\n return when (size) {\n 0 -> emptyList()\n 1 -> listOf(this[0])\n else -> this.toMutableList()\n }\n}\n\n/**\n * Returns a [List] containing all elements.\n */\npublic fun DoubleArray.toList(): List<Double> {\n return when (size) {\n 0 -> emptyList()\n 1 -> listOf(this[0])\n else -> this.toMutableList()\n }\n}\n\n/**\n * Returns a [List] containing all elements.\n */\npublic fun BooleanArray.toList(): List<Boolean> {\n return when (size) {\n 0 -> emptyList()\n 1 -> listOf(this[0])\n else -> this.toMutableList()\n }\n}\n\n/**\n * Returns a [List] containing all elements.\n */\npublic fun CharArray.toList(): List<Char> {\n

```
return when (size) {\n        0 -> emptyList()\n        1 -> listOf(this[0])\n        else -> this.toMutableList()\n    }\n}\n\n/**\n * Returns a new [MutableList] filled
 with all elements of this array.\n */\npublic fun <T> Array<out T>.toMutableList(): MutableList<T> {\n    return ArrayList(this.asCollection())\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun ByteArray.toMutableList(): MutableList<Byte> {\n    val list = ArrayList<Byte>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun ShortArray.toMutableList(): MutableList<Short> {\n    val list = ArrayList<Short>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun IntArray.toMutableList(): MutableList<Int> {\n    val list = ArrayList<Int>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun LongArray.toMutableList(): MutableList<Long>
 {\n    val list = ArrayList<Long>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun FloatArray.toMutableList(): MutableList<Float> {\n    val list = ArrayList<Float>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun DoubleArray.toMutableList(): MutableList<Double> {\n    val list = ArrayList<Double>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun BooleanArray.toMutableList(): MutableList<Boolean> {\n    val list = ArrayList<Boolean>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this array.\n */\npublic fun CharArray.toMutableList(): MutableList<Char> {\n    val
 list = ArrayList<Char>(size)\n    for (item in this) list.add(item)\n    return list\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun <T> Array<out T>.toSet(): Set<T> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<T>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun ByteArray.toSet(): Set<Byte> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Byte>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun ShortArray.toSet(): Set<Short> {\n    return when (size) {\n        0 -> emptySet()\n
        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Short>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun IntArray.toSet(): Set<Int> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Int>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun LongArray.toSet(): Set<Long> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Long>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun FloatArray.toSet(): Set<Float> {\n    return when (size) {\n
        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Float>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun DoubleArray.toSet(): Set<Double> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Double>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun BooleanArray.toSet(): Set<Boolean> {\n    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else -> toCollection(LinkedHashSet<Boolean>(mapCapacity(size)))\n    }\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun CharArray.toSet(): Set<Char> {\n
```

```
    return when (size) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else ->
toCollection(LinkedHashSet<Char>(mapCapacity(size.coerceAtMost(128))))\n    }\n}\n\n/**\n * Returns a single
list of all elements yielded from results of [transform] function being invoked on each element of original array.\n *
\n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <T, R> Array<out
T>.flatMap(transform: (T) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n
* Returns a single list of all elements yielded from results of [transform] function being invoked on each element of
original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun
<R> ByteArray.flatMap(transform: (Byte) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked
 on each element of original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n
*/\npublic inline fun <R> ShortArray.flatMap(transform: (Short) -> Iterable<R>): List<R> {\n    return
flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of
[transform] function being invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <R> IntArray.flatMap(transform:
(Int) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single
list of all elements yielded from results of [transform] function being invoked on each element of original array.\n *
\n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <R>
LongArray.flatMap(transform: (Long) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single
 list of all elements yielded from results of [transform] function being invoked on each element of original array.\n *
\n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <R>
FloatArray.flatMap(transform: (Float) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <R>
DoubleArray.flatMap(transform: (Double) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <R>
BooleanArray.flatMap(transform: (Boolean) -> Iterable<R>):
 List<R> {\n    return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements
yielded from results of [transform] function being invoked on each element of original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <R> CharArray.flatMap(transform:
(Char) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single
list of all elements yielded from results of [transform] function being invoked on each element of original array.\n *
\n * @sample samples.collections.Collections.Transformations.flatMap\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapSequence\")\npublic inline fun <T, R> Array<out
T>.flatMap(transform: (T) -> Sequence<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements
 yielded from results of [transform] function being invoked on each element\n * and its index in the original array.\n
* \n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterable\")\n@kotlin.internal.InlineOnly\npublic
inline fun <T, R> Array<out T>.flatMapIndexed(transform: (index: Int, T) -> Iterable<R>): List<R> {\n    return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
```

@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterable\")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> ByteArray.flatMapIndexed(transform: (index: Int, Byte) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterable\")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> ShortArray.flatMapIndexed(transform: (index: Int, Short) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterable\")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> IntArray.flatMapIndexed(transform: (index: Int, Int) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterable\")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> LongArray.flatMapIndexed(transform: (index: Int, Long) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(),
transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element\n * and its index in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterable\")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> FloatArray.flatMapIndexed(transform: (index: Int, Float) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterable\")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> DoubleArray.flatMapIndexed(transform: (index: Int, Double) -> Iterable<R>): List<R> {\n return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterable\")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> BooleanArray.flatMapIndexed(transform: (index: Int, Boolean) -> Iterable<R>): List<R> {\n
return flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded
from results of [transform] function being invoked
on each element\n * and its index in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterable\")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> CharArray.flatMapIndexed(transform: (index: Int, Char) -> Iterable<R>): List<R> {\n return

flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedSequence\")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out T>.flatMapIndexed(transform: (index: Int, T) -> Sequence<R>): List<R> {\n return flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given [destination].\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterableTo\")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapIndexedTo(destination: C, transform: (index: Int, T) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++, element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given [destination].\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterableTo\")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> ByteArray.flatMapIndexedTo(destination: C, transform: (index: Int, Byte) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++, element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given [destination].\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterableTo\")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> ShortArray.flatMapIndexedTo(destination: C, transform: (index: Int, Short) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++, element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given [destination].\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterableTo\")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> IntArray.flatMapIndexedTo(destination: C, transform: (index: Int, Int) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++, element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given [destination].\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterableTo\")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> LongArray.flatMapIndexedTo(destination: C, transform: (index: Int, Long) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++, element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given [destination].\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterableTo\")\n@kotlin.internal.InlineOnly\npublic

inline fun <R, C : MutableCollection<in R>> FloatArray.flatMapIndexedTo(destination: C, transform: (index: Int, Float) -> Iterable<R>): C {\n   var index = 0\n   for (element in this) {\n   val list = transform(index++, element)\n   destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given [destination].\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterableTo\")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> DoubleArray.flatMapIndexedTo(destination: C, transform: (index: Int, Double) -> Iterable<R>): C {\n   var index = 0\n   for (element in this) {\n   val list = transform(index++, element)\n   destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given [destination].\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterableTo\")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> BooleanArray.flatMapIndexedTo(destination: C, transform: (index: Int, Boolean) -> Iterable<R>): C {\n   var index = 0\n   for (element in this) {\n   val list = transform(index++, element)\n   destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given [destination].\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterableTo\")\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> CharArray.flatMapIndexedTo(destination: C, transform: (index: Int, Char) -> Iterable<R>): C {\n   var index = 0\n   for (element in this) {\n   val list = transform(index++, element)\n   destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array, to the given [destination].\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedSequenceTo\")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapIndexedTo(destination: C, transform: (index: Int, T) -> Sequence<R>): C {\n   var index = 0\n   for (element in this) {\n   val list = transform(index++, element)\n   destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n */\npublic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapTo(destination: C, transform: (T) -> Iterable<R>): C {\n   for (element in this) {\n   val list = transform(element)\n   destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> ByteArray.flatMapTo(destination: C, transform: (Byte) -> Iterable<R>): C {\n   for (element in this) {\n   val list = transform(element)\n   destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> ShortArray.flatMapTo(destination: C, transform: (Short) -> Iterable<R>): C {\n   for (element in this) {\n   val list = transform(element)\n   destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> IntArray.flatMapTo(destination: C, transform: (Int) -> Iterable<R>): C {\n   for (element in this) {\n   val list = transform(element)\n   destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given

[destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> LongArray.flatMapTo(destination: C, transform: (Long) -> Iterable<R>): C {\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> FloatArray.flatMapTo(destination: C, transform: (Float) -> Iterable<R>): C {\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> DoubleArray.flatMapTo(destination: C, transform: (Double) -> Iterable<R>): C {\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> BooleanArray.flatMapTo(destination: C, transform: (Boolean) -> Iterable<R>): C {\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> CharArray.flatMapTo(destination: C, transform: (Char) -> Iterable<R>): C {\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapSequenceTo\")\npublic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C {\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <T, K> Array<out T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {\n return groupByTo(LinkedHashMap<K, MutableList<T>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K> ByteArray.groupBy(keySelector: (Byte) -> K): Map<K, List<Byte>> {\n return groupByTo(LinkedHashMap<K, MutableList<Byte>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K> ShortArray.groupBy(keySelector: (Short) -> K): Map<K, List<Short>> {\n return groupByTo(LinkedHashMap<K, MutableList<Short>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K> IntArray.groupBy(keySelector: (Int) -> K): Map<K, List<Int>> {\n return groupByTo(LinkedHashMap<K, MutableList<Int>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the

given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a
list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K> LongArray.groupBy(keySelector: (Long) -> K): Map<K, List<Long>> {\n return groupByTo(LinkedHashMap<K, MutableList<Long>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K> FloatArray.groupBy(keySelector: (Float) -> K): Map<K, List<Float>> {\n return groupByTo(LinkedHashMap<K, MutableList<Float>>(), keySelector)\n}\n\n/**\n * Groups elements of the
original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K> DoubleArray.groupBy(keySelector: (Double) -> K): Map<K, List<Double>> {\n return groupByTo(LinkedHashMap<K, MutableList<Double>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K> BooleanArray.groupBy(keySelector:
(Boolean) -> K): Map<K, List<Boolean>> {\n return groupByTo(LinkedHashMap<K, MutableList<Boolean>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K> CharArray.groupBy(keySelector: (Char) -> K): Map<K, List<Char>> {\n return groupByTo(LinkedHashMap<K, MutableList<Char>>(), keySelector)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n
* \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <T, K, V> Array<out T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>> {\n return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V> ByteArray.groupBy(keySelector: (Byte) -> K, valueTransform: (Byte) -> V):
Map<K, List<V>> {\n return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample

samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V> ShortArray.groupBy(keySelector: (Short) -> K, valueTransform: (Short) -> V): Map<K, List<V>> {\n   return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n
 * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V> IntArray.groupBy(keySelector: (Int) -> K, valueTransform: (Int) -> V): Map<K, List<V>> {\n   return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic
 inline fun <K, V> LongArray.groupBy(keySelector: (Long) -> K, valueTransform: (Long) -> V): Map<K, List<V>> {\n   return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V> FloatArray.groupBy(keySelector: (Float) -> K, valueTransform: (Float) -> V): Map<K, List<V>> {\n   return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the
 original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V> DoubleArray.groupBy(keySelector: (Double) -> K, valueTransform: (Double) -> V): Map<K, List<V>> {\n   return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n
 * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V> BooleanArray.groupBy(keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): Map<K, List<V>> {\n   return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V> CharArray.groupBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, List<V>> {\n   return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n
 * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n

*/\npublic inline fun <T, K, M : MutableMap<in K, MutableList<T>>> Array<out T>.groupByTo(destination: M, keySelector: (T) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<T>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K, MutableList<Byte>>> ByteArray.groupByTo(destination: M, keySelector: (Byte) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<Byte>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K, MutableList<Short>>> ShortArray.groupByTo(destination: M, keySelector: (Short) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<Short>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K, MutableList<Int>>> IntArray.groupByTo(destination: M, keySelector: (Int) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<Int>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K, MutableList<Long>>> LongArray.groupByTo(destination: M, keySelector: (Long) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<Long>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K, MutableList<Float>>> FloatArray.groupByTo(destination: M, keySelector: (Float) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) {
ArrayList<Float>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K, MutableList<Double>>> DoubleArray.groupByTo(destination: M, keySelector: (Double) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<Double>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K, MutableList<Boolean>>> BooleanArray.groupByTo(destination: M, keySelector: (Boolean) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<Boolean>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements of the

original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K, MutableList<Char>>> CharArray.groupByTo(destination: M, keySelector: (Char) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<Char>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <T, K, V, M : MutableMap<in K, MutableList<V>>> Array<out T>.groupByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function
applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V, M : MutableMap<in K, MutableList<V>>> ByteArray.groupByTo(destination: M, keySelector: (Byte) -> K, valueTransform: (Byte) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with
a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V, M : MutableMap<in K, MutableList<V>>> ShortArray.groupByTo(destination: M, keySelector: (Short) -> K, valueTransform: (Short) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V, M : MutableMap<in
K, MutableList<V>>> IntArray.groupByTo(destination: M, keySelector: (Int) -> K, valueTransform: (Int) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V, M : MutableMap<in K, MutableList<V>>> LongArray.groupByTo(destination: M, keySelector: (Long) -> K, valueTransform: (Long) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination]

map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V, M : MutableMap<in K, MutableList<V>>> FloatArray.groupByTo(destination: M, keySelector: (Float) -> K, valueTransform: (Float) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V, M : MutableMap<in K, MutableList<V>>> DoubleArray.groupByTo(destination: M, keySelector: (Double) -> K, valueTransform: (Double) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V, M : MutableMap<in K, MutableList<V>>> BooleanArray.groupByTo(destination: M, keySelector: (Boolean) -> K, valueTransform: (Boolean) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V, M : MutableMap<in K, MutableList<V>>> CharArray.groupByTo(destination: M, keySelector: (Char) -> K, valueTransform: (Char) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Creates a [Grouping] source from an array to be used later with one of group-and-fold operations\n * using the specified [keySelector] function to extract a key from each element.\n * \n * @sample samples.collections.Grouping.groupingByEachCount\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <T, K> Array<out T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> {\n return object : Grouping<T, K> {\n override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()\n override fun keyOf(element: T): K = keySelector(element)\n }\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n */\npublic inline fun <T, R> Array<out T>.map(transform: (T) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> ByteArray.map(transform: (Byte) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> ShortArray.map(transform: (Short) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> IntArray.map(transform: (Int) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results

of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> LongArray.map(transform: (Long) -> R): List<R> {\n   return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> FloatArray.map(transform: (Float) -> R): List<R> {\n   return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the
given [transform] function\n * to each element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> DoubleArray.map(transform: (Double) -> R): List<R> {\n   return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> BooleanArray.map(transform: (Boolean) -> R): List<R> {\n   return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n */\npublic inline fun <R> CharArray.map(transform: (Char) -> R): List<R> {\n   return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying
the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <T, R> Array<out T>.mapIndexed(transform: (index: Int, T) -> R): List<R> {\n   return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R> ByteArray.mapIndexed(transform: (index: Int, Byte) -> R): List<R> {\n   return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in
the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R> ShortArray.mapIndexed(transform: (index: Int, Short) -> R): List<R> {\n   return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R> IntArray.mapIndexed(transform: (index: Int, Int) -> R): List<R> {\n   return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of
an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R> LongArray.mapIndexed(transform: (index: Int, Long) -> R): List<R> {\n   return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R> FloatArray.mapIndexed(transform: (index: Int, Float) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform
applied to the element.\n */\npublic inline fun <R> DoubleArray.mapIndexed(transform: (index: Int, Double) -> R): List<R> {\n   return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n *

@param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R> BooleanArray.mapIndexed(transform: (index: Int, Boolean) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R> CharArray.mapIndexed(transform: (index: Int, Char) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing only the non-null results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <T, R : Any> Array<out T>.mapIndexedNotNull(transform: (index: Int, T) -> R?): List<R> {\n return mapIndexedNotNullTo(ArrayList<R>(), transform)\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends only the non-null results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <T, R : Any, C : MutableCollection<in R>> Array<out T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {\n forEachIndexed { index, element -> transform(index, element)?.let { destination.add(it) } }\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.mapIndexedTo(destination: C, transform: (index: Int, T) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in R>> ByteArray.mapIndexedTo(destination: C, transform: (index: Int, Byte) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in R>> ShortArray.mapIndexedTo(destination: C, transform: (index: Int, Short) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in R>> IntArray.mapIndexedTo(destination: C, transform: (index: Int, Int) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in R>> LongArray.mapIndexedTo(destination: C, transform: (index: Int, Long) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in R>> FloatArray.mapIndexedTo(destination: C, transform: (index: Int, Float) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Applies the given

[transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in R>> DoubleArray.mapIndexedTo(destination: C, transform: (index: Int, Double) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in R>> BooleanArray.mapIndexedTo(destination: C, transform: (index: Int, Boolean) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <R, C : MutableCollection<in R>> CharArray.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Returns a list containing only the non-null results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.mapNotNull\n */\npublic inline fun <T, R : Any> Array<out T>.mapNotNull(transform: (T) -> R?): List<R> {\n return mapNotNullTo(ArrayList<R>(), transform)\n}\n\n/**\n * Applies the given [transform] function to each element in the original array\n * and appends only the non-null results to the given [destination].\n */\npublic inline fun <T, R : Any, C : MutableCollection<in R>> Array<out T>.mapNotNullTo(destination: C, transform: (T) -> R?): C {\n forEach { element -> transform(element)?.let { destination.add(it) } }\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */\npublic inline fun <T, R, C : MutableCollection<in R>> Array<out T>.mapTo(destination: C, transform: (T) -> R): C {\n for (item in this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> ByteArray.mapTo(destination: C, transform: (Byte) -> R): C {\n for (item in this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> ShortArray.mapTo(destination: C, transform: (Short) -> R): C {\n for (item in this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> IntArray.mapTo(destination: C, transform: (Int) -> R): C {\n for (item in this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> LongArray.mapTo(destination: C, transform: (Long) -> R): C {\n for (item in this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> FloatArray.mapTo(destination: C, transform: (Float) -> R): C {\n for (item in this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> DoubleArray.mapTo(destination: C, transform: (Double) -> R): C {\n for (item in this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> BooleanArray.mapTo(destination: C, transform: (Boolean)

-> R): C {\n   for (item in this)\n   destination.add(transform(item))\n   return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> CharArray.mapTo(destination: C, transform: (Char) -> R): C {\n   for (item in this)\n destination.add(transform(item))\n   return destination\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */\npublic fun <T> Array<out T>.withIndex(): Iterable<IndexedValue<T>> {\n   return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */\npublic fun ByteArray.withIndex(): Iterable<IndexedValue<Byte>> {\n   return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */\npublic fun ShortArray.withIndex(): Iterable<IndexedValue<Short>> {\n   return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */\npublic fun IntArray.withIndex(): Iterable<IndexedValue<Int>> {\n   return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */\npublic fun LongArray.withIndex(): Iterable<IndexedValue<Long>> {\n   return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */\npublic fun FloatArray.withIndex(): Iterable<IndexedValue<Float>> {\n   return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */\npublic fun DoubleArray.withIndex(): Iterable<IndexedValue<Double>> {\n   return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */\npublic fun BooleanArray.withIndex(): Iterable<IndexedValue<Boolean>> {\n   return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */\npublic fun CharArray.withIndex(): Iterable<IndexedValue<Char>> {\n   return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a list containing only distinct elements from the given array.\n * \n * Among equal elements of the given array, only the first one will be present in the resulting list.\n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun <T> Array<out T>.distinct(): List<T> {\n   return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun ByteArray.distinct(): List<Byte> {\n   return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same order as they were in
the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun ShortArray.distinct(): List<Short> {\n   return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun IntArray.distinct(): List<Int> {\n   return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun LongArray.distinct(): List<Long> {\n   return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing

only distinct elements from the
given array.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun FloatArray.distinct(): List<Float> {\n return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun DoubleArray.distinct(): List<Double> {\n return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun BooleanArray.distinct(): List<Boolean> {\n return
this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only distinct elements from the given array.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun CharArray.distinct(): List<Char> {\n return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * Among elements of the given array with equal keys, only the first one will be present in the resulting list.\n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <T, K> Array<out T>.distinctBy(selector: (T) -> K): List<T> {\n val set = HashSet<K>()\n val list = ArrayList<T>()\n for (e
in this) {\n val key = selector(e)\n if (set.add(key))\n list.add(e)\n }\n return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K> ByteArray.distinctBy(selector: (Byte) -> K): List<Byte> {\n val set = HashSet<K>()\n val list = ArrayList<Byte>()\n for (e in this) {\n val key = selector(e)\n if (set.add(key))\n list.add(e)\n }\n return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K> ShortArray.distinctBy(selector: (Short) -> K): List<Short> {\n val set = HashSet<K>()\n val list = ArrayList<Short>()\n for (e in this) {\n val key = selector(e)\n if (set.add(key))\n list.add(e)\n }\n return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K> IntArray.distinctBy(selector: (Int) -> K): List<Int> {\n val set = HashSet<K>()\n val list = ArrayList<Int>()\n for (e in this) {\n val key = selector(e)\n if (set.add(key))\n list.add(e)\n }\n return list\n}\n\n/**\n * Returns a list containing only elements from
the given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K> LongArray.distinctBy(selector: (Long) -> K): List<Long> {\n val set = HashSet<K>()\n val list = ArrayList<Long>()\n for (e in this) {\n val key = selector(e)\n if (set.add(key))\n list.add(e)\n }\n return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K>

FloatArray.distinctBy(selector: (Float) -> K): List<Float> {\n val set = HashSet<K>()\n val list = ArrayList<Float>()\n for (e in this) {\n val key = selector(e)\n if (set.add(key))\n list.add(e)\n }\n return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K> DoubleArray.distinctBy(selector: (Double) -> K): List<Double> {\n val set = HashSet<K>()\n val list = ArrayList<Double>()\n for (e in this) {\n val key = selector(e)\n if (set.add(key))\n list.add(e)\n }\n return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they were
in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K> BooleanArray.distinctBy(selector: (Boolean) -> K): List<Boolean> {\n val set = HashSet<K>()\n val list = ArrayList<Boolean>()\n for (e in this) {\n val key = selector(e)\n if (set.add(key))\n list.add(e)\n }\n return list\n}\n\n/**\n * Returns a list containing only elements from the given array\n * having distinct keys returned by the given [selector] function.\n * \n * The elements in the resulting list are in the same order as they were in the source array.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <K> CharArray.distinctBy(selector: (Char) -> K): List<Char> {\n val set = HashSet<K>()\n val list = ArrayList<Char>()\n for (e in this) {\n val key = selector(e)\n if (set.add(key))\n list.add(e)\n }\n return
list\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set containing all elements that are contained at least in one of these collections use [union].\n */\npublic infix fun <T> Array<out T>.intersect(other: Iterable<T>): Set<T> {\n val set = this.toMutableSet()\n set.retainAll(other)\n return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set containing all elements that are contained at least in one of these collections use [union].\n */\npublic infix fun ByteArray.intersect(other: Iterable<Byte>): Set<Byte> {\n val set = this.toMutableSet()\n set.retainAll(other)\n return set\n}\n\n/**\n * Returns a set containing
all elements that are contained by both this array and the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set containing all elements that are contained at least in one of these collections use [union].\n */\npublic infix fun ShortArray.intersect(other: Iterable<Short>): Set<Short> {\n val set = this.toMutableSet()\n set.retainAll(other)\n return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set containing all elements that are contained at least in one of these collections use [union].\n */\npublic infix fun IntArray.intersect(other: Iterable<Int>): Set<Int> {\n val set = this.toMutableSet()\n set.retainAll(other)\n return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this
array and the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set containing all elements that are contained at least in one of these collections use [union].\n */\npublic infix fun LongArray.intersect(other: Iterable<Long>): Set<Long> {\n val set = this.toMutableSet()\n set.retainAll(other)\n return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set containing all elements that are contained at least in one of these collections use [union].\n */\npublic infix fun FloatArray.intersect(other: Iterable<Float>): Set<Float> {\n val set = this.toMutableSet()\n set.retainAll(other)\n return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the specified collection.\n

```
 * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set containing all elements that are contained at least in one of these collections use [union].\n */\npublic infix fun DoubleArray.intersect(other: Iterable<Double>): Set<Double> {\n    val set = this.toMutableSet()\n    set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n * \n * To get a set containing all elements that are contained at least in one of these collections use [union].\n */\npublic infix fun BooleanArray.intersect(other: Iterable<Boolean>): Set<Boolean> {\n    val set = this.toMutableSet()\n    set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are contained by both this array and the specified collection.\n * \n * The returned set preserves
 the element iteration order of the original array.\n * \n * To get a set containing all elements that are contained at least in one of these collections use [union].\n */\npublic infix fun CharArray.intersect(other: Iterable<Char>): Set<Char> {\n    val set = this.toMutableSet()\n    set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are contained by this array and not contained by the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic infix fun <T> Array<out T>.subtract(other: Iterable<T>): Set<T> {\n    val set = this.toMutableSet()\n    set.removeAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are contained by this array and not contained by the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic infix fun ByteArray.subtract(other: Iterable<Byte>): Set<Byte> {\n
    val set = this.toMutableSet()\n    set.removeAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are contained by this array and not contained by the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic infix fun ShortArray.subtract(other: Iterable<Short>): Set<Short> {\n    val set = this.toMutableSet()\n    set.removeAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are contained by this array and not contained by the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic infix fun IntArray.subtract(other: Iterable<Int>): Set<Int> {\n    val set = this.toMutableSet()\n    set.removeAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are contained by this array and not contained by the specified collection.\n * \n * The returned set preserves the element
 iteration order of the original array.\n */\npublic infix fun LongArray.subtract(other: Iterable<Long>): Set<Long> {\n    val set = this.toMutableSet()\n    set.removeAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are contained by this array and not contained by the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic infix fun FloatArray.subtract(other: Iterable<Float>): Set<Float> {\n    val set = this.toMutableSet()\n    set.removeAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are contained by this array and not contained by the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic infix fun DoubleArray.subtract(other: Iterable<Double>): Set<Double> {\n    val set = this.toMutableSet()\n    set.removeAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements
 that are contained by this array and not contained by the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic infix fun BooleanArray.subtract(other: Iterable<Boolean>): Set<Boolean> {\n    val set = this.toMutableSet()\n    set.removeAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are contained by this array and not contained by the specified collection.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic infix fun CharArray.subtract(other: Iterable<Char>): Set<Char> {\n    val set = this.toMutableSet()\n    set.removeAll(other)\n    return set\n}\n\n/**\n * Returns a new [MutableSet] containing all distinct elements from the given array.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun <T> Array<out T>.toMutableSet(): MutableSet<T> {\n    return toCollection(LinkedHashSet<T>(mapCapacity(size)))\n}\n\n/**\n
 * Returns a new [MutableSet] containing all distinct elements from the given array.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun ByteArray.toMutableSet(): MutableSet<Byte> {\n    return toCollection(LinkedHashSet<Byte>(mapCapacity(size)))\n}\n\n/**\n * Returns a
```

new [MutableSet] containing all distinct elements from the given array.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun ShortArray.toMutableSet(): MutableSet<Short> {\n return toCollection(LinkedHashSet<Short>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [MutableSet] containing all distinct elements from the given array.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun IntArray.toMutableSet(): MutableSet<Int> {\n return toCollection(LinkedHashSet<Int>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [MutableSet] containing all distinct elements from the given array.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun LongArray.toMutableSet(): MutableSet<Long> {\n return toCollection(LinkedHashSet<Long>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [MutableSet] containing all distinct elements from the given array.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun FloatArray.toMutableSet(): MutableSet<Float> {\n return toCollection(LinkedHashSet<Float>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [MutableSet] containing all distinct elements from the given array.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun DoubleArray.toMutableSet(): MutableSet<Double> {\n return toCollection(LinkedHashSet<Double>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [MutableSet] containing all distinct elements from the given array.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun BooleanArray.toMutableSet(): MutableSet<Boolean> {\n return toCollection(LinkedHashSet<Boolean>(mapCapacity(size)))\n}\n\n/**\n * Returns a new [MutableSet] containing all distinct elements from the given array.\n * \n * The returned set preserves the element iteration order of the original array.\n */\npublic fun CharArray.toMutableSet(): MutableSet<Char> {\n return toCollection(LinkedHashSet<Char>(mapCapacity(size.coerceAtMost(128))))\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun <T> Array<out T>.union(other: Iterable<T>): Set<T> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun ByteArray.union(other: Iterable<Byte>): Set<Byte> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun ShortArray.union(other: Iterable<Short>): Set<Short> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun IntArray.union(other: Iterable<Int>): Set<Int> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun LongArray.union(other: Iterable<Long>): Set<Long> {\n val set = this.toMutableSet()\n set.addAll(other)\n return set\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those

elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun FloatArray.union(other: Iterable<Float>): Set<Float> {\n   val set = this.toMutableSet()\n   set.addAll(other)\n   return set\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of
the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun DoubleArray.union(other: Iterable<Double>): Set<Double> {\n   val set = this.toMutableSet()\n   set.addAll(other)\n   return set\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun BooleanArray.union(other: Iterable<Boolean>): Set<Boolean> {\n   val set = this.toMutableSet()\n   set.addAll(other)\n   return set\n}\n\n/**\n * Returns a set containing
all distinct elements from both collections.\n * \n * The returned set preserves the element iteration order of the original array.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun CharArray.union(other: Iterable<Char>): Set<Char> {\n   val set = this.toMutableSet()\n   set.addAll(other)\n   return set\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the array contains no elements, the function returns `true`\n * because there are no elements in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in [\"Vacuous truth\"](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun <T> Array<out T>.all(predicate:
(T) -> Boolean): Boolean {\n   for (element in this) if (!predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the array contains no elements, the function returns `true`\n * because there are no elements in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in [\"Vacuous truth\"](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun ByteArray.all(predicate: (Byte) -> Boolean): Boolean {\n   for (element in this) if (!predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the array contains no elements, the function returns `true`\n * because there are no elements in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in [\"Vacuous
truth\"](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun ShortArray.all(predicate: (Short) -> Boolean): Boolean {\n   for (element in this) if (!predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the array contains no elements, the function returns `true`\n * because there are no elements in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in [\"Vacuous truth\"](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun IntArray.all(predicate: (Int) -> Boolean): Boolean {\n   for (element in this) if (!predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the array contains
no elements, the function returns `true`\n * because there are no elements in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in [\"Vacuous
truth\"](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun LongArray.all(predicate: (Long) -> Boolean): Boolean {\n   for (element in this) if (!predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the array contains no elements, the function returns `true`\n * because there are no elements in it that _do not_ match the predicate.\n * See a more detailed explanation

of this logic concept in [\"Vacuous truth\"](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun FloatArray.all(predicate: (Float) -> Boolean): Boolean {\n for (element
in this) if (!predicate(element)) return false\n return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the array contains no elements, the function returns `true`\n * because there are no elements in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in [\"Vacuous truth\"](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun DoubleArray.all(predicate: (Double) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return false\n return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the array contains no elements, the function returns `true`\n * because there are no elements in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in [\"Vacuous truth\"](https://en.wikipedia.org/wiki/Vacuous_truth)
article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun BooleanArray.all(predicate: (Boolean) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return false\n return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the array contains no elements, the function returns `true`\n * because there are no elements in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in [\"Vacuous truth\"](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun CharArray.all(predicate: (Char) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return false\n return true\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\npublic fun <T>
Array<out T>.any(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\npublic fun ByteArray.any(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\npublic fun ShortArray.any(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\npublic fun IntArray.any(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\npublic fun LongArray.any(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\npublic fun FloatArray.any(): Boolean
{\n return !isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\npublic fun DoubleArray.any(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\npublic fun BooleanArray.any(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\npublic fun CharArray.any(): Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun <T> Array<out T>.any(predicate: (T) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return false\n}\n\n/**\n * Returns `true` if at least one element matches
the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun ByteArray.any(predicate: (Byte) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun ShortArray.any(predicate: (Short) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n *

@sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun IntArray.any(predicate: (Int) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun LongArray.any(predicate: (Long) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun FloatArray.any(predicate: (Float) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun DoubleArray.any(predicate: (Double) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun BooleanArray.any(predicate: (Boolean) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun CharArray.any(predicate: (Char) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return false\n}\n\n/**\n * Returns the number of elements in this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.count(): Int {\n return size\n}\n\n/**\n * Returns the number of elements in this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.count(): Int {\n return size\n}\n\n/**\n * Returns the number of elements in this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.count(): Int {\n return size\n}\n\n/**\n * Returns the number of elements in this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.count(): Int {\n return size\n}\n\n/**\n * Returns the number of elements in this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.count(): Int {\n return size\n}\n\n/**\n * Returns the number of elements in this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.count(): Int {\n return size\n}\n\n/**\n * Returns the number of elements in this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.count(): Int {\n return size\n}\n\n/**\n * Returns the number of elements in this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.count(): Int {\n return size\n}\n\n/**\n * Returns the number of elements in this array.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.count(): Int {\n return size\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n */\npublic inline fun <T> Array<out T>.count(predicate: (T) -> Boolean): Int {\n var count = 0\n for (element in this) if (predicate(element)) ++count\n return count\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n */\npublic inline fun ByteArray.count(predicate: (Byte) -> Boolean): Int {\n var count = 0\n for (element in this) if (predicate(element)) ++count\n return count\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n */\npublic inline fun ShortArray.count(predicate: (Short) -> Boolean): Int {\n var count = 0\n for (element in this) if (predicate(element)) ++count\n return count\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n */\npublic inline fun IntArray.count(predicate: (Int) -> Boolean): Int {\n var count = 0\n for (element in this) if (predicate(element)) ++count\n return count\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n */\npublic inline fun LongArray.count(predicate: (Long) -> Boolean): Int {\n var count = 0\n for (element in this) if (predicate(element)) ++count\n return count\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n */\npublic inline fun FloatArray.count(predicate: (Float) -> Boolean): Int {\n var count = 0\n for (element in this) if (predicate(element)) ++count\n return count\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n */\npublic inline fun DoubleArray.count(predicate: (Double) -> Boolean): Int {\n var count = 0\n for (element in this) if (predicate(element)) ++count\n return count\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n */\npublic inline fun BooleanArray.count(predicate: (Boolean) -> Boolean): Int {\n var count = 0\n for (element in this) if

(predicate(element)) ++count\n   return count\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n */\npublic inline fun CharArray.count(predicate: (Char) -> Boolean): Int {\n   var count = 0\n   for (element in this) if (predicate(element)) ++count\n   return count\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <T, R> Array<out T>.fold(initial: R, operation: (acc: R, T) -> R): R {\n   var accumulator = initial\n   for (element in this) accumulator = operation(accumulator, element)\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R> ByteArray.fold(initial: R, operation: (acc: R, Byte) -> R): R {\n   var accumulator = initial\n   for (element in this) accumulator = operation(accumulator, element)\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R> ShortArray.fold(initial: R, operation: (acc: R, Short) -> R): R {\n   var accumulator = initial\n   for (element in this) accumulator = operation(accumulator, element)\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R> IntArray.fold(initial: R, operation: (acc: R, Int) -> R): R {\n   var accumulator = initial\n   for (element in this) accumulator = operation(accumulator, element)\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R> LongArray.fold(initial: R, operation: (acc: R, Long) -> R): R {\n   var accumulator = initial\n   for (element in this) accumulator = operation(accumulator, element)\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R> FloatArray.fold(initial: R, operation: (acc: R, Float) -> R): R {\n   var accumulator = initial\n   for (element in this) accumulator = operation(accumulator, element)\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R> DoubleArray.fold(initial: R, operation: (acc: R, Double) -> R): R {\n   var accumulator = initial\n   for (element in this) accumulator = operation(accumulator, element)\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <R> BooleanArray.fold(initial: R, operation: (acc: R, Boolean) -> R): R {\n   var accumulator = initial\n   for (element in this) accumulator = operation(accumulator, element)\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline

fun <R> CharArray.fold(initial: R, operation: (acc: R, Char) -> R): R {\n var accumulator = initial\n for (element in this) accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n */\npublic inline fun <T, R> Array<out T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {\n var index = 0\n var accumulator = initial\n for (element in this) accumulator = operation(index++, accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n */\npublic inline fun <R> ByteArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): R {\n var index = 0\n var accumulator = initial\n for (element in this) accumulator = operation(index++, accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n */\npublic inline fun <R> ShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): R {\n var index = 0\n var accumulator = initial\n for (element in this) accumulator = operation(index++, accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n */\npublic inline fun <R> IntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): R {\n var index = 0\n var accumulator = initial\n for (element in this) accumulator = operation(index++, accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n */\npublic inline fun <R> LongArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Long) -> R): R {\n var index = 0\n var accumulator = initial\n for (element in this) accumulator = operation(index++, accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n */\npublic inline fun <R> FloatArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): R {\n var index = 0\n var accumulator = initial\n for (element in this) accumulator = operation(index++, accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n */\npublic inline fun <R> DoubleArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): R {\n var index = 0\n var accumulator = initial\n for (element in this) accumulator = operation(index++, accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the

specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n */\npublic inline fun <R> BooleanArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): R {\n var index = 0\n var accumulator = initial\n for (element in this) accumulator = operation(index++, accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n */\npublic inline fun <R> CharArray.foldIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): R {\n var index = 0\n var accumulator = initial\n for (element in this) accumulator = operation(index++, accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <T, R> Array<out T>.foldRight(initial: R, operation: (T, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> ByteArray.foldRight(initial: R, operation: (Byte, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> ShortArray.foldRight(initial: R, operation: (Short, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> IntArray.foldRight(initial: R, operation: (Int, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> LongArray.foldRight(initial: R, operation: (Long, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> FloatArray.foldRight(initial: R, operation: (Float, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun

<R> DoubleArray.foldRight(initial: R, operation: (Double, acc:
R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> BooleanArray.foldRight(initial: R, operation: (Boolean, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns
the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> CharArray.foldRight(initial: R, operation: (Char, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <T, R> Array<out T>.foldRightIndexed(initial: R, operation: (index: Int, T, acc:
R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> ByteArray.foldRightIndexed(initial: R, operation: (index: Int, Byte, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with
[initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> ShortArray.foldRightIndexed(initial: R, operation: (index: Int, Short, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function
that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> IntArray.foldRightIndexed(initial: R, operation: (index: Int, Int, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> LongArray.foldRightIndexed(initial: R, operation: (index: Int, Long, acc: R) -> R): R
{\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates

value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> FloatArray.foldRightIndexed(initial: R, operation: (index: Int, Float, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial]
value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> DoubleArray.foldRightIndexed(initial: R, operation: (index: Int, Double, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function
that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> BooleanArray.foldRightIndexed(initial: R, operation: (index: Int, Boolean, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> CharArray.foldRightIndexed(initial: R, operation: (index: Int, Char, acc: R) -> R): R
{\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline fun <T> Array<out T>.forEach(action: (T) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline fun ByteArray.forEach(action: (Byte) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline fun ShortArray.forEach(action: (Short) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline fun IntArray.forEach(action: (Int) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline fun LongArray.forEach(action:
(Long) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline fun FloatArray.forEach(action: (Float) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline fun DoubleArray.forEach(action: (Double) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline fun BooleanArray.forEach(action: (Boolean) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n */\npublic inline fun CharArray.forEach(action: (Char) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs
the action on the element.\n */\npublic inline fun <T> Array<out T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {\n var index = 0\n for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\npublic inline fun

ByteArray.forEachIndexed(action: (index: Int, Byte) -> Unit): Unit {\n   var index = 0\n   for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\npublic inline fun ShortArray.forEachIndexed(action: (index: Int, Short) -> Unit): Unit {\n   var index = 0\n   for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\npublic inline fun IntArray.forEachIndexed(action: (index: Int, Int) -> Unit): Unit {\n   var index = 0\n   for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\npublic inline fun LongArray.forEachIndexed(action: (index: Int, Long) -> Unit): Unit {\n   var index = 0\n   for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\npublic inline fun FloatArray.forEachIndexed(action: (index: Int, Float) -> Unit): Unit {\n   var index = 0\n   for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\npublic inline fun DoubleArray.forEachIndexed(action: (index: Int, Double) -> Unit): Unit {\n   var index = 0\n   for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\npublic inline fun BooleanArray.forEachIndexed(action: (index: Int, Boolean) -> Unit): Unit {\n   var index = 0\n   for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\npublic inline fun CharArray.forEachIndexed(action: (index: Int, Char) -> Unit): Unit {\n   var index = 0\n   for (item in this) action(index++, item)\n}\n\n/**\n * Returns the largest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun Array<out Double>.max(): Double {\n   if (isEmpty()) throw NoSuchElementException()\n   var max = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   max = maxOf(max, e)\n   }\n   return max\n}\n\n/**\n * Returns the largest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun Array<out Float>.max(): Float {\n   if (isEmpty()) throw NoSuchElementException()\n   var max = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   max = maxOf(max, e)\n   }\n   return max\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun <T : Comparable<T>> Array<out T>.max(): T {\n   if (isEmpty()) throw NoSuchElementException()\n   var max = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (max < e) max = e\n   }\n   return max\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun ByteArray.max(): Byte {\n   if (isEmpty()) throw NoSuchElementException()\n   var max = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (max < e) max = e\n   }\n   return max\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOA

DS\")\npublic fun ShortArray.max(): Short {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun IntArray.max(): Int {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun LongArray.max(): Long {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the largest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun FloatArray.max(): Float {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n * Returns the largest
element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun DoubleArray.max(): Double {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun CharArray.max(): Char {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n * @throws NoSuchElementException
if the array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <T, R : Comparable<R>> Array<out T>.maxBy(selector: (T) -> R): T {\n if (isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic
inline fun <R : Comparable<R>> ByteArray.maxBy(selector: (Byte) -> R): Byte {\n if (isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <R : Comparable<R>> ShortArray.maxBy(selector: (Short) -> R): Short {\n if

(isEmpty()) throw NoSuchElementException()\n   var maxElem = this[0]\n   val lastIndex
 = this.lastIndex\n   if (lastIndex == 0) return maxElem\n   var maxValue = selector(maxElem)\n   for (i in
1..lastIndex) {\n   val e = this[i]\n   val v = selector(e)\n   if (maxValue < v) {\n   maxElem = e\n
 maxValue = v\n   }\n   }\n   return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value
of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.maxBy\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow\")\n@Suppress(\"CONFLICTING_OVERL
OADS\")\npublic inline fun <R : Comparable<R>> IntArray.maxBy(selector: (Int) -> R): Int {\n   if (isEmpty())
throw NoSuchElementException()\n   var maxElem = this[0]\n   val lastIndex = this.lastIndex\n   if (lastIndex ==
0) return maxElem\n   var maxValue = selector(maxElem)\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   val
v = selector(e)\n   if (maxValue
 < v) {\n   maxElem = e\n   maxValue = v\n   }\n   }\n   return maxElem\n}\n\n/**\n * Returns the
first element yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the
array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow\")\n@Suppress(\"CONFLICTING_OVERL
OADS\")\npublic inline fun <R : Comparable<R>> LongArray.maxBy(selector: (Long) -> R): Long {\n   if
(isEmpty()) throw NoSuchElementException()\n   var maxElem = this[0]\n   val lastIndex = this.lastIndex\n   if
(lastIndex == 0) return maxElem\n   var maxValue = selector(maxElem)\n   for (i in 1..lastIndex) {\n   val e =
this[i]\n   val v = selector(e)\n   if (maxValue < v) {\n   maxElem = e\n   maxValue = v\n   }\n
}\n   return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n
 * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.maxBy\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow\")\n@Suppress(\"CONFLICTING_OVERL
OADS\")\npublic inline fun <R : Comparable<R>> FloatArray.maxBy(selector: (Float) -> R): Float {\n   if
(isEmpty()) throw NoSuchElementException()\n   var maxElem = this[0]\n   val lastIndex = this.lastIndex\n   if
(lastIndex == 0) return maxElem\n   var maxValue = selector(maxElem)\n   for (i in 1..lastIndex) {\n   val e =
this[i]\n   val v = selector(e)\n   if (maxValue < v) {\n   maxElem = e\n   maxValue = v\n   }\n
}\n   return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n
* @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.maxBy\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow\")\n@Suppress(\"CONFLICTING_OVERL
OADS\")\npublic
 inline fun <R : Comparable<R>> DoubleArray.maxBy(selector: (Double) -> R): Double {\n   if (isEmpty()) throw
NoSuchElementException()\n   var maxElem = this[0]\n   val lastIndex = this.lastIndex\n   if (lastIndex == 0)
return maxElem\n   var maxValue = selector(maxElem)\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   val v
= selector(e)\n   if (maxValue < v) {\n   maxElem = e\n   maxValue = v\n   }\n   }\n   return
maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n * @throws
NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.maxBy\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow\")\n@Suppress(\"CONFLICTING_OVERL
OADS\")\npublic inline fun <R : Comparable<R>> BooleanArray.maxBy(selector: (Boolean) -> R): Boolean {\n
if (isEmpty()) throw NoSuchElementException()\n   var
 maxElem = this[0]\n   val lastIndex = this.lastIndex\n   if (lastIndex == 0) return maxElem\n   var maxValue =
selector(maxElem)\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   val v = selector(e)\n   if (maxValue < v)
{\n   maxElem = e\n   maxValue = v\n   }\n   }\n   return maxElem\n}\n\n/**\n * Returns the first
element yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the array is
empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow\")\n@Suppress(\"CONFLICTING_OVERL

OADS\")\npublic inline fun <R : Comparable<R>> CharArray.maxBy(selector: (Char) -> R): Char {\n if (isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n
val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <T, R : Comparable<R>> Array<out T>.maxByOrNull(selector: (T) -> R): T? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> ByteArray.maxByOrNull(selector: (Byte) -> R): Byte? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> ShortArray.maxByOrNull(selector: (Short) -> R): Short? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue
= selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> IntArray.maxByOrNull(selector: (Int) -> R): Int? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function
or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> LongArray.maxByOrNull(selector: (Long) -> R): Long? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> FloatArray.maxByOrNull(selector: (Float) -> R): Float? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val
lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> DoubleArray.maxByOrNull(selector: (Double) -> R): Double? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v)

{\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> BooleanArray.maxByOrNull(selector: (Boolean) -> R): Boolean? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> CharArray.maxByOrNull(selector: (Char) -> R): Char? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOf(selector: (T) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOf(selector: (Byte) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOf(selector: (Short) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOf(selector: (Int) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOf(selector: (Long) ->

Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOf(selector: (Float) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element
in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.maxOf(selector: (Double) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun BooleanArray.maxOf(selector: (Boolean) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.maxOf(selector: (Char) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOf(selector: (T) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n
* \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOf(selector: (Byte) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to

each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOf(selector: (Short) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOf(selector: (Int) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex)\n {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOf(selector: (Long) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values
produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOf(selector: (Float) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.maxOf(selector: (Double) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.maxOf(selector: (Boolean) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v
= selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws

NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.maxOf(selector: (Char) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException
if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Array<out T>.maxOf(selector: (T) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ByteArray.maxOf(selector: (Byte) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ShortArray.maxOf(selector: (Short) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element
in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> IntArray.maxOf(selector: (Int) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> LongArray.maxOf(selector: (Long) -> R):
R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> FloatArray.maxOf(selector: (Float) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values

produced
by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
DoubleArray.maxOf(selector: (Double) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var
maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (maxValue < v) {\n
        maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array.\n * \n * @throws
NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R
: Comparable<R>> BooleanArray.maxOf(selector: (Boolean) -> R): R {\n   if (isEmpty()) throw
NoSuchElementException()\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v =
selector(this[i])\n   if (maxValue < v) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n *
Returns the largest value among all values produced by [selector] function\n * applied to each element in the
array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharArray.maxOf(selector: (Char) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var
maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (maxValue < v) {\n
        maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n
 * Returns the largest value among all values produced by [selector] function\n * applied to each element in the
array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOfOrNull(selector:
(T) -> Double): Double? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in
1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
 inline fun ByteArray.maxOfOrNull(selector: (Byte) -> Double): Double? {\n   if (isEmpty()) return null\n   var
maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue =
maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOfOrNull(selector: (Short) -
> Double): Double? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i
 in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOfOrNull(selector: (Int) ->

Double): Double? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n *
 If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOfOrNull(selector: (Long) -> Double): Double? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOfOrNull(selector:
 (Float) -> Double): Double? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.maxOfOrNull(selector:
(Double) -> Double): Double? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values
 produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.maxOfOrNull(selector:
(Boolean) -> Double): Double? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
 inline fun CharArray.maxOfOrNull(selector: (Char) -> Double): Double? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.maxOfOrNull(selector:
(T) -> Float): Float? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n
        maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.maxOfOrNull(selector: (Byte) -> Float): Float? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.maxOfOrNull(selector: (Short) -> Float): Float? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.maxOfOrNull(selector: (Int) -> Float): Float? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.maxOfOrNull(selector: (Long) -> Float): Float? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.maxOfOrNull(selector: (Float) -> Float): Float? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.maxOfOrNull(selector: (Double) -> Float): Float? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.maxOfOrNull(selector: (Boolean) -> Float): Float? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the

returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.maxOfOrNull(selector: (Char) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <T, R : Comparable<R>> Array<out T>.maxOfOrNull(selector: (T) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ByteArray.maxOfOrNull(selector: (Byte) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n
* Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ShortArray.maxOfOrNull(selector: (Short) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
IntArray.maxOfOrNull(selector: (Int) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> LongArray.maxOfOrNull(selector: (Long) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector]
function\n * applied to each element in the array or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> FloatArray.maxOfOrNull(selector: (Float) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>

```
DoubleArray.maxOfOrNull(selector: (Double) -> R): R? {\n    if
 (isEmpty()) return null\n    var maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v =
selector(this[i])\n        if (maxValue < v) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n *
Returns the largest value among all values produced by [selector] function\n * applied to each element in the array
or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
BooleanArray.maxOfOrNull(selector: (Boolean) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue =
selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n
maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value among all values produced
by [selector] function\n * applied to each element in the array
 or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharArray.maxOfOrNull(selector: (Char) -> R): R? {\n    if (isEmpty()) return null\n    var maxValue =
selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if (maxValue < v) {\n
maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n *
@throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out
T>.maxOfWith(comparator: Comparator<in R>, selector: (T)
 -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue = selector(this[0])\n    for (i in
1..lastIndex) {\n        val v = selector(this[i])\n        if (comparator.compare(maxValue, v) < 0) {\n            maxValue
= v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n *
@throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.maxOfWith(comparator:
Comparator<in R>, selector: (Byte) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue
 = v\n        }\n    }\n    return maxValue\n}\n\n/**\n * Returns the largest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n *
@throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.maxOfWith(comparator:
Comparator<in R>, selector: (Short) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var
maxValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n *
Returns the largest value according to the provided [comparator]\n * among all values produced by [selector]
function applied to each element in the array.\n * \n * @throws
 NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.maxOfWith(comparator:
Comparator<in R>, selector: (Int) -> R): R {\n    if (isEmpty()) throw NoSuchElementException()\n    var maxValue
= selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        if
(comparator.compare(maxValue, v) < 0) {\n            maxValue = v\n        }\n    }\n    return maxValue\n}\n\n/**\n *
```

Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.maxOfWith(comparator: Comparator<in R>, selector: (Long) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.maxOfWith(comparator: Comparator<in R>, selector: (Float) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.maxOfWith(comparator: Comparator<in R>, selector: (Double) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.maxOfWith(comparator: Comparator<in R>, selector: (Boolean) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.maxOfWith(comparator: Comparator<in R>, selector: (Char) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Byte) -> R): R? {\n if (isEmpty()) return

null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to
 each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Short) -> R): R? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.maxOfWithOrNull(comparator:
 Comparator<in R>, selector: (Int) -> R): R? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Long) -> R): R? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n
  }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Float) -> R): R? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
 inline fun <R> DoubleArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Double) -> R): R? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Boolean) -> R): R? {\n   if (isEmpty()) return null\n   var
 maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n *

Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic fun Array<out Double>.maxOrNull(): Double? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic fun Array<out Float>.maxOrNull(): Float? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T : Comparable<T>> Array<out T>.maxOrNull(): T? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun ByteArray.maxOrNull(): Byte? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun ShortArray.maxOrNull(): Short? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun IntArray.maxOrNull(): Int? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun LongArray.maxOrNull(): Long? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic fun FloatArray.maxOrNull(): Float? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic fun DoubleArray.maxOrNull(): Double? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun CharArray.maxOrNull(): Char? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun <T> Array<out T>.maxWith(comparator: Comparator<in T>): T {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun ByteArray.maxWith(comparator: Comparator<in Byte>): Byte {\n if (isEmpty()) throw

NoSuchElementException()\n   var max = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (comparator.compare(max, e) < 0) max = e\n   }\n   return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun ShortArray.maxWith(comparator: Comparator<in Short>): Short {\n   if (isEmpty()) throw NoSuchElementException()\n   var max = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (comparator.compare(max, e) < 0) max = e\n   }\n   return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun IntArray.maxWith(comparator: Comparator<in Int>): Int {\n   if (isEmpty()) throw NoSuchElementException()\n   var max = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (comparator.compare(max, e) < 0) max = e\n   }\n   return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun LongArray.maxWith(comparator: Comparator<in Long>): Long {\n   if (isEmpty()) throw NoSuchElementException()\n   var max = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (comparator.compare(max, e) < 0) max = e\n   }\n   return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun FloatArray.maxWith(comparator: Comparator<in Float>): Float {\n   if (isEmpty()) throw NoSuchElementException()\n   var max = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (comparator.compare(max, e) < 0) max = e\n   }\n   return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun DoubleArray.maxWith(comparator: Comparator<in Double>): Double {\n   if (isEmpty()) throw NoSuchElementException()\n   var max = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (comparator.compare(max, e) < 0) max = e\n   }\n   return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun BooleanArray.maxWith(comparator: Comparator<in Boolean>): Boolean {\n   if (isEmpty()) throw NoSuchElementException()\n   var max = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (comparator.compare(max, e) < 0) max = e\n   }\n   return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic
fun CharArray.maxWith(comparator: Comparator<in Char>): Char {\n   if (isEmpty()) throw NoSuchElementException()\n   var max = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (comparator.compare(max, e) < 0) max = e\n   }\n   return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\npublic fun <T> Array<out T>.maxWithOrNull(comparator: Comparator<in T>): T? {\n   if (isEmpty()) return null\n   var max = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if

(comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun ByteArray.maxWithOrNull(comparator: Comparator<in Byte>): Byte? {\n if (isEmpty())
return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun ShortArray.maxWithOrNull(comparator: Comparator<in Short>): Short? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun IntArray.maxWithOrNull(comparator: Comparator<in Int>): Int? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e)
< 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun LongArray.maxWithOrNull(comparator: Comparator<in Long>): Long? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun FloatArray.maxWithOrNull(comparator: Comparator<in Float>): Float? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator]
or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun DoubleArray.maxWithOrNull(comparator: Comparator<in Double>): Double? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun BooleanArray.maxWithOrNull(comparator: Comparator<in Boolean>): Boolean? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun CharArray.maxWithOrNull(comparator:
Comparator<in Char>): Char? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun Array<out Double>.min(): Double {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun Array<out Float>.min():
Float {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun <T : Comparable<T>> Array<out T>.min(): T {\n if (isEmpty()) throw

NoSuchElementException()\n   var min = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (min > e) min = e\n   }\n   return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun ByteArray.min(): Byte {\n   if (isEmpty()) throw NoSuchElementException()\n   var min = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (min > e) min = e\n   }\n   return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun ShortArray.min(): Short {\n   if (isEmpty()) throw NoSuchElementException()\n   var min = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (min > e) min = e\n   }\n   return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun IntArray.min(): Int {\n   if (isEmpty()) throw NoSuchElementException()\n   var min = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (min > e) min = e\n   }\n   return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException\n if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun LongArray.min(): Long {\n   if (isEmpty()) throw NoSuchElementException()\n   var min = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (min > e) min = e\n   }\n   return min\n}\n\n/**\n * Returns the smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun FloatArray.min(): Float {\n   if (isEmpty()) throw NoSuchElementException()\n   var min = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   min = minOf(min, e)\n   }\n   return min\n}\n\n/**\n * Returns the smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun DoubleArray.min(): Double {\n   if (isEmpty()) throw NoSuchElementException()\n   var min = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   min = minOf(min, e)\n   }\n   return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun CharArray.min(): Char {\n   if (isEmpty()) throw NoSuchElementException()\n   var min = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (min > e) min = e\n   }\n   return min\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.minBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic\n inline fun <T, R : Comparable<R>> Array<out T>.minBy(selector: (T) -> R): T {\n   if (isEmpty()) throw NoSuchElementException()\n   var minElem = this[0]\n   val lastIndex = this.lastIndex\n   if (lastIndex == 0) return minElem\n   var minValue = selector(minElem)\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   val v = selector(e)\n   if (minValue > v) {\n   minElem = e\n   minValue = v\n   }\n   }\n   return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.minBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <R : Comparable<R>> ByteArray.minBy(selector: (Byte) -> R): Byte {\n   if (isEmpty())

throw
NoSuchElementException()\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0)
return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v =
selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return
minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws
NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLO
ADS\")\npublic inline fun <R : Comparable<R>> ShortArray.minBy(selector: (Short) -> R): Short {\n if
(isEmpty()) throw NoSuchElementException()\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex)
{\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue
= v\n }\n }\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given
function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLO
ADS\")\npublic inline fun <R : Comparable<R>> IntArray.minBy(selector: (Int) -> R): Int {\n if (isEmpty())
throw NoSuchElementException()\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex ==
0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val
v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return
minElem\n}\n\n/**\n * Returns
the first element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if
the array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.minBy\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLO
ADS\")\npublic inline fun <R : Comparable<R>> LongArray.minBy(selector: (Long) -> R): Long {\n if
(isEmpty()) throw NoSuchElementException()\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n
}\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n
* @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLO
ADS\")\npublic inline fun <R : Comparable<R>> FloatArray.minBy(selector: (Float) -> R): Float {\n if
(isEmpty()) throw NoSuchElementException()\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if
(lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e =
this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n
}\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n
* @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLO
ADS\")\npublic inline fun <R : Comparable<R>> DoubleArray.minBy(selector: (Double) ->
R): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minElem = this[0]\n val lastIndex =
this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in
1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n
minValue = v\n }\n }\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value
of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLO

ADS\")\npublic inline fun <R : Comparable<R>> BooleanArray.minBy(selector: (Boolean) -> R): Boolean {\n if (isEmpty()) throw NoSuchElementException()\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.minBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <R : Comparable<R>> CharArray.minBy(selector: (Char) -> R): Char {\n if (isEmpty()) throw NoSuchElementException()\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <T, R : Comparable<R>> Array<out T>.minByOrNull(selector: (T) -> R): T? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> ByteArray.minByOrNull(selector: (Byte) -> R): Byte? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> ShortArray.minByOrNull(selector: (Short) -> R): Short? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> IntArray.minByOrNull(selector: (Int) -> R): Int? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> LongArray.minByOrNull(selector: (Long) -> R): Long? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> FloatArray.minByOrNull(selector: (Float) -> R): Float? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n

for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> DoubleArray.minByOrNull(selector: (Double) -> R): Double? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null`
if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> BooleanArray.minByOrNull(selector: (Boolean) -> R): Boolean? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> CharArray.minByOrNull(selector: (Char) -> R): Char? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex
= this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOf(selector: (T) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue
= minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.minOf(selector: (Byte) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOf(selector: (Short) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic

inline fun IntArray.minOf(selector: (Int) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOf(selector: (Long) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOf(selector: (Float) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOf(selector: (Double) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOf(selector: (Boolean) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.minOf(selector: (Char) -> Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOf(selector: (T) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result

is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.minOf(selector: (Byte) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOf(selector: (Short) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.minOf(selector: (Int) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOf(selector: (Long) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOf(selector: (Float) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOf(selector: (Double) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOf(selector: (Boolean) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return

minValue\n}\n\n/**\n
* Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.minOf(selector: (Char) -> Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <T, R : Comparable<R>> Array<out T>.minOf(selector: (T) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ByteArray.minOf(selector: (Byte) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val
v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ShortArray.minOf(selector: (Short) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> IntArray.minOf(selector: (Int) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> LongArray.minOf(selector: (Long) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n
for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> FloatArray.minOf(selector: (Float) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var

minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (minValue > v) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n *
 @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> DoubleArray.minOf(selector: (Double) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (minValue > v) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> BooleanArray.minOf(selector: (Boolean) -> R): R {\n   if (isEmpty())
 throw NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (minValue > v) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> CharArray.minOf(selector: (Char) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (minValue > v) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector]
 function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOfOrNull(selector: (T) -> Double): Double? {\n   if (isEmpty()) return null\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   minValue = minOf(minValue, v)\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
 inline fun ByteArray.minOfOrNull(selector: (Byte) -> Double): Double? {\n   if (isEmpty()) return null\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   minValue = minOf(minValue, v)\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.minOfOrNull(selector: (Short) -> Double): Double? {\n   if (isEmpty()) return null\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   minValue
 = minOf(minValue, v)\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.minOfOrNull(selector: (Int) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOfOrNull(selector: (Long) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOfOrNull(selector: (Float) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOfOrNull(selector: (Double) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOfOrNull(selector: (Boolean) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.minOfOrNull(selector: (Char) -> Double): Double? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.minOfOrNull(selector: (T) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the

returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.minOfOrNull(selector: (Byte) ->
Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the
smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if
there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun ShortArray.minOfOrNull(selector: (Short) -> Float): Float? {\n if (isEmpty()) return null\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue =
minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.minOfOrNull(selector: (Int) ->
Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex)
{\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n *
Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array
or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned
result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.minOfOrNull(selector: (Long) -
> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the
smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if
there are no elements.\n * \n * If any
of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.minOfOrNull(selector: (Float) ->
Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n
val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the
smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if
there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is
`NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.minOfOrNull(selector:
(Double) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.minOfOrNull(selector:
(Boolean) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue\n}\n\n/**\n * Returns the smallest value among all values

produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.minOfOrNull(selector: (Char) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Array<out T>.minOfOrNull(selector: (T) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ByteArray.minOfOrNull(selector: (Byte) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ShortArray.minOfOrNull(selector: (Short) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> IntArray.minOfOrNull(selector: (Int) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> LongArray.minOfOrNull(selector: (Long) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the\n array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> FloatArray.minOfOrNull(selector: (Float) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>

DoubleArray.minOfOrNull(selector: (Double) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n
for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> BooleanArray.minOfOrNull(selector: (Boolean) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> CharArray.minOfOrNull(selector: (Char) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out T>.minOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n
var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.minOfWith(comparator: Comparator<in R>, selector: (Byte) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.minOfWith(comparator: Comparator<in R>, selector: (Short) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.minOfWith(comparator: Comparator<in R>, selector: (Int) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.minOfWith(comparator: Comparator<in R>, selector:
(Long) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.minOfWith(comparator: Comparator<in R>, selector: (Float) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.minOfWith(comparator: Comparator<in R>, selector: (Double) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.minOfWith(comparator: Comparator<in R>, selector: (Boolean) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <R> CharArray.minOfWith(comparator: Comparator<in R>, selector: (Char) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Array<out T>.minOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue,
v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Byte) -> R): R? {\n if (isEmpty()) return

null\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(minValue, v) > 0) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null`
 if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Short) -> R): R? {\n   if (isEmpty()) return null\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(minValue, v) > 0) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Int)
 -> R): R? {\n   if (isEmpty()) return null\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(minValue, v) > 0) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Long) -> R): R? {\n   if (isEmpty()) return null\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(minValue, v) > 0) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n
 * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Float) -> R): R? {\n   if (isEmpty()) return null\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(minValue, v) > 0) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
 inline fun <R> DoubleArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Double) -> R): R? {\n   if (isEmpty()) return null\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(minValue, v) > 0) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Boolean) -> R): R? {\n   if (isEmpty()) return null\n   var minValue = selector(this[0])\n
    for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(minValue, v) > 0) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n

*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic fun Array<out Double>.minOrNull(): Double? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic fun Array<out Float>.minOrNull(): Float? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T : Comparable<T>> Array<out T>.minOrNull(): T? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun ByteArray.minOrNull(): Byte? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun ShortArray.minOrNull(): Short? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun IntArray.minOrNull(): Int? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun LongArray.minOrNull(): Long? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic fun FloatArray.minOrNull(): Float? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic fun DoubleArray.minOrNull(): Double? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun CharArray.minOrNull(): Char? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun <T> Array<out T>.minWith(comparator: Comparator<in T>): T {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun ByteArray.minWith(comparator: Comparator<in Byte>): Byte {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the

smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun ShortArray.minWith(comparator: Comparator<in Short>): Short {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun IntArray.minWith(comparator: Comparator<in Int>): Int {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun LongArray.minWith(comparator: Comparator<in Long>): Long {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min,
e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun FloatArray.minWith(comparator: Comparator<in Float>): Float {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun DoubleArray.minWith(comparator: Comparator<in Double>): Double {\n if (isEmpty()) throw NoSuchElementException()\n
var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun BooleanArray.minWith(comparator: Comparator<in Boolean>): Boolean {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic
fun CharArray.minWith(comparator: Comparator<in Char>): Char {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\npublic fun <T> Array<out T>.minWithOrNull(comparator: Comparator<in T>): T? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n

*/\n@SinceKotlin(\"1.4\")\npublic fun ByteArray.minWithOrNull(comparator: Comparator<in Byte>): Byte? {\n if (isEmpty()) return

null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun ShortArray.minWithOrNull(comparator: Comparator<in Short>): Short? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun IntArray.minWithOrNull(comparator: Comparator<in Int>): Int? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min

= e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun LongArray.minWithOrNull(comparator: Comparator<in Long>): Long? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun FloatArray.minWithOrNull(comparator: Comparator<in Float>): Float? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator]

or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun DoubleArray.minWithOrNull(comparator: Comparator<in Double>): Double? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun BooleanArray.minWithOrNull(comparator: Comparator<in Boolean>): Boolean? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun CharArray.minWithOrNull(comparator: Comparator<in

Char>): Char? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\npublic fun <T> Array<out T>.none(): Boolean {\n return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\npublic fun ByteArray.none(): Boolean {\n return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\npublic fun ShortArray.none(): Boolean {\n return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\npublic fun IntArray.none(): Boolean {\n return isEmpty()\n}\n\n/**\n *

Returns `true` if the array has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\npublic fun LongArray.none(): Boolean {\n return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\npublic fun FloatArray.none(): Boolean {\n return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\npublic fun DoubleArray.none(): Boolean {\n return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\npublic fun BooleanArray.none(): Boolean {\n return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample

samples.collections.Collections.Aggregates.none\n */\npublic fun CharArray.none(): Boolean {\n   return isEmpty()\n}\n\n/**\n * Returns `true` if no elements match the
 given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun <T> Array<out T>.none(predicate: (T) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun ByteArray.none(predicate: (Byte) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun ShortArray.none(predicate: (Short) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n
 */\npublic inline fun IntArray.none(predicate: (Int) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun LongArray.none(predicate: (Long) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun FloatArray.none(predicate: (Float) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun
 DoubleArray.none(predicate: (Double) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun BooleanArray.none(predicate: (Boolean) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun CharArray.none(predicate: (Char) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return false\n   return true\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.onEach(action: (T) -> Unit): Array<out
 T> {\n   return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.onEach(action: (Byte) -> Unit): ByteArray {\n   return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.onEach(action: (Short) -> Unit): ShortArray {\n   return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.onEach(action: (Int) -> Unit): IntArray {\n   return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns
 the array itself afterwards.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.onEach(action: (Long) -> Unit): LongArray {\n   return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.onEach(action: (Float) -> Unit): FloatArray {\n   return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.onEach(action: (Double) ->

Unit): DoubleArray {\n   return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.onEach(action: (Boolean) -> Unit): BooleanArray {\n   return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.onEach(action: (Char) -> Unit): CharArray {\n   return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.onEachIndexed(action: (index: Int, T) -> Unit): Array<out T> {\n   return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the
 array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.onEachIndexed(action: (index: Int, Byte) -> Unit): ByteArray {\n   return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.onEachIndexed(action: (index: Int, Short) -> Unit): ShortArray {\n   return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function
 that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.onEachIndexed(action: (index: Int, Int) -> Unit): IntArray {\n   return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.onEachIndexed(action: (index: Int, Long) -> Unit): LongArray {\n   return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element
 itself\n * and performs the action on the element.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.onEachIndexed(action: (index: Int, Float) -> Unit): FloatArray {\n   return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.onEachIndexed(action: (index: Int, Double) -> Unit): DoubleArray {\n   return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the
 element.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.onEachIndexed(action: (index: Int, Boolean) -> Unit): BooleanArray {\n   return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n

*/\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.onEachIndexed(action: (index: Int, Char) -> Unit): CharArray {\n return apply { forEachIndexed(action) }\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null`
when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun <S, T : S> Array<out T>.reduce(operation: (acc: S, T) -> S): S {\n if (isEmpty())\n throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator: S = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value
and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun ByteArray.reduce(operation: (acc: Byte, Byte) -> Byte): Byte {\n if (isEmpty())\n throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n
*/\npublic inline fun ShortArray.reduce(operation: (acc: Short, Short) -> Short): Short {\n if (isEmpty())\n throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun IntArray.reduce(operation: (acc: Int, Int) -> Int): Int {\n
if (isEmpty())\n throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun LongArray.reduce(operation: (acc: Long, Long) -> Long): Long {\n if (isEmpty())\n throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator
= this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates

the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun FloatArray.reduce(operation: (acc: Float, Float) -> Float): Float {\n   if (isEmpty())\n   throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n   var accumulator = this[0]\n   for (index in 1..lastIndex) {\n   accumulator = operation(accumulator, this[index])\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun DoubleArray.reduce(operation: (acc: Double, Double) -> Double): Double {\n   if (isEmpty())\n   throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n   var accumulator = this[0]\n   for (index in 1..lastIndex) {\n   accumulator = operation(accumulator, this[index])\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element
 and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun BooleanArray.reduce(operation: (acc: Boolean, Boolean) -> Boolean): Boolean {\n   if (isEmpty())\n   throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n   var accumulator = this[0]\n   for (index in 1..lastIndex) {\n   accumulator = operation(accumulator, this[index])\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun CharArray.reduce(operation: (acc: Char, Char) -> Char): Char {\n   if (isEmpty())\n   throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n   var accumulator = this[0]\n   for (index in 1..lastIndex) {\n   accumulator = operation(accumulator, this[index])\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can
 be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun <S, T : S> Array<out T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {\n   if (isEmpty())\n   throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n   var accumulator: S = this[0]\n   for (index in 1..lastIndex) {\n   accumulator = operation(index, accumulator, this[index])\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be
 empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun ByteArray.reduceIndexed(operation: (index: Int, acc: Byte, Byte) -> Byte): Byte {\n   if (isEmpty())\n   throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n   var accumulator = this[0]\n   for (index in 1..lastIndex) {\n   accumulator = operation(index, accumulator, this[index])\n   }\n   return

accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty
in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun ShortArray.reduceIndexed(operation: (index: Int, acc: Short, Short) -> Short): Short {\n if (isEmpty())\n throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in
an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun IntArray.reduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): Int {\n if (isEmpty())\n throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected
way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun LongArray.reduceIndexed(operation: (index: Int, acc: Long, Long) -> Long): Long {\n if (isEmpty())\n throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n
* please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun FloatArray.reduceIndexed(operation: (index: Int, acc: Float, Float) -> Float): Float {\n if (isEmpty())\n throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n *
please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun DoubleArray.reduceIndexed(operation: (index: Int, acc: Double, Double) -> Double): Double {\n if (isEmpty())\n throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an

exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun BooleanArray.reduceIndexed(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): Boolean {\n if (isEmpty())\n throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun CharArray.reduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): Char {\n if (isEmpty())\n throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <S, T : S> Array<out T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {\n if (isEmpty())\n return null\n var accumulator: S = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun ByteArray.reduceIndexedOrNull(operation: (index: Int, acc: Byte, Byte) -> Byte): Byte? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun ShortArray.reduceIndexedOrNull(operation: (index: Int, acc: Short, Short) -> Short): Short? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun IntArray.reduceIndexedOrNull(operation: (index: Int, acc: Int, Int) -> Int): Int? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return

accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun LongArray.reduceIndexedOrNull(operation: (index: Int, acc: Long, Long) -> Long): Long? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun FloatArray.reduceIndexedOrNull(operation: (index: Int, acc: Float, Float) -> Float): Float? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun DoubleArray.reduceIndexedOrNull(operation: (index: Int, acc: Double, Double) -> Double): Double? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\npublic
inline fun BooleanArray.reduceIndexedOrNull(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): Boolean? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun CharArray.reduceIndexedOrNull(operation: (index: Int, acc: Char, Char) -> Char): Char? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S> Array<out T>.reduceOrNull(operation: (acc: S, T) -> S): S? {\n if (isEmpty())\n return null\n var accumulator: S = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with

the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun ByteArray.reduceOrNull(operation: (acc: Byte, Byte) -> Byte): Byte? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun ShortArray.reduceOrNull(operation: (acc: Short, Short) -> Short): Short? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun IntArray.reduceOrNull(operation: (acc: Int, Int) -> Int): Int? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun LongArray.reduceOrNull(operation: (acc: Long, Long) -> Long): Long? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun FloatArray.reduceOrNull(operation: (acc: Float, Float) -> Float): Float? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun DoubleArray.reduceOrNull(operation: (acc: Double, Double) -> Double): Double? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n *

and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
BooleanArray.reduceOrNull(operation: (acc: Boolean, Boolean) -> Boolean): Boolean? {\n if (isEmpty())\n
return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator =
operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the
first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and
an element,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceOrNull\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun
CharArray.reduceOrNull(operation: (acc: Char, Char) ->
Char): Char? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n
accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value
starting with the last element and applying [operation] from right to left\n * to each element and current accumulator
value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please
use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function
that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun <S, T : S> Array<out
T>.reduceRight(operation: (T, acc: S) -> S): S {\n var index = lastIndex\n if (index < 0) throw
UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator:
S = get(index--)\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the
array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its
receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n *
and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun ByteArray.reduceRight(operation:
(Byte, acc: Byte) -> Byte): Byte {\n var index = lastIndex\n if (index < 0) throw
UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while
(index >= 0) {\n accumulator = operation(get(index--),
accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and
applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Throws an
exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull]
instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and
current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun ShortArray.reduceRight(operation:
(Short, acc: Short) -> Short): Short {\n var index = lastIndex\n if (index < 0) throw
UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while
(index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n *
Accumulates
value starting with the last element and applying [operation] from right to left\n * to each element and current
accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected
way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param
[operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun
IntArray.reduceRight(operation: (Int, acc: Int) -> Int): Int {\n var index = lastIndex\n if (index < 0) throw
UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while

(index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun LongArray.reduceRight(operation: (Long, acc: Long) -> Long): Long {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun FloatArray.reduceRight(operation: (Float, acc: Float) -> Float): Float {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun DoubleArray.reduceRight(operation: (Double, acc: Double) -> Double): Double {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver
is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun BooleanArray.reduceRight(operation: (Boolean, acc: Boolean) -> Boolean): Boolean {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element
and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun CharArray.reduceRight(operation: (Char, acc: Char) -> Char): Char {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null`

when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself
and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun <S, T : S> Array<out T>.reduceRightIndexed(operation: (index: Int, T, acc: S) -> S): S {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator: S = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the
index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun ByteArray.reduceRightIndexed(operation: (index: Int, Byte, acc: Byte) -> Byte): Byte {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param
[operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun ShortArray.reduceRightIndexed(operation: (index: Int, Short, acc: Short) -> Short): Short {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null`
when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun IntArray.reduceRightIndexed(operation: (index: Int, Int, acc: Int) -> Int): Int {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull]
instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun LongArray.reduceRightIndexed(operation: (index: Int, Long, acc: Long) -> Long): Long {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and

current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n
* please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun FloatArray.reduceRightIndexed(operation: (index: Int, Float, acc: Float) -> Float): Float {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If
the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun DoubleArray.reduceRightIndexed(operation: (index: Int, Double, acc: Double) -> Double): Double {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n *
Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun BooleanArray.reduceRightIndexed(operation: (index: Int, Boolean, acc: Boolean) -> Boolean): Boolean {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original
array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun CharArray.reduceRightIndexed(operation: (index: Int, Char, acc: Char) -> Char): Char {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to
each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <S, T : S> Array<out T>.reduceRightIndexedOrNull(operation: (index: Int, T, acc: S) -> S): S? {\n var index = lastIndex\n if (index < 0) return null\n var accumulator: S = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with

its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun ByteArray.reduceRightIndexedOrNull(operation: (index: Int, Byte, acc: Byte) -> Byte): Byte? {\n var index = lastIndex\n if (index < 0) return null\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates
the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun ShortArray.reduceRightIndexedOrNull(operation: (index: Int, Short, acc: Short) -> Short): Short? {\n var index = lastIndex\n if (index < 0) return null\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\npublic
inline fun IntArray.reduceRightIndexedOrNull(operation: (index: Int, Int, acc: Int) -> Int): Int? {\n var index = lastIndex\n if (index < 0) return null\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun LongArray.reduceRightIndexedOrNull(operation: (index: Int, Long, acc: Long) -> Long): Long? {\n var index = lastIndex\n
if (index < 0) return null\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun FloatArray.reduceRightIndexedOrNull(operation: (index: Int, Float, acc: Float) -> Float): Float? {\n var index = lastIndex\n if (index < 0) return null\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index,
get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun DoubleArray.reduceRightIndexedOrNull(operation: (index: Int, Double, acc: Double) -> Double): Double? {\n var index = lastIndex\n if (index < 0) return null\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index),

accumulator)\n   --index\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun BooleanArray.reduceRightIndexedOrNull(operation: (index: Int, Boolean, acc: Boolean) -> Boolean): Boolean? {\n   var index = lastIndex\n   if (index < 0) return null\n   var accumulator = get(index--)\n   while (index >= 0) {\n   accumulator = operation(index, get(index), accumulator)\n   --index\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun CharArray.reduceRightIndexedOrNull(operation: (index: Int, Char, acc: Char) -> Char): Char? {\n   var index = lastIndex\n   if (index < 0) return null\n   var accumulator = get(index--)\n   while (index >= 0) {\n   accumulator = operation(index, get(index), accumulator)\n   --index\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S> Array<out T>.reduceRightOrNull(operation: (T, acc: S) -> S): S? {\n   var index = lastIndex\n   if (index < 0) return null\n   var accumulator: S = get(index--)\n   while (index >= 0) {\n   accumulator = operation(get(index--), accumulator)\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun ByteArray.reduceRightOrNull(operation: (Byte, acc: Byte) -> Byte): Byte? {\n   var index = lastIndex\n   if (index < 0) return null\n   var accumulator = get(index--)\n   while (index >= 0) {\n   accumulator = operation(get(index--), accumulator)\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun ShortArray.reduceRightOrNull(operation: (Short, acc: Short) -> Short): Short? {\n   var index = lastIndex\n   if (index < 0) return null\n   var accumulator = get(index--)\n   while (index >= 0) {\n   accumulator = operation(get(index--), accumulator)\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun IntArray.reduceRightOrNull(operation: (Int, acc: Int) -> Int): Int? {\n   var index = lastIndex\n   if (index < 0) return null\n   var accumulator = get(index--)\n   while (index >= 0) {\n   accumulator = operation(get(index--),

accumulator)\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation]
 from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun LongArray.reduceRightOrNull(operation: (Long, acc: Long) -> Long): Long? {\n   var index = lastIndex\n   if (index < 0) return null\n   var accumulator = get(index--)\n   while (index >= 0) {\n   accumulator = operation(get(index--), accumulator)\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator
 value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun FloatArray.reduceRightOrNull(operation: (Float, acc: Float) -> Float): Float? {\n   var index = lastIndex\n   if (index < 0) return null\n   var accumulator = get(index--)\n   while (index >= 0) {\n   accumulator = operation(get(index--), accumulator)\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
 inline fun DoubleArray.reduceRightOrNull(operation: (Double, acc: Double) -> Double): Double? {\n   var index = lastIndex\n   if (index < 0) return null\n   var accumulator = get(index--)\n   while (index >= 0) {\n   accumulator = operation(get(index--), accumulator)\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun BooleanArray.reduceRightOrNull(operation: (Boolean, acc: Boolean) -> Boolean): Boolean? {\n   var index = lastIndex\n   if (index < 0) return null\n
    var accumulator = get(index--)\n   while (index >= 0) {\n   accumulator = operation(get(index--), accumulator)\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun CharArray.reduceRightOrNull(operation: (Char, acc: Char) -> Char): Char? {\n   var index = lastIndex\n   if (index < 0) return null\n   var accumulator = get(index--)\n   while (index >= 0) {\n   accumulator = operation(get(index--), accumulator)\n   }\n   return accumulator\n}\n\n/**\n * Returns a list containing successive
 accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\npublic inline fun

<T, R> Array<out T>.runningFold(initial: R, operation: (acc: R, T) -> R): List<R> {\n   if (isEmpty()) return listOf(initial)\n   val result = ArrayList<R>(size + 1).apply { add(initial) }\n   var accumulator = initial\n   for (element in this) {\n   accumulator = operation(accumulator, element)\n   result.add(accumulator)\n   }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation
 values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.runningFold(initial: R, operation: (acc: R, Byte) -> R): List<R> {\n   if (isEmpty()) return listOf(initial)\n   val result = ArrayList<R>(size + 1).apply { add(initial) }\n   var accumulator = initial\n   for (element in this) {\n   accumulator = operation(accumulator, element)\n   result.add(accumulator)\n   }\n   return result\n}\n\n/**\n * Returns a list containing successive accumulation
 values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.runningFold(initial: R, operation: (acc: R, Short) -> R): List<R> {\n   if (isEmpty()) return listOf(initial)\n   val result = ArrayList<R>(size + 1).apply { add(initial) }\n   var accumulator = initial\n   for (element in this) {\n   accumulator = operation(accumulator, element)\n   result.add(accumulator)\n   }\n   return result\n}\n\n/**\n * Returns a list containing successive accumulation
 values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.runningFold(initial: R, operation: (acc: R, Int) -> R): List<R> {\n   if (isEmpty()) return listOf(initial)\n   val result = ArrayList<R>(size + 1).apply { add(initial) }\n   var accumulator = initial\n   for (element in this) {\n   accumulator = operation(accumulator, element)\n   result.add(accumulator)\n   }\n   return result\n}\n\n/**\n * Returns a list containing successive
 accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.runningFold(initial: R, operation: (acc: R, Long) -> R): List<R> {\n   if (isEmpty()) return listOf(initial)\n   val result = ArrayList<R>(size + 1).apply { add(initial) }\n   var accumulator = initial\n   for (element in this) {\n   accumulator = operation(accumulator, element)\n   result.add(accumulator)\n   }\n   return result\n}\n\n/**\n * Returns a list containing successive
 accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n *

@sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.runningFold(initial: R, operation: (acc: R, Float) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive
accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.runningFold(initial: R, operation: (acc: R, Double) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing
successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.runningFold(initial: R, operation: (acc: R, Boolean) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a
list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.runningFold(initial: R, operation: (acc: R, Char) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns
a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <T, R> Array<out T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n
result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]

function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Long) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n

for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and the element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S> Array<out T>.runningReduce(operation: (acc: S, T) -> S): List<S> {\n if (isEmpty()) return emptyList()\n var accumulator: S = this[0]\n val result = ArrayList<S>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample

samples.collections.Collections.Aggregates.runningReduce\n
*/\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.runningReduce(operation: (acc: Byte, Byte) -> Byte): List<Byte> {\n   if (isEmpty()) return emptyList()\n   var accumulator = this[0]\n   val result = ArrayList<Byte>(size).apply { add(accumulator) }\n   for (index in 1 until size) {\n   accumulator = operation(accumulator, this[index])\n   result.add(accumulator)\n   }\n   return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator
value that starts with the first element of this array.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*/\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.runningReduce(operation: (acc: Short, Short) -> Short): List<Short> {\n   if (isEmpty()) return emptyList()\n   var accumulator = this[0]\n   val result = ArrayList<Short>(size).apply { add(accumulator) }\n   for (index in 1 until size) {\n   accumulator = operation(accumulator, this[index])\n   result.add(accumulator)\n   }\n   return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that takes current accumulator
value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*/\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.runningReduce(operation: (acc: Int, Int) -> Int): List<Int> {\n   if (isEmpty()) return emptyList()\n   var accumulator = this[0]\n   val result = ArrayList<Int>(size).apply { add(accumulator) }\n   for (index in 1 until size) {\n   accumulator = operation(accumulator, this[index])\n   result.add(accumulator)\n   }\n   return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*/\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.runningReduce(operation: (acc: Long, Long) -> Long): List<Long> {\n   if (isEmpty()) return emptyList()\n   var accumulator = this[0]\n   val result = ArrayList<Long>(size).apply { add(accumulator) }\n   for (index in 1 until size) {\n   accumulator = operation(accumulator, this[index])\n   result.add(accumulator)\n   }\n   return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*/\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.runningReduce(operation: (acc:
Float, Float) -> Float): List<Float> {\n   if (isEmpty()) return emptyList()\n   var accumulator = this[0]\n   val result = ArrayList<Float>(size).apply { add(accumulator) }\n   for (index in 1 until size) {\n   accumulator = operation(accumulator, this[index])\n   result.add(accumulator)\n   }\n   return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n
*/\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.runningReduce(operation: (acc: Double, Double) -> Double): List<Double> {\n   if (isEmpty()) return emptyList()\n   var accumulator = this[0]\n   val

result = ArrayList<Double>(size).apply { add(accumulator) }\n   for (index in 1 until size) {\n   accumulator = operation(accumulator, this[index])\n   result.add(accumulator)\n   }\n   return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.runningReduce(operation: (acc: Boolean, Boolean) -> Boolean): List<Boolean> {\n   if (isEmpty()) return emptyList()\n   var accumulator = this[0]\n   val result = ArrayList<Boolean>(size).apply { add(accumulator) }\n   for (index in 1 until size) {\n accumulator
= operation(accumulator, this[index])\n   result.add(accumulator)\n   }\n   return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.runningReduce(operation: (acc: Char, Char) -> Char): List<Char> {\n   if (isEmpty()) return emptyList()\n   var accumulator = this[0]\n   val result = ArrayList<Char>(size).apply { add(accumulator) }\n   for (index in 1 until size) {\n   accumulator = operation(accumulator, this[index])\n   result.add(accumulator)\n   }\n   return result\n}\n\n/**\n * Returns
a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <S, T : S> Array<out T>.runningReduceIndexed(operation: (index: Int, acc: S, T) -> S): List<S> {\n   if (isEmpty()) return emptyList()\n   var accumulator: S = this[0]\n   val result = ArrayList<S>(size).apply { add(accumulator) }\n   for (index in 1 until size) {\n   accumulator = operation(index, accumulator,
this[index])\n   result.add(accumulator)\n   }\n   return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.runningReduceIndexed(operation: (index: Int, acc: Byte, Byte) -> Byte): List<Byte> {\n   if (isEmpty()) return emptyList()\n   var accumulator = this[0]\n   val result = ArrayList<Byte>(size).apply { add(accumulator) }\n   for (index in 1 until size) {\n   accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n
}\n   return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.runningReduceIndexed(operation: (index: Int, acc: Short, Short) -> Short): List<Short> {\n   if (isEmpty()) return emptyList()\n   var accumulator = this[0]\n   val result = ArrayList<Short>(size).apply { add(accumulator) }\n   for (index in 1 until size) {\n   accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n   }\n

```
    return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.runningReduceIndexed(operation: (index: Int, acc: Int, Int) -> Int): List<Int> {\n    if (isEmpty()) return emptyList()\n    var accumulator = this[0]\n    val result = ArrayList<Int>(size).apply { add(accumulator) }\n    for (index in 1 until size) {\n        accumulator = operation(index, accumulator, this[index])\n        result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.runningReduceIndexed(operation: (index: Int, acc: Long, Long) -> Long): List<Long> {\n    if (isEmpty()) return emptyList()\n    var accumulator = this[0]\n    val result = ArrayList<Long>(size).apply { add(accumulator) }\n    for (index in 1 until size) {\n        accumulator = operation(index, accumulator, this[index])\n        result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.runningReduceIndexed(operation: (index: Int, acc: Float, Float) -> Float): List<Float> {\n    if (isEmpty()) return emptyList()\n    var accumulator = this[0]\n    val result = ArrayList<Float>(size).apply { add(accumulator) }\n    for (index in 1 until size) {\n        accumulator = operation(index, accumulator, this[index])\n        result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.runningReduceIndexed(operation: (index: Int, acc: Double, Double) -> Double): List<Double> {\n    if (isEmpty()) return emptyList()\n    var accumulator = this[0]\n    val result = ArrayList<Double>(size).apply { add(accumulator) }\n    for (index in 1 until size) {\n        accumulator = operation(index, accumulator, this[index])\n        result.add(accumulator)\n    }\n    return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.runningReduceIndexed(operation: (index: Int, acc: Boolean, Boolean) -> Boolean): List<Boolean> {\n    if (isEmpty()) return emptyList()\n    var accumulator = this[0]\n    val result =
```

ArrayList<Boolean>(size).apply { add(accumulator) }\n   for (index in 1 until size) {\n   accumulator = operation(index, accumulator, this[index])\n   result.add(accumulator)\n   }\n   return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.runningReduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): List<Char> {\n   if (isEmpty()) return emptyList()\n   var accumulator = this[0]\n   val result = ArrayList<Char>(size).apply { add(accumulator) }\n   for (index in 1 until size) {\n   accumulator = operation(index, accumulator, this[index])\n   result.add(accumulator)\n   }\n   return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <T, R> Array<out T>.scan(initial: R, operation: (acc: R, T) -> R): List<R> {\n   return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.scan(initial: R, operation: (acc: R, Byte) -> R): List<R> {\n   return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.scan(initial: R, operation: (acc: R, Short) -> R): List<R> {\n   return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.scan(initial: R, operation: (acc: R, Int) -> R): List<R> {\n   return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n

*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.scan(initial: R, operation: (acc: R, Long) -> R): List<R> {\n   return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.scan(initial: R, operation: (acc: R, Float) -> R): List<R> {\n   return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.scan(initial: R, operation: (acc: R, Double) -> R): List<R> {\n   return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> BooleanArray.scan(initial: R, operation: (acc: R, Boolean) -> R): List<R> {\n   return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.scan(initial: R, operation: (acc: R, Char) -> R): List<R> {\n   return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <T, R> Array<out T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {\n   return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the

next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> ByteArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Byte) -> R): List<R> {\n return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> ShortArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Short) -> R): List<R> {\n return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> IntArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Int) -> R): List<R> {\n return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> LongArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Long) -> R): List<R> {\n return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> FloatArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Float) -> R): List<R> {\n return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> DoubleArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Double) -> R): List<R> {\n return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current

accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun <R> BooleanArray.scanIndexed(initial: R, operation: (index: Int, acc: R, Boolean) -> R): List<R> {\n return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharArray.scanIndexed(initial:
R, operation: (index: Int, acc: R, Char) -> R): List<R> {\n return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\",
ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun <T> Array<out T>.sumBy(selector: (T) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\",
ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun ByteArray.sumBy(selector: (Byte) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\",
ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun ShortArray.sumBy(selector: (Short) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\",
ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun IntArray.sumBy(selector: (Int) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\",
ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince
= \"1.5\")\npublic inline fun LongArray.sumBy(selector: (Long) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\",
ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun FloatArray.sumBy(selector: (Float) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\",
ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun DoubleArray.sumBy(selector: (Double) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\",
ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun BooleanArray.sumBy(selector: (Boolean) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum +=

selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun CharArray.sumBy(selector: (Char) -> Int): Int {\n   var sum: Int = 0\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun <T> Array<out T>.sumByDouble(selector: (T) -> Double): Double {\n   var sum: Double = 0.0\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun ByteArray.sumByDouble(selector: (Byte) -> Double): Double {\n   var sum: Double = 0.0\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun ShortArray.sumByDouble(selector: (Short) -> Double): Double {\n   var sum: Double = 0.0\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun IntArray.sumByDouble(selector: (Int) -> Double): Double {\n   var sum: Double = 0.0\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun LongArray.sumByDouble(selector: (Long) -> Double): Double {\n   var sum: Double = 0.0\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun FloatArray.sumByDouble(selector: (Float) -> Double): Double {\n   var sum: Double = 0.0\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun DoubleArray.sumByDouble(selector: (Double) -> Double): Double {\n   var sum: Double = 0.0\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun BooleanArray.sumByDouble(selector: (Boolean) -> Double): Double {\n   var sum: Double = 0.0\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun CharArray.sumByDouble(selector: (Char) -> Double): Double {\n   var sum: Double = 0.0\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfDouble\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.sumOf(selector: (T) -> Double): Double {\n   var sum: Double = 0.toDouble()\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n

*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfDouble\")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.sumOf(selector: (Byte) -> Double): Double {\n   var sum: Double = 0.toDouble()\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfDouble\")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.sumOf(selector: (Short) -> Double): Double {\n   var sum: Double = 0.toDouble()\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfDouble\")\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.sumOf(selector: (Int) -> Double): Double {\n   var sum: Double = 0.toDouble()\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfDouble\")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.sumOf(selector: (Long) -> Double): Double {\n   var sum: Double = 0.toDouble()\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfDouble\")\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.sumOf(selector: (Float) -> Double): Double {\n   var sum: Double = 0.toDouble()\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfDouble\")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.sumOf(selector: (Double) -> Double): Double {\n   var sum: Double = 0.toDouble()\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfDouble\")\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.sumOf(selector: (Boolean) -> Double): Double {\n   var sum: Double = 0.toDouble()\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfDouble\")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.sumOf(selector: (Char) -> Double): Double {\n   var sum: Double = 0.toDouble()\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfInt\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.sumOf(selector: (T) -> Int): Int {\n   var sum: Int = 0.toInt()\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfInt\")\n@kotlin.internal.InlineOnly\npublic

inline fun ByteArray.sumOf(selector: (Byte) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfInt\")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.sumOf(selector: (Short) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfInt\")\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.sumOf(selector: (Int) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfInt\")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.sumOf(selector: (Long) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfInt\")\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.sumOf(selector: (Float) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfInt\")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.sumOf(selector: (Double) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfInt\")\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.sumOf(selector: (Boolean) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfInt\")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.sumOf(selector: (Char) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfLong\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.sumOf(selector: (T) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfLong\")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.sumOf(selector: (Byte) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n

*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfLong\")\n@kotlin.internal.InlineOnly\npublic
inline fun ShortArray.sumOf(selector: (Short) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfLong\")\n@kotlin.internal.InlineOnly\npublic inline fun
IntArray.sumOf(selector: (Int) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n
sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfLong\")\n@kotlin.internal.InlineOnly\npublic
inline fun LongArray.sumOf(selector: (Long) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfLong\")\n@kotlin.internal.InlineOnly\npublic inline fun
FloatArray.sumOf(selector: (Float) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n
sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfLong\")\n@kotlin.internal.InlineOnly\npublic
inline fun DoubleArray.sumOf(selector: (Double) -> Long): Long {\n var sum: Long = 0.toLong()\n for
(element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfLong\")\n@kotlin.internal.InlineOnly\npublic inline fun
BooleanArray.sumOf(selector: (Boolean) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this)
{\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfLong\")\n@kotlin.internal.InlineOnly\npublic
inline fun CharArray.sumOf(selector: (Char) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.sumOf(selector: (T) -> UInt): UInt {\n
var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the
array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedType
s::class)\n@kotlin.internal.InlineOnly\npublic
inline fun ByteArray.sumOf(selector: (Byte) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.sumOf(selector: (Short) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.sumOf(selector: (Int) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.sumOf(selector: (Long) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n
}\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.sumOf(selector: (Float) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.sumOf(selector: (Double) -> UInt): UInt
{\n var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.sumOf(selector: (Boolean) -> UInt): UInt
{\n var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return
sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic
inline fun CharArray.sumOf(selector: (Char) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Array<out T>.sumOf(selector: (T) -> ULong):
ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n
return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in
the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic

inline fun ByteArray.sumOf(selector: (Byte) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.sumOf(selector: (Short) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.sumOf(selector: (Int) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.sumOf(selector: (Long) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n *\n Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.sumOf(selector: (Float) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.sumOf(selector: (Double) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun BooleanArray.sumOf(selector: (Boolean) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic\n inline fun CharArray.sumOf(selector: (Char) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns an original collection containing all the non-`null` elements, throwing an [IllegalArgumentException] if there are any `null` elements.\n */\npublic fun <T : Any> Array<T?>.requireNoNulls(): Array<T> {\n for (element in this) {\n if (element == null) {\n throw IllegalArgumentException(\"null element found in $this.\")\n }\n }\n @Suppress(\"UNCHECKED_CAST\")\n return this as Array<T>\n}\n\n/**\n * Splits the original array into pair

of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n * @sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun <T> Array<out T>.partition(predicate:
(T) -> Boolean): Pair<List<T>, List<T>> {\n val first = ArrayList<T>()\n val second = ArrayList<T>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n * @sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun ByteArray.partition(predicate: (Byte) -> Boolean): Pair<List<Byte>, List<Byte>> {\n val first = ArrayList<Byte>()\n val second = ArrayList<Byte>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n
* Splits the original array into pair of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n * @sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun ShortArray.partition(predicate: (Short) -> Boolean): Pair<List<Short>, List<Short>> {\n val first = ArrayList<Short>()\n val second = ArrayList<Short>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n * @sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n
*/\npublic inline fun IntArray.partition(predicate: (Int) -> Boolean): Pair<List<Int>, List<Int>> {\n val first = ArrayList<Int>()\n val second = ArrayList<Int>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n * @sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun LongArray.partition(predicate: (Long) -> Boolean): Pair<List<Long>, List<Long>> {\n val first = ArrayList<Long>()\n val second = ArrayList<Long>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n
}\n }\n return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n * @sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun FloatArray.partition(predicate: (Float) -> Boolean): Pair<List<Float>, List<Float>> {\n val first = ArrayList<Float>()\n val second = ArrayList<Float>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n * @sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n
*/\npublic inline fun DoubleArray.partition(predicate: (Double) -> Boolean): Pair<List<Double>, List<Double>> {\n val first = ArrayList<Double>()\n val second = ArrayList<Double>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n * @sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n

*/\npublic inline fun BooleanArray.partition(predicate: (Boolean) -> Boolean): Pair<List<Boolean>, List<Boolean>> {\n   val first = ArrayList<Boolean>()\n   val second = ArrayList<Boolean>()\n   for (element in this) {\n   if (predicate(element)) {\n   first.add(element)\n   } else {\n   second.add(element)\n   }\n   }\n   return Pair(first, second)\n}\n\n/**\n * Splits the original array into pair of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n * @sample samples.collections.Arrays.Transformations.partitionArrayOfPrimitives\n */\npublic inline fun CharArray.partition(predicate: (Char) -> Boolean): Pair<List<Char>, List<Char>> {\n   val first = ArrayList<Char>()\n   val second = ArrayList<Char>()\n   for (element in this) {\n   if (predicate(element)) {\n   first.add(element)\n   } else {\n   second.add(element)\n   }\n   }\n   return Pair(first, second)\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <T, R> Array<out T>.zip(other: Array<out R>): List<Pair<T, R>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> ByteArray.zip(other: Array<out R>): List<Pair<Byte, R>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> ShortArray.zip(other: Array<out R>): List<Pair<Short, R>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> IntArray.zip(other: Array<out R>): List<Pair<Int, R>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> LongArray.zip(other: Array<out R>): List<Pair<Long, R>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> FloatArray.zip(other: Array<out R>): List<Pair<Float, R>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> DoubleArray.zip(other: Array<out R>): List<Pair<Double, R>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> BooleanArray.zip(other: Array<out R>): List<Pair<Boolean, R>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> CharArray.zip(other: Array<out R>): List<Pair<Char, R>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <T, R, V> Array<out T>.zip(other: Array<out R>, transform: (a: T, b: R) -> V): List<V> {\n   val size = minOf(size, other.size)\n   val list = ArrayList<V>(size)\n   for (i in 0 until size) {\n   list.add(transform(this[i], other[i]))\n   }\n   return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other]

array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V> ByteArray.zip(other: Array<out R>, transform: (a: Byte, b: R) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V> ShortArray.zip(other: Array<out R>, transform: (a: Short, b: R) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V> IntArray.zip(other: Array<out R>, transform: (a: Int, b: R) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n *
The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V> LongArray.zip(other: Array<out R>, transform: (a: Long, b: R) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V> FloatArray.zip(other: Array<out R>, transform: (a: Float, b: R) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V> DoubleArray.zip(other: Array<out R>, transform: (a: Double, b: R) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V> BooleanArray.zip(other: Array<out R>, transform: (a: Boolean, b: R) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V> CharArray.zip(other: Array<out R>, transform: (a: Char, b: R) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of pairs

```
 built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <T, R> Array<out T>.zip(other: Iterable<R>): List<Pair<T, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> ByteArray.zip(other: Iterable<R>): List<Pair<Byte, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic
 infix fun <R> ShortArray.zip(other: Iterable<R>): List<Pair<Short, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> IntArray.zip(other: Iterable<R>): List<Pair<Int, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> LongArray.zip(other: Iterable<R>): List<Pair<Long, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n
 * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> FloatArray.zip(other: Iterable<R>): List<Pair<Float, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> DoubleArray.zip(other: Iterable<R>): List<Pair<Double, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> BooleanArray.zip(other: Iterable<R>): List<Pair<Boolean, R>> {\n
    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <R> CharArray.zip(other: Iterable<R>): List<Pair<Char, R>> {\n    return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <T, R, V> Array<out T>.zip(other: Iterable<R>, transform: (a: T, b: R) -> V): List<V> {\n    val arraySize = size\n    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n
 var i = 0\n    for (element in other) {\n        if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n
 }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V> ByteArray.zip(other: Iterable<R>, transform: (a: Byte, b: R) -> V): List<V> {\n    val arraySize = size\n    val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n    var i = 0\n    for (element in other) {\n        if (i >= arraySize) break\n        list.add(transform(this[i++], element))\n    }\n    return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n * using
 the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline
```

fun <R, V> ShortArray.zip(other: Iterable<R>, transform: (a: Short, b: R) -> V): List<V> {\n val arraySize = size\n val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in other) {\n if (i >= arraySize) break\n list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V> IntArray.zip(other: Iterable<R>, transform:
(a: Int, b: R) -> V): List<V> {\n val arraySize = size\n val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in other) {\n if (i >= arraySize) break\n list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V> LongArray.zip(other: Iterable<R>, transform: (a: Long, b: R) -> V): List<V> {\n val arraySize = size\n val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in other) {\n if (i >= arraySize) break\n list.add(transform(this[i++], element))\n
}\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V> FloatArray.zip(other: Iterable<R>, transform: (a: Float, b: R) -> V): List<V> {\n val arraySize = size\n val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in other) {\n if (i >= arraySize) break\n list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n
* \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V> DoubleArray.zip(other: Iterable<R>, transform: (a: Double, b: R) -> V): List<V> {\n val arraySize = size\n val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in other) {\n if (i >= arraySize) break\n list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V> BooleanArray.zip(other: Iterable<R>, transform: (a: Boolean, b: R) -> V): List<V> {\n val arraySize = size\n val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10),
arraySize))\n var i = 0\n for (element in other) {\n if (i >= arraySize) break\n list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <R, V> CharArray.zip(other: Iterable<R>, transform: (a: Char, b: R) -> V): List<V> {\n val arraySize = size\n val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in other) {\n if (i >= arraySize) break\n list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array
with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun ByteArray.zip(other: ByteArray): List<Pair<Byte, Byte>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from

the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun ShortArray.zip(other: ShortArray): List<Pair<Short, Short>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun IntArray.zip(other: IntArray): List<Pair<Int, Int>> {\n
return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun LongArray.zip(other: LongArray): List<Pair<Long, Long>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun FloatArray.zip(other: FloatArray): List<Pair<Float, Float>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample
samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun DoubleArray.zip(other: DoubleArray): List<Pair<Double, Double>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun BooleanArray.zip(other: BooleanArray): List<Pair<Boolean, Boolean>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun CharArray.zip(other: CharArray): List<Pair<Char, Char>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values built
from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <V> ByteArray.zip(other: ByteArray, transform: (a: Byte, b: Byte) -> V): List<V> {\n   val size = minOf(size, other.size)\n   val list = ArrayList<V>(size)\n   for (i in 0 until size) {\n   list.add(transform(this[i], other[i]))\n   }\n   return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <V> ShortArray.zip(other: ShortArray, transform: (a: Short,
b: Short) -> V): List<V> {\n   val size = minOf(size, other.size)\n   val list = ArrayList<V>(size)\n   for (i in 0 until size) {\n   list.add(transform(this[i], other[i]))\n   }\n   return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <V> IntArray.zip(other: IntArray, transform: (a: Int, b: Int) -> V): List<V> {\n   val size = minOf(size, other.size)\n   val list = ArrayList<V>(size)\n   for (i in 0 until size) {\n   list.add(transform(this[i], other[i]))\n   }\n   return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied
to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <V> LongArray.zip(other: LongArray, transform: (a: Long, b: Long) -> V): List<V> {\n   val size = minOf(size, other.size)\n   val list = ArrayList<V>(size)\n   for (i in 0 until size) {\n   list.add(transform(this[i], other[i]))\n   }\n   return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array

with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <V> FloatArray.zip(other: FloatArray, transform: (a: Float, b: Float) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <V> DoubleArray.zip(other: DoubleArray, transform: (a: Double, b: Double) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <V> BooleanArray.zip(other: BooleanArray, transform: (a: Boolean, b: Boolean) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <V> CharArray.zip(other: CharArray, transform: (a: Char, b: Char) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n */\npublic fun <T, A : Appendable> Array<out T>.joinTo(buffer: A, separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((T) -> CharSequence)? = null): A {\n buffer.append(prefix)\n var count = 0\n for (element in this) {\n if (++count > 1) buffer.append(separator)\n if (limit < 0 || count <= limit) {\n buffer.appendElement(element, transform)\n } else break\n }\n if (limit >= 0 && count > limit) buffer.append(truncated)\n buffer.append(postfix)\n return buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n */\npublic fun <A : Appendable> ByteArray.joinTo(buffer: A, separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Byte) -> CharSequence)? = null): A {\n buffer.append(prefix)\n var count = 0\n for (element in this) {\n if (++count > 1) buffer.append(separator)\n if (limit < 0 || count <= limit) {\n if (transform != null)\n buffer.append(transform(element))\n else\n buffer.append(element.toString())\n } else break\n }\n if (limit >= 0 && count > limit) buffer.append(truncated)\n buffer.append(postfix)\n return buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n */\npublic fun <A : Appendable>

ShortArray.joinTo(buffer: A, separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Short) -> CharSequence)? = null): A {\n buffer.append(prefix)\n var count = 0\n for (element in this) {\n if (++count > 1) buffer.append(separator)\n if (limit < 0 || count <= limit) {\n if (transform != null)\n buffer.append(transform(element))\n else\n buffer.append(element.toString())\n } else break\n }\n if (limit >= 0 && count > limit) buffer.append(truncated)\n buffer.append(postfix)\n return buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n */\npublic fun <A : Appendable> IntArray.joinTo(buffer: A, separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Int) -> CharSequence)? = null): A {\n buffer.append(prefix)\n var count = 0\n for (element in this) {\n if (++count > 1) buffer.append(separator)\n if (limit < 0 || count <= limit) {\n if (transform != null)\n buffer.append(transform(element))\n else\n buffer.append(element.toString())\n } else break\n }\n if (limit >= 0 && count > limit) buffer.append(truncated)\n buffer.append(postfix)\n return buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n */\npublic fun <A : Appendable> LongArray.joinTo(buffer: A, separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Long) -> CharSequence)? = null): A {\n buffer.append(prefix)\n var count = 0\n for (element in this) {\n if (++count > 1) buffer.append(separator)\n if (limit < 0 || count <= limit) {\n if (transform != null)\n buffer.append(transform(element))\n else\n buffer.append(element.toString())\n } else break\n }\n if (limit >= 0 && count > limit) buffer.append(truncated)\n buffer.append(postfix)\n return buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n */\npublic fun <A : Appendable> FloatArray.joinTo(buffer: A, separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Float) -> CharSequence)? = null): A {\n buffer.append(prefix)\n var count = 0\n for (element in this) {\n if (++count > 1) buffer.append(separator)\n if (limit < 0 || count <= limit) {\n if (transform != null)\n buffer.append(transform(element))\n else\n buffer.append(element.toString())\n } else break\n }\n if (limit >= 0 && count > limit) buffer.append(truncated)\n buffer.append(postfix)\n return buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n */\npublic fun <A : Appendable> DoubleArray.joinTo(buffer: A, separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Double) -> CharSequence)? = null): A {\n buffer.append(prefix)\n var count = 0\n for (element in this) {\n if (++count > 1) buffer.append(separator)\n if (limit < 0 || count <= limit) {\n if (transform != null)\n buffer.append(transform(element))\n else\n buffer.append(element.toString())\n } else break\n }\n if (limit >= 0 && count > limit) buffer.append(truncated)\n buffer.append(postfix)\n return

buffer\n}\n\n/**\n * Appends the string
from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n */\npublic fun <A : Appendable> BooleanArray.joinTo(buffer: A, separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Boolean) -> CharSequence)? = null): A {\n buffer.append(prefix)\n var count = 0\n for (element in this) {\n if (++count > 1) buffer.append(separator)\n if (limit < 0 || count <= limit) {\n if (transform != null)\n buffer.append(transform(element))\n else\n buffer.append(element.toString())\n }
else break\n }\n if (limit >= 0 && count > limit) buffer.append(truncated)\n buffer.append(postfix)\n return buffer\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n */\npublic fun <A : Appendable> CharArray.joinTo(buffer: A, separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Char) -> CharSequence)? = null): A {\n buffer.append(prefix)\n var count = 0\n for (element in this) {\n if (++count > 1) buffer.append(separator)\n if (limit < 0 || count <= limit) {\n
if (transform != null)\n buffer.append(transform(element))\n else\n buffer.append(element)\n } else break\n }\n if (limit >= 0 && count > limit) buffer.append(truncated)\n buffer.append(postfix)\n return buffer\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinToString\n */\npublic fun <T> Array<out T>.joinToString(separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((T) -> CharSequence)? = null): String {\n return joinTo(StringBuilder(), separator,
prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinToString\n */\npublic fun ByteArray.joinToString(separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Byte) -> CharSequence)? = null): String {\n return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection
could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinToString\n */\npublic fun ShortArray.joinToString(separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Short) -> CharSequence)? = null): String {\n return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample

samples.collections.Collections.Transformations.joinToString\n */\npublic fun IntArray.joinToString(separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Int) -> CharSequence)? = null): String {\n return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinToString\n */\npublic fun LongArray.joinToString(separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Long) -> CharSequence)? = null): String {\n return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinToString\n */\npublic fun FloatArray.joinToString(separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Float) -> CharSequence)? = null): String {\n return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinToString\n */\npublic fun DoubleArray.joinToString(separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Double) -> CharSequence)? = null): String {\n return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinToString\n */\npublic fun BooleanArray.joinToString(separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Boolean) -> CharSequence)? = null): String {\n return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinToString\n */\npublic fun CharArray.joinToString(separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((Char) -> CharSequence)? = null): String {\n return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.\n */\npublic fun <T> Array<out T>.asIterable(): Iterable<T> {\n if (isEmpty()) return emptyList()\n return Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.\n */\npublic fun ByteArray.asIterable(): Iterable<Byte> {\n if (isEmpty()) return emptyList()\n return Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.\n */\npublic fun ShortArray.asIterable(): Iterable<Short> {\n if (isEmpty()) return emptyList()\n return Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.\n */\npublic fun IntArray.asIterable(): Iterable<Int> {\n if (isEmpty()) return emptyList()\n return Iterable { this.iterator()

}\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.\n */\npublic fun LongArray.asIterable(): Iterable<Long> {\n   if (isEmpty()) return emptyList()\n   return Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.\n */\npublic fun FloatArray.asIterable(): Iterable<Float> {\n   if (isEmpty()) return emptyList()\n   return Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.\n */\npublic
fun DoubleArray.asIterable(): Iterable<Double> {\n   if (isEmpty()) return emptyList()\n   return Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.\n */\npublic fun BooleanArray.asIterable(): Iterable<Boolean> {\n   if (isEmpty()) return emptyList()\n   return Iterable { this.iterator() }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original array returning its elements when being iterated.\n */\npublic fun CharArray.asIterable(): Iterable<Char> {\n   if (isEmpty()) return emptyList()\n   return Iterable { this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n * @sample
samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun <T> Array<out T>.asSequence(): Sequence<T> {\n   if (isEmpty()) return emptySequence()\n   return Sequence { this.iterator() }\n}\n\n/**\n
* Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun ByteArray.asSequence(): Sequence<Byte> {\n   if (isEmpty()) return emptySequence()\n   return Sequence { this.iterator() }\n}\n\n/**\n *
Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun ShortArray.asSequence(): Sequence<Short> {\n   if (isEmpty()) return emptySequence()\n   return Sequence { this.iterator() }\n}\n\n/**\n *
Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun IntArray.asSequence(): Sequence<Int> {\n   if (isEmpty()) return emptySequence()\n   return Sequence { this.iterator() }\n}\n\n/**\n
* Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun LongArray.asSequence(): Sequence<Long> {\n   if (isEmpty()) return emptySequence()\n   return Sequence { this.iterator() }\n}\n\n/**\n *
Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun FloatArray.asSequence(): Sequence<Float> {\n   if (isEmpty()) return emptySequence()\n   return Sequence { this.iterator() }\n}\n\n/**\n *
Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n *
@sample samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun DoubleArray.asSequence(): Sequence<Double> {\n   if (isEmpty()) return emptySequence()\n   return Sequence { this.iterator()
}\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n * @sample samples.collections.Sequences.Building.sequenceFromArray\n */\npublic fun
BooleanArray.asSequence(): Sequence<Boolean> {\n   if (isEmpty()) return emptySequence()\n   return Sequence { this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original array returning its elements when being iterated.\n * \n * @sample samples.collections.Sequences.Building.sequenceFromArray\n */\npublic
fun CharArray.asSequence(): Sequence<Char> {\n   if (isEmpty()) return emptySequence()\n   return Sequence { this.iterator() }\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\n@kotlin.jvm.JvmName(\"averageOfByte\")\npublic fun Array<out Byte>.average(): Double {\n   var sum: Double = 0.0\n   var count: Int = 0\n   for (element in this) {\n   sum += element\n   ++count\n   }\n   return
if (count == 0) Double.NaN
else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\n@kotlin.jvm.JvmName(\"averageOfShort\")\npublic fun Array<out Short>.average(): Double {\n   var sum: Double = 0.0\n   var count: Int = 0\n   for (element in this) {\n   sum += element\n   ++count\n   }\n   return
if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n
*/\n@kotlin.jvm.JvmName(\"averageOfInt\")\npublic fun Array<out Int>.average(): Double {\n   var sum: Double

= 0.0\n   var count: Int = 0\n   for (element in this) {\n   sum += element\n   ++count\n   }\n   return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n */\n@kotlin.jvm.JvmName(\"averageOfLong\")\npublic fun Array<out Long>.average(): Double {\n   var sum: Double = 0.0\n   var count: Int = 0\n   for (element in this) {\n   sum += element\n   ++count\n   }\n   return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n */\n@kotlin.jvm.JvmName(\"averageOfFloat\")\npublic fun Array<out Float>.average(): Double {\n   var sum: Double = 0.0\n   var count: Int = 0\n   for (element in this) {\n   sum += element\n   ++count\n   }\n   return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n */\n@kotlin.jvm.JvmName(\"averageOfDouble\")\npublic fun Array<out Double>.average(): Double {\n   var sum: Double = 0.0\n   var count: Int = 0\n   for (element in this) {\n   sum += element\n   ++count\n   }\n   return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n */\npublic fun ByteArray.average(): Double {\n   var sum: Double = 0.0\n   var count: Int = 0\n   for (element in this) {\n   sum += element\n   ++count\n   }\n   return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n */\npublic fun ShortArray.average(): Double {\n   var sum: Double = 0.0\n   var count: Int = 0\n   for (element in this) {\n   sum += element\n   ++count\n   }\n   return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n */\npublic fun IntArray.average(): Double {\n   var sum: Double = 0.0\n   var count: Int = 0\n   for (element in this) {\n   sum += element\n   ++count\n   }\n   return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n */\npublic fun LongArray.average(): Double {\n   var sum: Double = 0.0\n   var count: Int = 0\n   for (element in this) {\n   sum += element\n   ++count\n   }\n   return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n */\npublic fun FloatArray.average(): Double {\n   var sum: Double = 0.0\n   var count: Int = 0\n   for (element in this) {\n   sum += element\n   ++count\n   }\n   return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the array.\n */\npublic fun DoubleArray.average(): Double {\n   var sum: Double = 0.0\n   var count: Int = 0\n   for (element in this) {\n   sum += element\n   ++count\n   }\n   return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\n@kotlin.jvm.JvmName(\"sumOfByte\")\npublic fun Array<out Byte>.sum(): Int {\n   var sum: Int = 0\n   for (element in this) {\n   sum += element\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\n@kotlin.jvm.JvmName(\"sumOfShort\")\npublic fun Array<out Short>.sum(): Int {\n   var sum: Int = 0\n   for (element in this) {\n   sum += element\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\n@kotlin.jvm.JvmName(\"sumOfInt\")\npublic fun Array<out Int>.sum(): Int {\n   var sum: Int = 0\n   for (element in this) {\n   sum += element\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\n@kotlin.jvm.JvmName(\"sumOfLong\")\npublic fun Array<out Long>.sum(): Long {\n   var sum: Long = 0L\n   for (element in this) {\n   sum += element\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\n@kotlin.jvm.JvmName(\"sumOfFloat\")\npublic fun Array<out Float>.sum(): Float {\n   var sum: Float = 0.0f\n   for (element in this) {\n   sum += element\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\n@kotlin.jvm.JvmName(\"sumOfDouble\")\npublic fun Array<out Double>.sum(): Double {\n   var sum: Double = 0.0\n   for (element in this) {\n   sum += element\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\npublic fun ByteArray.sum(): Int {\n   var sum: Int = 0\n   for (element in this) {\n   sum += element\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\npublic fun ShortArray.sum(): Int {\n   var sum: Int = 0\n   for (element in this) {\n   sum += element\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\npublic fun IntArray.sum(): Int {\n   var sum: Int = 0\n   for (element in this) {\n   sum += element\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\npublic fun LongArray.sum(): Long {\n   var sum: Long = 0L\n   for (element in this) {\n   sum += element\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\npublic

fun FloatArray.sum(): Float {\n var sum: Float = 0.0f\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns
the sum of all elements in the array.\n */\npublic fun DoubleArray.sum(): Double {\n var sum: Double = 0.0\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"RangesKt\")\n\npackage kotlin.ranges\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.random.*\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the progression is empty.\n */\n@SinceKotlin(\"1.7\")\npublic fun IntProgression.first(): Int {\n if (isEmpty())\n throw NoSuchElementException(\"Progression $this is empty.\")\n return this.first\n}\n\n/**\n * Returns
the first element.\n * \n * @throws NoSuchElementException if the progression is empty.\n */\n@SinceKotlin(\"1.7\")\npublic fun LongProgression.first(): Long {\n if (isEmpty())\n throw NoSuchElementException(\"Progression $this is empty.\")\n return this.first\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the progression is empty.\n */\n@SinceKotlin(\"1.7\")\npublic fun CharProgression.first(): Char {\n if (isEmpty())\n throw NoSuchElementException(\"Progression $this is empty.\")\n return this.first\n}\n\n/**\n * Returns the first element, or `null` if the progression is empty.\n */\n@SinceKotlin(\"1.7\")\npublic fun IntProgression.firstOrNull(): Int? {\n return if (isEmpty()) null else this.first\n}\n\n/**\n * Returns the first element, or `null` if the progression is empty.\n */\n@SinceKotlin(\"1.7\")\npublic fun LongProgression.firstOrNull(): Long? {\n return if (isEmpty()) null else this.first\n}\n\n/**\n *
Returns the first element, or `null` if the progression is empty.\n */\n@SinceKotlin(\"1.7\")\npublic fun CharProgression.firstOrNull(): Char? {\n return if (isEmpty()) null else this.first\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the progression is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.7\")\npublic fun IntProgression.last(): Int {\n if (isEmpty())\n throw NoSuchElementException(\"Progression $this is empty.\")\n return this.last\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the progression is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.7\")\npublic fun LongProgression.last(): Long {\n if (isEmpty())\n throw NoSuchElementException(\"Progression $this is empty.\")\n return this.last\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the progression
is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.7\")\npublic fun CharProgression.last(): Char {\n if (isEmpty())\n throw NoSuchElementException(\"Progression $this is empty.\")\n return this.last\n}\n\n/**\n * Returns the last element, or `null` if the progression is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.7\")\npublic fun IntProgression.lastOrNull(): Int? {\n return if (isEmpty()) null else this.last\n}\n\n/**\n * Returns the last element, or `null` if the progression is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.7\")\npublic fun LongProgression.lastOrNull(): Long? {\n return if (isEmpty()) null else this.last\n}\n\n/**\n * Returns the last element, or `null` if the progression is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.7\")\npublic fun CharProgression.lastOrNull():
Char? {\n return if (isEmpty()) null else this.last\n}\n\n/**\n * Returns a random element from this range.\n * \n * @throws IllegalArgumentException if this range is empty.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun IntRange.random(): Int {\n return random(Random)\n}\n\n/**\n * Returns a random element from this range.\n * \n * @throws IllegalArgumentException if this range is empty.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun LongRange.random(): Long {\n return random(Random)\n}\n\n/**\n * Returns a random element from

this range.\n * \n * @throws IllegalArgumentException if this range is empty.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun CharRange.random(): Char {\n return random(Random)\n}\n\n/**\n * Returns a random element from this range using the specified source of randomness.\n * \n * @throws IllegalArgumentException if this range is empty.\n */\n@SinceKotlin(\"1.3\")\npublic fun IntRange.random(random: Random): Int {\n try {\n return random.nextInt(this)\n } catch(e: IllegalArgumentException) {\n throw NoSuchElementException(e.message)\n }\n}\n\n/**\n * Returns a random element from this range using the specified source of randomness.\n * \n * @throws IllegalArgumentException if this range is empty.\n */\n@SinceKotlin(\"1.3\")\npublic fun LongRange.random(random: Random): Long {\n try {\n return random.nextLong(this)\n } catch(e: IllegalArgumentException) {\n throw NoSuchElementException(e.message)\n }\n}\n\n/**\n * Returns a random element from this range using the specified source of randomness.\n * \n * @throws IllegalArgumentException if this range is empty.\n */\n@SinceKotlin(\"1.3\")\npublic fun CharRange.random(random: Random): Char {\n try {\n return random.nextInt(first.code, last.code + 1).toChar()\n } catch(e: IllegalArgumentException) {\n throw NoSuchElementException(e.message)\n }\n}\n\n/**\n * Returns a random element from this range, or `null` if this range is empty.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun IntRange.randomOrNull(): Int? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this range, or `null` if this range is empty.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun LongRange.randomOrNull(): Long? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this range, or `null` if this range is empty.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun CharRange.randomOrNull(): Char? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this range using the specified source of randomness, or `null` if this range is empty.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun IntRange.randomOrNull(random: Random): Int? {\n if (isEmpty())\n return null\n return random.nextInt(this)\n}\n\n/**\n * Returns a random element from this range using the specified source of randomness, or `null` if this range is empty.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun LongRange.randomOrNull(random: Random): Long? {\n if (isEmpty())\n return null\n return random.nextLong(this)\n}\n\n/**\n * Returns a random element from this range using the specified source of randomness, or `null` if this range is empty.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun CharRange.randomOrNull(random: Random): Char? {\n if (isEmpty())\n return null\n return random.nextInt(first.code, last.code + 1).toChar()\n}\n\n/**\n * Returns `true` if this range contains the specified [element].\n * \n * Always returns `false` if the [element] is `null`.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline operator fun IntRange.contains(element: Int?): Boolean {\n return element != null && contains(element)\n}\n\n/**\n * Returns `true` if this range contains the specified [element].\n * \n * Always returns `false` if the [element] is `null`.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline operator fun LongRange.contains(element: Long?): Boolean {\n return element != null && contains(element)\n}\n\n/**\n * Returns `true` if this range contains the specified [element].\n * \n * Always returns `false` if the [element] is `null`.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline operator fun CharRange.contains(element: Char?): Boolean {\n return element != null && contains(element)\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"intRangeContains\")\npublic operator fun ClosedRange<Int>.contains(value:

Byte): Boolean {\n   return contains(value.toInt())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"longRangeContains\")\npublic operator fun ClosedRange<Long>.contains(value: Byte): Boolean {\n   return contains(value.toLong())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"shortRangeContains\")\npublic operator fun ClosedRange<Short>.contains(value: Byte): Boolean {\n   return contains(value.toShort())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"doubleRangeContains\")\npublic operator fun ClosedRange<Double>.contains(value: Byte): Boolean {\n   return contains(value.toDouble())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"floatRangeContains\")\npublic operator fun ClosedRange<Float>.contains(value: Byte): Boolean {\n   return contains(value.toFloat())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"intRangeContains\")\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Int>.contains(value: Byte): Boolean {\n   return contains(value.toInt())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"longRangeContains\")\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Long>.contains(value: Byte): Boolean {\n   return contains(value.toLong())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"shortRangeContains\")\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Short>.contains(value: Byte): Boolean {\n   return contains(value.toShort())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun IntRange.contains(value: Byte): Boolean {\n   return (this as ClosedRange<Int>).contains(value)\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun LongRange.contains(value: Byte): Boolean {\n   return (this as ClosedRange<Long>).contains(value)\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"intRangeContains\")\npublic operator fun ClosedRange<Int>.contains(value: Double): Boolean {\n   return value.toIntExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"longRangeContains\")\npublic operator fun ClosedRange<Long>.contains(value: Double): Boolean {\n   return value.toLongExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"byteRangeContains\")\npublic operator fun ClosedRange<Byte>.contains(value: Double): Boolean {\n   return value.toByteExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"shortRangeContains\")\npublic operator fun

ClosedRange<Short>.contains(value: Double): Boolean {\n   return value.toShortExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"floatRangeContains\")\npublic operator fun ClosedRange<Float>.contains(value: Double): Boolean {\n   return
 contains(value.toFloat())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"intRangeContains\")\npublic operator fun ClosedRange<Int>.contains(value: Float): Boolean {\n   return value.toIntExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"longRangeContains\")\npublic operator fun ClosedRange<Long>.contains(value: Float): Boolean {\n   return value.toLongExactOrNull().let
 { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"byteRangeContains\")\npublic operator fun ClosedRange<Byte>.contains(value: Float): Boolean {\n   return value.toByteExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"shortRangeContains\")\npublic operator fun ClosedRange<Short>.contains(value: Float): Boolean {\n   return value.toShortExactOrNull().let
 { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"doubleRangeContains\")\npublic operator fun ClosedRange<Double>.contains(value: Float): Boolean {\n   return contains(value.toDouble())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"doubleRangeContains\")\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Double>.contains(value: Float): Boolean {\n   return contains(value.toDouble())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"longRangeContains\")\npublic operator fun ClosedRange<Long>.contains(value: Int): Boolean {\n   return contains(value.toLong())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"byteRangeContains\")\npublic operator fun ClosedRange<Byte>.contains(value: Int): Boolean {\n
    return value.toByteExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"shortRangeContains\")\npublic operator fun ClosedRange<Short>.contains(value: Int): Boolean {\n   return value.toShortExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"doubleRangeContains\")\npublic operator fun ClosedRange<Double>.contains(value: Int): Boolean {\n   return contains(value.toDouble())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer and floating point arguments has ambiguous
 semantics and is going to be removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"floatRangeContains\")\npublic operator fun ClosedRange<Float>.contains(value: Int): Boolean {\n   return contains(value.toFloat())\n}\n\n/**\n * Checks if the

specified [value] belongs to this range.\n
*/\n@kotlin.jvm.JvmName(\"longRangeContains\")\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Long>.contains(value: Int): Boolean {\n   return contains(value.toLong())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@kotlin.jvm.JvmName(\"byteRangeContains\")\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Byte>.contains(value: Int): Boolean {\n   return value.toByteExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@kotlin.jvm.JvmName(\"shortRangeContains\")\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Short>.contains(value: Int): Boolean {\n   return value.toShortExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@kotlin.internal.InlineOnly\npublic inline operator fun LongRange.contains(value: Int): Boolean {\n   return (this as ClosedRange<Long>).contains(value)\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@kotlin.jvm.JvmName(\"intRangeContains\")\npublic operator fun ClosedRange<Int>.contains(value: Long): Boolean {\n   return value.toIntExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"byteRangeContains\")\npublic operator fun ClosedRange<Byte>.contains(value: Long): Boolean {\n   return value.toByteExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n
 * Checks if the specified [value] belongs to this range.\n
*/\n@kotlin.jvm.JvmName(\"shortRangeContains\")\npublic operator fun ClosedRange<Short>.contains(value: Long): Boolean {\n   return value.toShortExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"doubleRangeContains\")\npublic operator fun ClosedRange<Double>.contains(value: Long): Boolean {\n   return contains(value.toDouble())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince
 = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"floatRangeContains\")\npublic operator fun ClosedRange<Float>.contains(value: Long): Boolean {\n   return contains(value.toFloat())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@kotlin.jvm.JvmName(\"intRangeContains\")\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Int>.contains(value: Long): Boolean {\n   return value.toIntExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@kotlin.jvm.JvmName(\"byteRangeContains\")\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Byte>.contains(value: Long): Boolean {\n   return value.toByteExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@kotlin.jvm.JvmName(\"shortRangeContains\")\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Short>.contains(value: Long): Boolean {\n   return value.toShortExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@kotlin.internal.InlineOnly\npublic inline operator fun IntRange.contains(value: Long): Boolean {\n   return (this as ClosedRange<Int>).contains(value)\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@kotlin.jvm.JvmName(\"intRangeContains\")\npublic operator fun ClosedRange<Int>.contains(value: Short): Boolean {\n   return contains(value.toInt())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@kotlin.jvm.JvmName(\"longRangeContains\")\npublic operator fun ClosedRange<Long>.contains(value: Short): Boolean {\n   return contains(value.toLong())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"byteRangeContains\")\npublic operator fun ClosedRange<Byte>.contains(value:

Short): Boolean {\n   return value.toByteExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"doubleRangeContains\")\npublic operator fun ClosedRange<Double>.contains(value: Short): Boolean {\n   return contains(value.toDouble())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@Deprecated(\"This `contains` operation mixing integer and floating point arguments has ambiguous semantics and is going to be removed.\")\n@DeprecatedSinceKotlin(warningSince = \"1.3\", errorSince = \"1.4\", hiddenSince = \"1.5\")\n@kotlin.jvm.JvmName(\"floatRangeContains\")\npublic operator fun ClosedRange<Float>.contains(value: Short): Boolean {\n   return contains(value.toFloat())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"intRangeContains\")\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Int>.contains(value: Short): Boolean {\n   return contains(value.toInt())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"longRangeContains\")\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Long>.contains(value: Short): Boolean {\n   return contains(value.toLong())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.jvm.JvmName(\"byteRangeContains\")\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun OpenEndRange<Byte>.contains(value: Short): Boolean {\n   return value.toByteExactOrNull().let { if (it != null) contains(it) else false }\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun IntRange.contains(value: Short): Boolean {\n   return (this as ClosedRange<Int>).contains(value)\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun LongRange.contains(value: Short): Boolean {\n   return (this as ClosedRange<Long>).contains(value)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Int.downTo(to: Byte): IntProgression {\n   return IntProgression.fromClosedRange(this, to.toInt(), -1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Long.downTo(to: Byte): LongProgression {\n   return LongProgression.fromClosedRange(this, to.toLong(), -1L)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Byte.downTo(to: Byte): IntProgression {\n   return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Short.downTo(to: Byte): IntProgression {\n   return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Char.downTo(to: Char): CharProgression {\n   return CharProgression.fromClosedRange(this, to, -1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Int.downTo(to: Int): IntProgression {\n   return IntProgression.fromClosedRange(this, to, -1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun

Long.downTo(to: Int): LongProgression {\n return LongProgression.fromClosedRange(this, to.toLong(), -1L)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Byte.downTo(to: Int): IntProgression {\n return IntProgression.fromClosedRange(this.toInt(), to, -1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Short.downTo(to: Int): IntProgression {\n return IntProgression.fromClosedRange(this.toInt(), to, -1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to]
value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Int.downTo(to: Long): LongProgression {\n return LongProgression.fromClosedRange(this.toLong(), to, -1L)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Long.downTo(to: Long): LongProgression {\n return LongProgression.fromClosedRange(this, to, -1L)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Byte.downTo(to: Long): LongProgression {\n return LongProgression.fromClosedRange(this.toLong(),
to, -1L)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Short.downTo(to: Long): LongProgression {\n return LongProgression.fromClosedRange(this.toLong(), to, -1L)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Int.downTo(to: Short): IntProgression {\n return IntProgression.fromClosedRange(this, to.toInt(), -1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n
* If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Long.downTo(to: Short): LongProgression {\n return LongProgression.fromClosedRange(this, to.toLong(), -1L)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Byte.downTo(to: Short): IntProgression {\n return IntProgression.fromClosedRange(this.toInt(), to.toInt(), -1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\npublic infix fun Short.downTo(to: Short): IntProgression {\n return IntProgression.fromClosedRange(this.toInt(), to.toInt(),
-1)\n}\n\n/**\n * Returns a progression that goes over the same range in the opposite direction with the same step.\n */\npublic fun IntProgression.reversed(): IntProgression {\n return IntProgression.fromClosedRange(last, first, -step)\n}\n\n/**\n * Returns a progression that goes over the same range in the opposite direction with the same step.\n */\npublic fun LongProgression.reversed(): LongProgression {\n return LongProgression.fromClosedRange(last, first, -step)\n}\n\n/**\n * Returns a progression that goes over the same range in the opposite direction with the same step.\n */\npublic fun CharProgression.reversed(): CharProgression {\n return CharProgression.fromClosedRange(last, first, -step)\n}\n\n/**\n * Returns a progression that goes over the same range with the given step.\n */\npublic infix fun IntProgression.step(step: Int): IntProgression {\n checkStepIsPositive(step > 0, step)\n return IntProgression.fromClosedRange(first, last, if (this.step > 0) step
else -step)\n}\n\n/**\n * Returns a progression that goes over the same range with the given step.\n */\npublic infix fun LongProgression.step(step: Long): LongProgression {\n checkStepIsPositive(step > 0, step)\n return

```
LongProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)\n}\n\n/**\n * Returns a progression that goes over the same range with the given step.\n */\npublic infix fun CharProgression.step(step: Int): CharProgression {\n    checkStepIsPositive(step > 0, step)\n    return CharProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)\n}\n\ninternal fun Int.toByteExactOrNull(): Byte? {\n    return if (this in Byte.MIN_VALUE.toInt()..Byte.MAX_VALUE.toInt()) this.toByte() else null\n}\n\ninternal fun Long.toByteExactOrNull(): Byte? {\n    return if (this in Byte.MIN_VALUE.toLong()..Byte.MAX_VALUE.toLong()) this.toByte() else null\n}\n\ninternal fun Short.toByteExactOrNull(): Byte? {\n    return if (this in Byte.MIN_VALUE.toShort()..Byte.MAX_VALUE.toShort())
 this.toByte() else null\n}\n\ninternal fun Double.toByteExactOrNull(): Byte? {\n    return if (this in Byte.MIN_VALUE.toDouble()..Byte.MAX_VALUE.toDouble()) this.toInt().toByte() else null\n}\n\ninternal fun Float.toByteExactOrNull(): Byte? {\n    return if (this in Byte.MIN_VALUE.toFloat()..Byte.MAX_VALUE.toFloat()) this.toInt().toByte() else null\n}\n\ninternal fun Long.toIntExactOrNull(): Int? {\n    return if (this in Int.MIN_VALUE.toLong()..Int.MAX_VALUE.toLong()) this.toInt() else null\n}\n\ninternal fun Double.toIntExactOrNull(): Int? {\n    return if (this in Int.MIN_VALUE.toDouble()..Int.MAX_VALUE.toDouble()) this.toInt() else null\n}\n\ninternal fun Float.toIntExactOrNull(): Int? {\n    return if (this in Int.MIN_VALUE.toFloat()..Int.MAX_VALUE.toFloat()) this.toInt() else null\n}\n\ninternal fun Double.toLongExactOrNull(): Long? {\n    return if (this in Long.MIN_VALUE.toDouble()..Long.MAX_VALUE.toDouble()) this.toLong() else null\n}\n\ninternal
 fun Float.toLongExactOrNull(): Long? {\n    return if (this in Long.MIN_VALUE.toFloat()..Long.MAX_VALUE.toFloat()) this.toLong() else null\n}\n\ninternal fun Int.toShortExactOrNull(): Short? {\n    return if (this in Short.MIN_VALUE.toInt()..Short.MAX_VALUE.toInt()) this.toShort() else null\n}\n\ninternal fun Long.toShortExactOrNull(): Short? {\n    return if (this in Short.MIN_VALUE.toLong()..Short.MAX_VALUE.toLong()) this.toShort() else null\n}\n\ninternal fun Double.toShortExactOrNull(): Short? {\n    return if (this in Short.MIN_VALUE.toDouble()..Short.MAX_VALUE.toDouble()) this.toInt().toShort() else null\n}\n\ninternal fun Float.toShortExactOrNull(): Short? {\n    return if (this in Short.MIN_VALUE.toFloat()..Short.MAX_VALUE.toFloat()) this.toInt().toShort() else null\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix
 fun Int.until(to: Byte): IntRange {\n    return this .. (to.toInt() - 1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Long.until(to: Byte): LongRange {\n    return this .. (to.toLong() - 1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Byte.until(to: Byte): IntRange {\n    return this.toInt() .. (to.toInt() - 1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Short.until(to: Byte): IntRange {\n    return this.toInt() .. (to.toInt() - 1).toInt()\n}\n\n/**\n * Returns
 a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Char.until(to: Char): CharRange {\n    if (to <= '\\u0000') return CharRange.EMPTY\n    return this .. (to - 1).toChar()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Int.until(to: Int): IntRange {\n    if (to <= Int.MIN_VALUE) return IntRange.EMPTY\n    return this .. (to - 1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Long.until(to: Int): LongRange {\n    return this .. (to.toLong() - 1).toLong()\n}\n\n/**\n * Returns a range from this value up to
```

but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Byte.until(to: Int): IntRange {\n if (to <= Int.MIN_VALUE) return IntRange.EMPTY\n return this.toInt() .. (to - 1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Short.until(to: Int): IntRange {\n if (to <= Int.MIN_VALUE) return IntRange.EMPTY\n return this.toInt() .. (to - 1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Int.until(to: Long): LongRange {\n if (to <= Long.MIN_VALUE) return LongRange.EMPTY\n return this.toLong() .. (to - 1).toLong()\n}\n\n/**\n *
Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Long.until(to: Long): LongRange {\n if (to <= Long.MIN_VALUE) return LongRange.EMPTY\n return this .. (to - 1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Byte.until(to: Long): LongRange {\n if (to <= Long.MIN_VALUE) return LongRange.EMPTY\n return this.toLong() .. (to - 1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Short.until(to: Long): LongRange {\n if (to <= Long.MIN_VALUE) return LongRange.EMPTY\n return this.toLong()
.. (to - 1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Int.until(to: Short): IntRange {\n return this .. (to.toInt() - 1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Long.until(to: Short): LongRange {\n return this .. (to.toLong() - 1).toLong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Byte.until(to: Short): IntRange {\n return this.toInt() .. (to.toInt() - 1).toInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified
[to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\npublic infix fun Short.until(to: Short): IntRange {\n return this.toInt() .. (to.toInt() - 1).toInt()\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeastComparable\n */\npublic fun <T : Comparable<T>> T.coerceAtLeast(minimumValue: T): T {\n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Byte.coerceAtLeast(minimumValue: Byte): Byte {\n return if (this < minimumValue) minimumValue
else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Short.coerceAtLeast(minimumValue: Short): Short {\n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Int.coerceAtLeast(minimumValue: Int): Int {\n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Long.coerceAtLeast(minimumValue: Long): Long {\n return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less

than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Float.coerceAtLeast(minimumValue: Float): Float {\n   return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeast\n */\npublic fun Double.coerceAtLeast(minimumValue: Double): Double {\n   return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value
 is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMostComparable\n */\npublic fun <T : Comparable<T>> T.coerceAtMost(maximumValue: T): T {\n   return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Byte.coerceAtMost(maximumValue: Byte): Byte {\n   return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n
 */\npublic fun Short.coerceAtMost(maximumValue: Short): Short {\n   return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Int.coerceAtMost(maximumValue: Int): Int {\n   return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Long.coerceAtMost(maximumValue: Long): Long {\n   return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n *
 \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Float.coerceAtMost(maximumValue: Float): Float {\n   return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtMost\n */\npublic fun Double.coerceAtMost(maximumValue: Double): Double {\n   return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceInComparable\n
 */\npublic fun <T : Comparable<T>> T.coerceIn(minimumValue: T?, maximumValue: T?): T {\n   if (minimumValue !== null && maximumValue !== null) {\n   if (minimumValue > maximumValue) throw IllegalArgumentException(\"Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.\")\n   if (this < minimumValue) return minimumValue\n   if (this > maximumValue) return maximumValue\n   }\n   else {\n   if (minimumValue !== null && this < minimumValue) return minimumValue\n   if (maximumValue !== null && this > maximumValue) return maximumValue\n   }\n   return this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample

samples.comparisons.ComparableOps.coerceIn\n */\npublic
fun Byte.coerceIn(minimumValue: Byte, maximumValue: Byte): Byte {\n if (minimumValue > maximumValue) throw IllegalArgumentException(\"Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.\")\n if (this < minimumValue) return minimumValue\n if (this > maximumValue) return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceIn\n */\npublic fun Short.coerceIn(minimumValue: Short, maximumValue: Short): Short {\n if (minimumValue > maximumValue) throw IllegalArgumentException(\"Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.\")\n if (this < minimumValue) return minimumValue\n
if (this > maximumValue) return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceIn\n */\npublic fun Int.coerceIn(minimumValue: Int, maximumValue: Int): Int {\n if (minimumValue > maximumValue) throw IllegalArgumentException(\"Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.\")\n if (this < minimumValue) return minimumValue\n if (this > maximumValue) return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value
is greater than [maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceIn\n */\npublic fun Long.coerceIn(minimumValue: Long, maximumValue: Long): Long {\n if (minimumValue > maximumValue) throw IllegalArgumentException(\"Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.\")\n if (this < minimumValue) return minimumValue\n if (this > maximumValue) return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceIn\n */\npublic fun Float.coerceIn(minimumValue: Float, maximumValue: Float): Float {\n if (minimumValue > maximumValue) throw IllegalArgumentException(\"Cannot coerce value to an empty range: maximum $maximumValue
is less than minimum $minimumValue.\")\n if (this < minimumValue) return minimumValue\n if (this > maximumValue) return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than [maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceIn\n */\npublic fun Double.coerceIn(minimumValue: Double, maximumValue: Double): Double {\n if (minimumValue > maximumValue) throw IllegalArgumentException(\"Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.\")\n if (this < minimumValue) return minimumValue\n if (this > maximumValue) return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the specified [range].\n * \n * @return this value if it's in the [range], or `range.start`
if this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.\n * \n * @sample samples.comparisons.ComparableOps.coerceInFloatingPointRange\n */\n@SinceKotlin(\"1.1\")\npublic fun <T : Comparable<T>> T.coerceIn(range: ClosedFloatingPointRange<T>): T {\n if (range.isEmpty()) throw IllegalArgumentException(\"Cannot coerce value to an empty range: $range.\")\n return when {\n // this < start equiv to this <= start && !(this >= start)\n range.lessThanOrEquals(this, range.start) && !range.lessThanOrEquals(range.start, this) -> range.start\n // this > end equiv to this >= end && !(this <= end)\n range.lessThanOrEquals(range.endInclusive, this) && !range.lessThanOrEquals(this, range.endInclusive) ->

```
range.endInclusive\n        else -> this\n    }\n}\n\n/**\n * Ensures that this value lies in the specified [range].\n * \n * @return this value if it's in the [range], or `range.start` if this value
 is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.\n * \n * @sample samples.comparisons.ComparableOps.coerceInComparable\n */\npublic fun <T : Comparable<T>>
T.coerceIn(range: ClosedRange<T>): T {\n    if (range is ClosedFloatingPointRange) {\n        return this.coerceIn<T>(range)\n    }\n    if (range.isEmpty()) throw IllegalArgumentException(\"Cannot coerce value to an empty range: $range.\")\n    return when {\n        this < range.start -> range.start\n        this > range.endInclusive -> range.endInclusive\n        else -> this\n    }\n}\n\n/**\n * Ensures that this value lies in the specified [range].\n * \n * @return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.\n * \n * @sample samples.comparisons.ComparableOps.coerceIn\n */\npublic fun Int.coerceIn(range: ClosedRange<Int>): Int {\n    if (range is
 ClosedFloatingPointRange) {\n        return this.coerceIn<Int>(range)\n    }\n    if (range.isEmpty()) throw IllegalArgumentException(\"Cannot coerce value to an empty range: $range.\")\n    return when {\n        this < range.start -> range.start\n        this > range.endInclusive -> range.endInclusive\n        else -> this\n    }\n}\n\n/**\n * Ensures that this value lies in the specified [range].\n * \n * @return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.\n * \n * @sample samples.comparisons.ComparableOps.coerceIn\n */\npublic fun Long.coerceIn(range:
ClosedRange<Long>): Long {\n    if (range is ClosedFloatingPointRange) {\n        return this.coerceIn<Long>(range)\n    }\n    if (range.isEmpty()) throw IllegalArgumentException(\"Cannot coerce value to an empty range: $range.\")\n    return when {\n        this < range.start -> range.start\n        this > range.endInclusive
 -> range.endInclusive\n        else -> this\n    }\n}\n\n","/*\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin\n\nimport kotlin.experimental.*\nimport
kotlin.jvm.*\n\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@JvmInline\npublic value class UByte @kotlin.internal.IntrinsicConstEvaluation @PublishedApi internal
constructor(@PublishedApi internal val data: Byte) : Comparable<UByte> {\n\n    companion object {\n        /**\n         * A constant holding the minimum value an instance of UByte can have.\n         */\n        public const val
MIN_VALUE: UByte = UByte(0)\n\n        /**\n         * A constant holding the maximum value an instance of UByte can have.\n         */\n        public const val MAX_VALUE: UByte = UByte(-1)\n\n
        /**\n         * The number of bytes used to represent an instance of UByte in a binary form.\n         */\n
public const val SIZE_BYTES: Int = 1\n\n        /**\n         * The number of bits used to represent an instance of UByte in a binary form.\n         */\n        public const val SIZE_BITS: Int = 8\n    }\n\n    /**\n     * Compares this value with the specified value for order.\n     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n     * or a positive number if it's greater than other.\n     */\n    @kotlin.internal.InlineOnly\n    @Suppress(\"OVERRIDE_BY_INLINE\")\n    public override inline operator fun compareTo(other: UByte): Int = this.toInt().compareTo(other.toInt())\n\n    /**\n     * Compares this value with the specified value for order.\n     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n     * or a positive number if it's
 greater than other.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun compareTo(other: UShort): Int = this.toInt().compareTo(other.toInt())\n\n    /**\n     * Compares this value with the specified value for order.\n
 * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n     * or a positive number if it's greater than other.\n     */\n    @kotlin.internal.InlineOnly\n    public inline operator fun compareTo(other: UInt): Int = this.toUInt().compareTo(other)\n\n    /**\n     * Compares this value with the specified value for order.\n     * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n     * or a positive number if it's greater than other.\n     */\n    @kotlin.internal.InlineOnly\n
public inline operator fun compareTo(other: ULong): Int = this.toULong().compareTo(other)\n\n    /** Adds the
```

other value to this value. */\n   @kotlin.internal.InlineOnly\n
    public inline operator fun plus(other: UByte): UInt = this.toUInt().plus(other.toUInt())\n   /** Adds the other value to this value. */\n   @kotlin.internal.InlineOnly\n   public inline operator fun plus(other: UShort): UInt = this.toUInt().plus(other.toUInt())\n   /** Adds the other value to this value. */\n   @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UInt): UInt = this.toUInt().plus(other)\n   /** Adds the other value to this value. */\n   @kotlin.internal.InlineOnly\n   public inline operator fun plus(other: ULong): ULong = this.toULong().plus(other)\n\n   /** Subtracts the other value from this value. */\n   @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UByte): UInt = this.toUInt().minus(other.toUInt())\n   /** Subtracts the other value from this value. */\n   @kotlin.internal.InlineOnly\n   public inline operator fun minus(other: UShort): UInt = this.toUInt().minus(other.toUInt())\n   /**
 Subtracts the other value from this value. */\n   @kotlin.internal.InlineOnly\n   public inline operator fun minus(other: UInt): UInt = this.toUInt().minus(other)\n   /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n   public inline operator fun minus(other: ULong): ULong = this.toULong().minus(other)\n\n   /** Multiplies this value by the other value. */\n   @kotlin.internal.InlineOnly\n public inline operator fun times(other: UByte): UInt = this.toUInt().times(other.toUInt())\n   /** Multiplies this value by the other value. */\n   @kotlin.internal.InlineOnly\n   public inline operator fun times(other: UShort): UInt = this.toUInt().times(other.toUInt())\n   /** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n   public inline operator fun times(other: UInt): UInt = this.toUInt().times(other)\n /** Multiplies this value by the other value. */\n   @kotlin.internal.InlineOnly\n   public inline operator
 fun times(other: ULong): ULong = this.toULong().times(other)\n\n   /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n   @kotlin.internal.InlineOnly\n   public inline operator fun div(other: UByte): UInt = this.toUInt().div(other.toUInt())\n   /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n   @kotlin.internal.InlineOnly\n   public inline operator fun div(other: UShort): UInt = this.toUInt().div(other.toUInt())\n   /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n   @kotlin.internal.InlineOnly\n   public inline operator fun div(other: UInt): UInt = this.toUInt().div(other)\n   /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n   @kotlin.internal.InlineOnly\n   public inline operator fun div(other: ULong): ULong = this.toULong().div(other)\n\n   /**\n
     * Calculates the remainder of truncating division of this value by the other value.\n   * \n   * The result is always less than the divisor.\n   */\n   @kotlin.internal.InlineOnly\n   public inline operator fun rem(other: UByte): UInt = this.toUInt().rem(other.toUInt())\n   /**\n   * Calculates the remainder of truncating division of this value by the other value.\n   * \n   * The result is always less than the divisor.\n   */\n @kotlin.internal.InlineOnly\n   public inline operator fun rem(other: UShort): UInt = this.toUInt().rem(other.toUInt())\n   /**\n   * Calculates the remainder of truncating division of this value by the other value.\n   * \n   * The result is always less than the divisor.\n   */\n   @kotlin.internal.InlineOnly\n   public inline operator fun rem(other: UInt): UInt = this.toUInt().rem(other)\n   /**\n   * Calculates the remainder of truncating division of this value by the other value.\n   * \n   * The result is always
 less than the divisor.\n   */\n   @kotlin.internal.InlineOnly\n   public inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)\n\n   /**\n   * Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n   * \n   * For unsigned types, the results of flooring division and truncating division are the same.\n   */\n   @kotlin.internal.InlineOnly\n   public inline fun floorDiv(other: UByte): UInt = this.toUInt().floorDiv(other.toUInt())\n   /**\n   * Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n   * \n   * For unsigned types, the results of flooring division and truncating division are the same.\n   */\n   @kotlin.internal.InlineOnly\n   public inline fun floorDiv(other: UShort): UInt = this.toUInt().floorDiv(other.toUInt())\n   /**\n   * Divides this value by the other value, flooring the result to an integer that is closer
 to negative infinity.\n   * \n   * For unsigned types, the results of flooring division and truncating division are the same.\n   */\n   @kotlin.internal.InlineOnly\n   public inline fun floorDiv(other: UInt): UInt =

this.toUInt().floorDiv(other)\n   /**\n   * Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n   * \n   * For unsigned types, the results of flooring division and truncating division are the same.\n   */\n   @kotlin.internal.InlineOnly\n   public inline fun floorDiv(other: ULong): ULong = this.toULong().floorDiv(other)\n\n   /**\n   * Calculates the remainder of flooring division of this value by the other value.\n   * \n   * The result is always less than the divisor.\n   * \n   * For unsigned types, the remainders of flooring division and truncating division are the same.\n   */\n   @kotlin.internal.InlineOnly\n   public inline fun mod(other: UByte): UByte = this.toUInt().mod(other.toUInt()).toUByte()\n   /**\n   * Calculates the remainder of flooring division of this value by the other value.\n   * \n   * The result is always less than the divisor.\n   * \n   * For unsigned types, the remainders of flooring division and truncating division are the same.\n   */\n   @kotlin.internal.InlineOnly\n   public inline fun mod(other: UShort): UShort = this.toUInt().mod(other.toUInt()).toUShort()\n   /**\n   * Calculates the remainder of flooring division of this value by the other value.\n   * \n   * The result is always less than the divisor.\n   * \n   * For unsigned types, the remainders of flooring division and truncating division are the same.\n   */\n   @kotlin.internal.InlineOnly\n   public inline fun mod(other: UInt): UInt = this.toUInt().mod(other)\n   /**\n   * Calculates the remainder of flooring division of this value by the other value.\n   * \n   * The result is always less than the divisor.\n   * \n   * For unsigned
types, the remainders of flooring division and truncating division are the same.\n   */\n   @kotlin.internal.InlineOnly\n   public inline fun mod(other: ULong): ULong = this.toULong().mod(other)\n\n   /**\n   * Returns this value incremented by one.\n   *\n   * @sample samples.misc.Builtins.inc\n   */\n   @kotlin.internal.InlineOnly\n   public inline operator fun inc(): UByte = UByte(data.inc())\n\n   /**\n   * Returns this value decremented by one.\n   *\n   * @sample samples.misc.Builtins.dec\n   */\n   @kotlin.internal.InlineOnly\n   public inline operator fun dec(): UByte = UByte(data.dec())\n\n   /** Creates a range from this value to the specified [other] value. */\n   @kotlin.internal.InlineOnly\n   public inline operator fun rangeTo(other: UByte): UIntRange = UIntRange(this.toUInt(), other.toUInt())\n\n   /**\n   * Creates a range from this value up to but excluding the specified [other] value.\n   *\n   * If the [other] value is less
than or equal to `this` value, then the returned range is empty.\n   */\n   @SinceKotlin(\"1.7\")\n   @ExperimentalStdlibApi\n   @kotlin.internal.InlineOnly\n   public inline operator fun rangeUntil(other: UByte): UIntRange = this.toUInt() until other.toUInt()\n\n   /** Performs a bitwise AND operation between the two values. */\n   @kotlin.internal.InlineOnly\n   public inline infix fun and(other: UByte): UByte = UByte(this.data and other.data)\n   /** Performs a bitwise OR operation between the two values. */\n   @kotlin.internal.InlineOnly\n   public inline infix fun or(other: UByte): UByte = UByte(this.data or other.data)\n   /** Performs a bitwise XOR operation between the two values. */\n   @kotlin.internal.InlineOnly\n   public inline infix fun xor(other: UByte): UByte = UByte(this.data xor other.data)\n   /** Inverts the bits in this value. */\n   @kotlin.internal.InlineOnly\n   public inline fun inv(): UByte = UByte(data.inv())\n\n   /**\n   * Converts this
[UByte] value to [Byte].\n   *\n   * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value represents\n   * the same numerical value as this `UByte`. Otherwise the result is negative.\n   *\n   * The resulting `Byte` value has the same binary representation as this `UByte` value.\n   */\n   @kotlin.internal.InlineOnly\n   public inline fun toByte(): Byte = data\n   /**\n   * Converts this [UByte] value to [Short].\n   *\n   * The resulting `Short` value represents the same numerical value as this `UByte`.\n   *\n   * The least significant 8 bits of the resulting `Short` value are the same as the bits of this `UByte` value,\n   * whereas the most significant 8 bits are filled with zeros.\n   */\n   @kotlin.internal.InlineOnly\n   public inline fun toShort(): Short = data.toShort() and 0xFF\n   /**\n   * Converts this [UByte] value to [Int].\n   *\n   * The resulting `Int` value represents the same numerical value as
this `UByte`.\n   *\n   * The least significant 8 bits of the resulting `Int` value are the same as the bits of this `UByte` value,\n   * whereas the most significant 24 bits are filled with zeros.\n   */\n   @kotlin.internal.InlineOnly\n   public inline fun toInt(): Int = data.toInt() and 0xFF\n   /**\n   * Converts this [UByte] value to [Long].\n   *\n   * The resulting `Long` value represents the same numerical value as this `UByte`.\n   *\n   * The least significant 8 bits of the resulting `Long` value are the same as the bits of this

`UByte` value,\n * whereas the most significant 56 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toLong(): Long = data.toLong() and 0xFF\n\n /** Returns this value. */\n @kotlin.internal.InlineOnly\n public inline fun toUByte(): UByte = this\n /**\n * Converts this [UByte] value to [UShort].\n *\n * The resulting `UShort` value represents the same numerical value as this `UByte`.\n *\n * The least significant 8 bits of the resulting `UShort` value are the same as the bits of this `UByte` value,\n * whereas the most significant 8 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toUShort(): UShort = UShort(data.toShort() and 0xFF)\n /**\n * Converts this [UByte] value to [UInt].\n *\n * The resulting `UInt` value represents the same numerical value as this `UByte`.\n *\n * The least significant 8 bits of the resulting `UInt` value are the same as the bits of this `UByte` value,\n * whereas the most significant 24 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toUInt(): UInt = UInt(data.toInt() and 0xFF)\n /**\n * Converts this [UByte] value to [ULong].\n *\n * The resulting `ULong` value represents the same numerical value as this `UByte`.\n *\n * The least significant 8 bits of the resulting `ULong` value are the same as the bits of this `UByte` value,\n * whereas the most significant 56 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toULong(): ULong = ULong(data.toLong() and 0xFF)\n\n /**\n * Converts this [UByte] value to [Float].\n *\n * The resulting `Float` value represents the same numerical value as this `UByte`.\n */\n @kotlin.internal.InlineOnly\n public inline fun toFloat(): Float = this.toInt().toFloat()\n /**\n * Converts this [UByte] value to [Double].\n *\n * The resulting `Double` value represents the same numerical value as this `UByte`.\n */\n @kotlin.internal.InlineOnly\n public inline fun toDouble(): Double = this.toInt().toDouble()\n\n public override fun toString(): String = toInt().toString()\n\n}\n\n/**\n * Converts this [Byte] value to [UByte].\n *\n * If this value is positive, the resulting `UByte` value represents the same numerical value as this `Byte`.\n *\n * The resulting `UByte` value has the same binary representation as this `Byte` value.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.toUByte(): UByte = UByte(this)\n/**\n * Converts this [Short] value to [UByte].\n *\n * If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents\n * the same numerical value as this `Short`.\n *\n * The resulting `UByte` value is represented by the least significant 8 bits of this `Short` value.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.toUByte(): UByte = UByte(this.toByte())\n/**\n * Converts this [Int] value to [UByte].\n *\n * If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents\n * the same numerical value as this `Int`.\n *\n * The resulting `UByte` value is represented by the least significant 8 bits of this `Int` value.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Int.toUByte(): UByte = UByte(this.toByte())\n/**\n * Converts this [Long] value to [UByte].\n *\n * If this value is positive and less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents\n * the same numerical value as this `Long`.\n *\n * The resulting `UByte` value is represented by the least significant 8 bits of this `Long` value.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Long.toUByte(): UByte = UByte(this.toByte())\n","/*\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin\n\nimport kotlin.experimental.*\nimport kotlin.jvm.*\n\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@JvmInline\npublic value class UInt @kotlin.internal.IntrinsicConstEvaluation @PublishedApi internal constructor(@PublishedApi internal val data: Int) : Comparable<UInt> {\n\n companion object {\n /**\n * A constant holding the minimum value an instance of UInt can have.\n */\n public const val MIN_VALUE: UInt = UInt(0)\n\n /**\n * A constant holding the maximum value an instance of UInt can have.\n */\n public const val

MAX_VALUE: UInt = UInt(-1)\n\n /**\n * The number of bytes used to represent an instance of UInt in a binary form.\n */\n public const val SIZE_BYTES: Int = 4\n\n /**\n * The number of bits used to represent an instance of UInt in a binary form.\n */\n public const val SIZE_BITS: Int = 32\n }\n\n /**\n * Compares this value with the specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun compareTo(other: UByte): Int = this.compareTo(other.toUInt())\n\n /**\n * Compares this value with the specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun compareTo(other: UShort): Int = this.compareTo(other.toUInt())\n\n /**\n * Compares this value with the specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n @Suppress(\"OVERRIDE_BY_INLINE\")\n public override inline operator fun compareTo(other: UInt): Int = uintCompare(this.data, other.data)\n\n /**\n * Compares this value with the specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun compareTo(other: ULong): Int = this.toULong().compareTo(other)\n\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UByte): UInt = this.plus(other.toUInt())\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UShort): UInt = this.plus(other.toUInt())\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UInt): UInt = UInt(this.data.plus(other.data))\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: ULong): ULong = this.toULong().plus(other)\n\n /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UByte): UInt = this.minus(other.toUInt())\n /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UShort): UInt = this.minus(other.toUInt())\n /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UInt): UInt = UInt(this.data.minus(other.data))\n /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: ULong): ULong = this.toULong().minus(other)\n\n /** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: UByte): UInt = this.times(other.toUInt())\n /** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: UShort): UInt = this.times(other.toUInt())\n /** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: UInt): UInt = UInt(this.data.times(other.data))\n /** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: ULong): ULong = this.toULong().times(other)\n\n /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: UByte): UInt = this.div(other.toUInt())\n /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: UShort): UInt = this.div(other.toUInt())\n /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: UInt): UInt = uintDivide(this, other)\n /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: ULong): ULong = this.toULong().div(other)\n\n /**\n * Calculates the remainder of truncating division of this value by the other value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UByte): UInt = this.rem(other.toUInt())\n /**\n * Calculates the remainder of

truncating division of this value by the other value.\n * \n * The result
is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UShort): UInt = this.rem(other.toUInt())\n /**\n * Calculates the remainder of truncating division of this value by the other value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UInt): UInt = uintRemainder(this, other)\n /**\n * Calculates the remainder of truncating division of this value by the other value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)\n\n /**\n * Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n
public inline fun floorDiv(other: UByte): UInt = this.floorDiv(other.toUInt())\n /**\n * Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UShort): UInt = this.floorDiv(other.toUInt())\n /**\n * Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UInt): UInt = div(other)\n /**\n * Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and
truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: ULong): ULong = this.toULong().floorDiv(other)\n\n /**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UByte): UByte = this.mod(other.toUInt()).toUByte()\n /**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UShort): UShort = this.mod(other.toUInt()).toUShort()\n /**\n * Calculates the remainder
of flooring division of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UInt): UInt = rem(other)\n /**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: ULong): ULong = this.toULong().mod(other)\n\n /**\n * Returns this value incremented by one.\n *\n * @sample samples.misc.Builtins.inc\n */\n @kotlin.internal.InlineOnly\n public inline operator fun inc(): UInt = UInt(data.inc())\n\n /**\n * Returns this value decremented by one.\n *\n
* @sample samples.misc.Builtins.dec\n */\n @kotlin.internal.InlineOnly\n public inline operator fun dec(): UInt = UInt(data.dec())\n\n /** Creates a range from this value to the specified [other] value. */\n @kotlin.internal.InlineOnly\n public inline operator fun rangeTo(other: UInt): UIntRange = UIntRange(this, other)\n\n /**\n * Creates a range from this value up to but excluding the specified [other] value.\n *\n * If the [other] value is less than or equal to `this` value, then the returned range is empty.\n */\n @SinceKotlin(\"1.7\")\n @ExperimentalStdlibApi\n @kotlin.internal.InlineOnly\n public inline operator fun rangeUntil(other: UInt): UIntRange = this until other\n\n /**\n * Shifts this value left by the [bitCount] number of bits.\n *\n * Note that only the five lowest-order bits of the [bitCount] are used as the shift distance.\n * The shift distance actually used is therefore always in the range `0..31`.\n
*/\n @kotlin.internal.InlineOnly\n public inline infix fun shl(bitCount: Int): UInt = UInt(data shl bitCount)\n\n /**\n * Shifts this value right by the [bitCount] number of bits, filling the leftmost bits with

zeros.\n *\n * Note that only the five lowest-order bits of the [bitCount] are used as the shift distance.\n * The shift distance actually used is therefore always in the range `0..31`.\n */\n @kotlin.internal.InlineOnly\n public inline infix fun shr(bitCount: Int): UInt = UInt(data ushr bitCount)\n\n /** Performs a bitwise AND operation between the two values. */\n @kotlin.internal.InlineOnly\n public inline infix fun and(other: UInt): UInt = UInt(this.data and other.data)\n /** Performs a bitwise OR operation between the two values. */\n @kotlin.internal.InlineOnly\n public inline infix fun or(other: UInt): UInt = UInt(this.data or other.data)\n /** Performs a bitwise XOR operation between the two values. */\n @kotlin.internal.InlineOnly\n public inline infix fun xor(other: UInt): UInt = UInt(this.data xor other.data)\n /** Inverts the bits in this value. */\n @kotlin.internal.InlineOnly\n public inline fun inv(): UInt = UInt(data.inv())\n\n /**\n * Converts this [UInt] value to [Byte].\n *\n * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value represents\n * the same numerical value as this `UInt`.\n *\n * The resulting `Byte` value is represented by the least significant 8 bits of this `UInt` value.\n * Note that the resulting `Byte` value may be negative.\n */\n @kotlin.internal.InlineOnly\n public inline fun toByte(): Byte = data.toByte()\n /**\n * Converts this [UInt] value to [Short].\n *\n * If this value is less than or equals to [Short.MAX_VALUE], the resulting `Short` value represents\n * the same numerical value as this `UInt`.\n *\n * The resulting `Short` value is represented by the least significant 16 bits of this `UInt` value.\n * Note that the resulting `Short` value may be negative.\n */\n @kotlin.internal.InlineOnly\n public inline fun toShort(): Short = data.toShort()\n /**\n * Converts this [UInt] value to [Int].\n *\n * If this value is less than or equals to [Int.MAX_VALUE], the resulting `Int` value represents\n * the same numerical value as this `UInt`. Otherwise the result is negative.\n *\n * The resulting `Int` value has the same binary representation as this `UInt` value.\n */\n @kotlin.internal.InlineOnly\n public inline fun toInt(): Int = data\n /**\n * Converts this [UInt] value to [Long].\n *\n * The resulting `Long` value represents the same numerical value as this `UInt`.\n *\n * The least significant 32 bits of the resulting `Long` value are the same as the bits of this `UInt` value,\n * whereas the most significant 32 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toLong(): Long = data.toLong() and 0xFFFF_FFFF\n\n /**\n * Converts this [UInt] value to [UByte].\n *\n * If this value is less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents\n * the same numerical value as this `UInt`.\n *\n * The resulting `UByte` value is represented by the least significant 8 bits of this `UInt` value.\n */\n @kotlin.internal.InlineOnly\n public inline fun toUByte(): UByte = data.toUByte()\n /**\n * Converts this [UInt] value to [UShort].\n *\n * If this value is less than or equals to [UShort.MAX_VALUE], the resulting `UShort` value represents\n * the same numerical value as this `UInt`.\n *\n * The resulting `UShort` value is represented by the least significant 16 bits of this `UInt` value.\n */\n @kotlin.internal.InlineOnly\n public inline fun toUShort(): UShort = data.toUShort()\n /** Returns this value. */\n @kotlin.internal.InlineOnly\n public inline fun toUInt(): UInt = this\n /**\n * Converts this [UInt] value to [ULong].\n *\n * The resulting `ULong` value represents the same numerical value as this `UInt`.\n *\n * The least significant 32 bits of the resulting `ULong` value are the same as the bits of this `UInt` value,\n * whereas the most significant 32 bits are filled with zeros.\n */\n @kotlin.internal.InlineOnly\n public inline fun toULong(): ULong = ULong(data.toLong() and 0xFFFF_FFFF)\n\n /**\n * Converts this [UInt] value to [Float].\n *\n * The resulting value is the closest `Float` to this `UInt` value.\n * In case when this `UInt` value is exactly between two `Float`s,\n * the one with zero at least significant bit of mantissa is selected.\n */\n @kotlin.internal.InlineOnly\n public inline fun toFloat(): Float = this.toDouble().toFloat()\n /**\n * Converts this [UInt] value to [Double].\n *\n * The resulting `Double` value represents the same numerical value as this `UInt`.\n */\n @kotlin.internal.InlineOnly\n public inline fun toDouble(): Double = uintToDouble(data)\n\n public override fun toString(): String = toLong().toString()\n\n}\n\n/**\n * Converts this [Byte] value to [UInt].\n *\n * If this value is positive, the resulting `UInt` value represents the same numerical value as this `Byte`.\n *\n * The least significant 8 bits of the resulting `UInt` value are the same as the bits of this `Byte` value,\n * whereas the most significant 24

bits are filled with the sign bit of this value.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.toUInt(): UInt = UInt(this.toInt())\n/**\n * Converts this [Short] value to [UInt].\n *\n * If this value is positive, the resulting `UInt` value represents the same numerical value as this `Short`.\n *\n * The least significant
16 bits of the resulting `UInt` value are the same as the bits of this `Short` value,\n * whereas the most significant 16 bits are filled with the sign bit of this value.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.toUInt(): UInt = UInt(this.toInt())\n/**\n * Converts this [Int] value to [UInt].\n *\n * If this value is positive, the resulting `UInt` value represents the same numerical value as this `Int`.\n *\n * The resulting `UInt` value has the same binary representation as this `Int` value.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Int.toUInt(): UInt = UInt(this)\n/**\n * Converts this [Long] value to [UInt].\n *\n * If this value is positive and less than or equals to [UInt.MAX_VALUE], the resulting `UInt` value represents\n * the same numerical value as this `Long`.\n *\n * The resulting `UInt`
value is represented by the least significant 32 bits of this `Long` value.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Long.toUInt(): UInt = UInt(this.toInt())\n\n/**\n * Converts this [Float] value to [UInt].\n *\n * The fractional part, if any, is rounded down towards zero.\n * Returns zero if this `Float` value is negative or `NaN`, [UInt.MAX_VALUE] if it's bigger than `UInt.MAX_VALUE`.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Float.toUInt(): UInt = doubleToUInt(this.toDouble())\n/**\n * Converts this [Double] value to [UInt].\n *\n * The fractional part, if any, is rounded down towards zero.\n * Returns zero if this `Double` value is negative or `NaN`, [UInt.MAX_VALUE] if it's bigger than `UInt.MAX_VALUE`.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic
inline fun Double.toUInt(): UInt = doubleToUInt(this)\n","/*\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin\n\nimport kotlin.experimental.*\nimport
kotlin.jvm.*\n\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@JvmInline\npublic value class UShort @kotlin.internal.IntrinsicConstEvaluation @PublishedApi internal constructor(@PublishedApi internal val data: Short) : Comparable<UShort> {\n\n   companion object {\n   /**\n   * A constant holding the minimum value an instance of UShort can have.\n   */\n   public const val MIN_VALUE: UShort = UShort(0)\n\n   /**\n   * A constant holding the maximum value an instance of UShort can have.\n   */\n   public const val MAX_VALUE: UShort = UShort(-1)\n\n   /**\n   * The number of bytes used to represent an instance of UShort in a binary form.\n   */\n   public const val SIZE_BYTES: Int = 2\n\n   /**\n   * The number of bits used to represent an instance of UShort in a binary form.\n   */\n   public const val SIZE_BITS: Int = 16\n   }\n\n   /**\n   * Compares this value with the specified value for order.\n   * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n   * or a positive number if it's greater than other.\n   */\n   @kotlin.internal.InlineOnly\n   public inline operator fun compareTo(other: UByte): Int = this.toInt().compareTo(other.toInt())\n\n   /**\n   * Compares this value with the specified value for order.\n   * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n   * or a positive number if it's greater than other.\n   */\n   @kotlin.internal.InlineOnly\n   @Suppress(\"OVERRIDE_BY_INLINE\")\n   public override inline operator fun compareTo(other: UShort): Int = this.toInt().compareTo(other.toInt())\n\n   /**\n   * Compares this value with the specified value for order.\n   * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n   * or a

positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun compareTo(other: UInt): Int = this.toUInt().compareTo(other)\n\n /**\n * Compares this value with the specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun compareTo(other: ULong): Int = this.toULong().compareTo(other)\n\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UByte): UInt = this.toUInt().plus(other.toUInt())\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UShort): UInt = this.toUInt().plus(other.toUInt())\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UInt): UInt = this.toUInt().plus(other)\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: ULong): ULong = this.toULong().plus(other)\n\n /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UByte): UInt = this.toUInt().minus(other.toUInt())\n /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UShort): UInt = this.toUInt().minus(other.toUInt())\n /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UInt): UInt = this.toUInt().minus(other)\n /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: ULong): ULong = this.toULong().minus(other)\n\n /** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: UByte): UInt = this.toUInt().times(other.toUInt())\n /** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: UShort): UInt = this.toUInt().times(other.toUInt())\n /** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: UInt): UInt = this.toUInt().times(other)\n /** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: ULong): ULong = this.toULong().times(other)\n\n /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: UByte): UInt = this.toUInt().div(other.toUInt())\n /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: UShort): UInt = this.toUInt().div(other.toUInt())\n /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: UInt): UInt = this.toUInt().div(other)\n /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: ULong): ULong = this.toULong().div(other)\n\n /**\n * Calculates the remainder of truncating division of this value by the other value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UByte): UInt = this.toUInt().rem(other.toUInt())\n /**\n * Calculates the remainder of truncating division of this value by the other value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UShort): UInt = this.toUInt().rem(other.toUInt())\n /**\n * Calculates the remainder of truncating division of this value by the other value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UInt): UInt = this.toUInt().rem(other)\n /**\n * Calculates the remainder of truncating division of this value by the other value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: ULong): ULong = this.toULong().rem(other)\n\n /**\n * Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UByte): UInt = this.toUInt().floorDiv(other.toUInt())\n /**\n * Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring

division and truncating division are the same.\n * */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UShort): UInt = this.toUInt().floorDiv(other.toUInt())\n /**\n * Divides this value by the other value, flooring the result to an integer that is closer
to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UInt): UInt = this.toUInt().floorDiv(other)\n /**\n * Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: ULong): ULong = this.toULong().floorDiv(other)\n\n /**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UByte): UByte = this.toUInt().mod(other.toUInt()).toUByte()\n
/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UShort): UShort = this.toUInt().mod(other.toUInt()).toUShort()\n /**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UInt): UInt = this.toUInt().mod(other)\n /**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For
unsigned types, the remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: ULong): ULong = this.toULong().mod(other)\n\n /**\n * Returns this value incremented by one.\n *\n * @sample samples.misc.Builtins.inc\n */\n @kotlin.internal.InlineOnly\n public inline operator fun inc(): UShort = UShort(data.inc())\n\n /**\n * Returns this value decremented by one.\n *\n * @sample samples.misc.Builtins.dec\n */\n @kotlin.internal.InlineOnly\n public inline operator fun dec(): UShort = UShort(data.dec())\n\n /** Creates a range from this value to the specified [other] value. */\n @kotlin.internal.InlineOnly\n public inline operator fun rangeTo(other: UShort): UIntRange = UIntRange(this.toUInt(), other.toUInt())\n\n /**\n * Creates a range from this value up to but excluding the specified [other] value.\n *\n * If the [other] value is
less than or equal to `this` value, then the returned range is empty.\n */\n @SinceKotlin(\"1.7\")\n @ExperimentalStdlibApi\n @kotlin.internal.InlineOnly\n public inline operator fun rangeUntil(other: UShort): UIntRange = this.toUInt() until other.toUInt()\n\n /** Performs a bitwise AND operation between the two values. */\n @kotlin.internal.InlineOnly\n public inline infix fun and(other: UShort): UShort = UShort(this.data and other.data)\n /** Performs a bitwise OR operation between the two values. */\n @kotlin.internal.InlineOnly\n public inline infix fun or(other: UShort): UShort = UShort(this.data or other.data)\n /** Performs a bitwise XOR operation between the two values. */\n @kotlin.internal.InlineOnly\n public inline infix fun xor(other: UShort): UShort = UShort(this.data xor other.data)\n /** Inverts the bits in this value. */\n @kotlin.internal.InlineOnly\n public inline fun inv(): UShort = UShort(data.inv())\n\n /**\n
* Converts this [UShort] value to [Byte].\n *\n * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value represents\n * the same numerical value as this `UShort`.\n *\n * The resulting `Byte` value is represented by the least significant 8 bits of this `UShort` value.\n * Note that the resulting `Byte` value may be negative.\n */\n @kotlin.internal.InlineOnly\n public inline fun toByte(): Byte = data.toByte()\n /**\n * Converts this [UShort] value to [Short].\n *\n * If this value is less than or equals to [Short.MAX_VALUE], the resulting `Short` value represents\n * the same numerical value as this `UShort`. Otherwise the result is negative.\n *\n * The resulting `Short` value has the same binary representation as this `UShort` value.\n */\n @kotlin.internal.InlineOnly\n public inline fun toShort(): Short = data\n /**\n * Converts this [UShort] value to [Int].\n *\n

```
 * The resulting `Int` value represents the same numerical value as this `UShort`.\n     *\n     * The least significant 16 bits of the resulting `Int` value are the same as the bits of this `UShort` value,\n     * whereas the most significant 16 bits are filled with zeros.\n     */\n    @kotlin.internal.InlineOnly\n    public inline fun toInt(): Int = data.toInt() and 0xFFFF\n    /**\n     * Converts this [UShort] value to [Long].\n     *\n     * The resulting `Long` value represents the same numerical value as this `UShort`.\n     *\n     * The least significant 16 bits of the resulting `Long` value are the same as the bits of this `UShort` value,\n     * whereas the most significant 48 bits are filled with zeros.\n     */\n    @kotlin.internal.InlineOnly\n    public inline fun toLong(): Long = data.toLong() and 0xFFFF\n\n    /**\n     * Converts this [UShort] value to [UByte].\n     *\n     * If this value is less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents\n
     * the same numerical value as this `UShort`.\n     *\n     * The resulting `UByte` value is represented by the least significant 8 bits of this `UShort` value.\n     */\n    @kotlin.internal.InlineOnly\n    public inline fun toUByte(): UByte = data.toUByte()\n    /** Returns this value. */\n    @kotlin.internal.InlineOnly\n    public inline fun toUShort(): UShort = this\n    /**\n     * Converts this [UShort] value to [UInt].\n     *\n     * The resulting `UInt` value represents the same numerical value as this `UShort`.\n     *\n     * The least significant 16 bits of the resulting `UInt` value are the same as the bits of this `UShort` value,\n     * whereas the most significant 16 bits are filled with zeros.\n     */\n    @kotlin.internal.InlineOnly\n    public inline fun toUInt(): UInt = UInt(data.toInt() and 0xFFFF)\n    /**\n     * Converts this [UShort] value to [ULong].\n     *\n     * The resulting `ULong` value represents the same numerical value as this `UShort`.\n
     *\n     * The least significant 16 bits of the resulting `ULong` value are the same as the bits of this `UShort` value,\n     * whereas the most significant 48 bits are filled with zeros.\n     */\n    @kotlin.internal.InlineOnly\n    public inline fun toULong(): ULong = ULong(data.toLong() and 0xFFFF)\n\n    /**\n     * Converts this [UShort] value to [Float].\n     *\n     * The resulting `Float` value represents the same numerical value as this `UShort`.\n     */\n    @kotlin.internal.InlineOnly\n    public inline fun toFloat(): Float = this.toInt().toFloat()\n    /**\n     * Converts this [UShort] value to [Double].\n     *\n     * The resulting `Double` value represents the same numerical value as this `UShort`.\n     */\n    @kotlin.internal.InlineOnly\n    public inline fun toDouble(): Double = this.toInt().toDouble()\n\n    public override fun toString(): String = toInt().toString()\n\n}\n\n/**\n * Converts this [Byte] value to [UShort].\n *\n * If this value is positive,
 the resulting `UShort` value represents the same numerical value as this `Byte`.\n *\n * The least significant 8 bits of the resulting `UShort` value are the same as the bits of this `Byte` value,\n * whereas the most significant 8 bits are filled with the sign bit of this value.\n
 */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.toUShort(): UShort = UShort(this.toShort())\n/**\n * Converts this [Short] value to [UShort].\n *\n * If this value is positive, the resulting `UShort` value represents the same numerical value as this `Short`.\n *\n * The resulting `UShort` value has the same binary representation as this `Short` value.\n
 */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.toUShort(): UShort = UShort(this)\n/**\n * Converts this [Int] value to [UShort].\n *\n * If this value is positive and less than or equals to [UShort.MAX_VALUE],
 the resulting `UShort` value represents\n * the same numerical value as this `Int`.\n *\n * The resulting `UShort` value is represented by the least significant 16 bits of this `Int` value.\n
 */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Int.toUShort(): UShort = UShort(this.toShort())\n/**\n * Converts this [Long] value to [UShort].\n *\n * If this value is positive and less than or equals to [UShort.MAX_VALUE], the resulting `UShort` value represents\n * the same numerical value as this `Long`.\n *\n * The resulting `UShort` value is represented by the least significant 16 bits of this `Long` value.\n
 */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Long.toUShort(): UShort = UShort(this.toShort())\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source
```

```
 code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"CollectionsKt\")\n@file:OptIn(kotlin.exper
imental.ExperimentalTypeInference::class)\n\npackage kotlin.collections\n\nimport kotlin.contracts.*\nimport
kotlin.random.Random\n\ninternal object EmptyIterator : ListIterator<Nothing> {\n    override fun hasNext():
Boolean = false\n    override fun hasPrevious(): Boolean = false\n    override fun nextIndex(): Int = 0\n    override
fun previousIndex(): Int = -1\n    override fun next(): Nothing = throw NoSuchElementException()\n    override fun
previous(): Nothing = throw NoSuchElementException()\n}\n\ninternal object EmptyList : List<Nothing>,
Serializable, RandomAccess {\n    private const val serialVersionUID: Long = -7390468764508069838L\n\n
override fun equals(other: Any?): Boolean = other is List<*> && other.isEmpty()\n    override fun hashCode(): Int
= 1\n    override fun
 toString(): String = \"[]\"\n\n    override val size: Int get() = 0\n    override fun isEmpty(): Boolean = true\n
override fun contains(element: Nothing): Boolean = false\n    override fun containsAll(elements:
Collection<Nothing>): Boolean = elements.isEmpty()\n\n    override fun get(index: Int): Nothing = throw
IndexOutOfBoundsException(\"Empty list doesn't contain element at index $index.\")\n    override fun
indexOf(element: Nothing): Int = -1\n    override fun lastIndexOf(element: Nothing): Int = -1\n\n    override fun
iterator(): Iterator<Nothing> = EmptyIterator\n    override fun listIterator(): ListIterator<Nothing> = EmptyIterator\n
  override fun listIterator(index: Int): ListIterator<Nothing> {\n        if (index != 0) throw
IndexOutOfBoundsException(\"Index: $index\")\n        return EmptyIterator\n    }\n\n    override fun
subList(fromIndex: Int, toIndex: Int): List<Nothing> {\n        if (fromIndex == 0 && toIndex == 0) return this\n
throw IndexOutOfBoundsException(\"fromIndex:
 $fromIndex, toIndex: $toIndex\")\n    }\n\n    private fun readResolve(): Any = EmptyList\n}\n\ninternal fun <T>
Array<out T>.asCollection(): Collection<T> = ArrayAsCollection(this, isVarargs = false)\n\nprivate class
ArrayAsCollection<T>(val values: Array<out T>, val isVarargs: Boolean) : Collection<T> {\n    override val size:
Int get() = values.size\n    override fun isEmpty(): Boolean = values.isEmpty()\n    override fun contains(element:
T): Boolean = values.contains(element)\n    override fun containsAll(elements: Collection<T>): Boolean =
elements.all { contains(it) }\n    override fun iterator(): Iterator<T> = values.iterator()\n    // override hidden toArray
implementation to prevent copying of values array\n    public fun toArray(): Array<out Any?> =
values.copyToArrayOfAny(isVarargs)\n}\n\n/**\n * Returns an empty read-only list.  The returned list is
serializable (JVM).\n * @sample samples.collections.Collections.Lists.emptyReadOnlyList\n */\npublic fun
 <T> emptyList(): List<T> = EmptyList\n\n/**\n * Returns a new read-only list of given elements.  The returned list
is serializable (JVM).\n * @sample samples.collections.Collections.Lists.readOnlyList\n */\npublic fun <T>
listOf(vararg elements: T): List<T> = if (elements.size > 0) elements.asList() else emptyList()\n\n/**\n * Returns an
empty read-only list.  The returned list is serializable (JVM).\n * @sample
samples.collections.Collections.Lists.emptyReadOnlyList\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T>
listOf(): List<T> = emptyList()\n\n/**\n * Returns an empty new [MutableList].\n * @sample
samples.collections.Collections.Lists.emptyMutableList\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> mutableListOf(): MutableList<T> =
ArrayList()\n\n/**\n * Returns an empty new [ArrayList].\n * @sample
samples.collections.Collections.Lists.emptyArrayList\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> arrayListOf():
 ArrayList<T> = ArrayList()\n\n/**\n * Returns a new [MutableList] with the given elements.\n * @sample
samples.collections.Collections.Lists.mutableList\n */\npublic fun <T> mutableListOf(vararg elements: T):
MutableList<T> =\n    if (elements.size == 0) ArrayList() else ArrayList(ArrayAsCollection(elements, isVarargs =
true))\n\n/**\n * Returns a new [ArrayList] with the given elements.\n * @sample
samples.collections.Collections.Lists.arrayList\n */\npublic fun <T> arrayListOf(vararg elements: T): ArrayList<T>
=\n    if (elements.size == 0) ArrayList() else ArrayList(ArrayAsCollection(elements, isVarargs = true))\n\n/**\n *
Returns a new read-only list either of single given element, if it is not null, or empty list if the element is null. The
returned list is serializable (JVM).\n * @sample samples.collections.Collections.Lists.listOfNotNull\n */\npublic fun
```

<T : Any> listOfNotNull(element: T?): List<T> = if (element != null) listOf(element) else emptyList()\n\n/**\n * Returns
 a new read-only list only of those given elements, that are not null. The returned list is serializable (JVM).\n * @sample samples.collections.Collections.Lists.listOfNotNull\n */\npublic fun <T : Any> listOfNotNull(vararg elements: T?): List<T> = elements.filterNotNull()\n\n/**\n * Creates a new read-only list with the specified [size], where each element is calculated by calling the specified\n * [init] function.\n *\n * The function [init] is called for each list element sequentially starting from the first one.\n * It should return the value for a list element given its index.\n *\n * @sample samples.collections.Collections.Lists.readOnlyListFromInitializer\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> List(size: Int, init: (index: Int) -> T): List<T> = MutableList(size, init)\n\n/**\n * Creates a new mutable list with the specified [size], where each element is calculated by calling the specified\n * [init] function.\n *\n * The function [init]
 is called for each list element sequentially starting from the first one.\n * It should return the value for a list element given its index.\n *\n * @sample samples.collections.Collections.Lists.mutableListFromInitializer\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> MutableList(size: Int, init: (index: Int) -> T): MutableList<T> {\n   val list = ArrayList<T>(size)\n   repeat(size) { index -> list.add(init(index)) }\n return list\n}\n\n/**\n * Builds a new read-only [List] by populating a [MutableList] using the given [builderAction]\n * and returning a read-only list with the same elements.\n *\n * The list passed as a receiver to the [builderAction] is valid only inside that function.\n * Using it outside of the function produces an unspecified behavior.\n *\n * The returned list is serializable (JVM).\n *\n * @sample samples.collections.Builders.Lists.buildListSample\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Suppress(\"DEPRECATION\")\npublic
 inline fun <E> buildList(@BuilderInference builderAction: MutableList<E>.() -> Unit): List<E> {\n   contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n   return buildListInternal(builderAction)\n}\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\ninternal expect inline fun <E> buildListInternal(builderAction: MutableList<E>.() -> Unit): List<E>\n\n/**\n * Builds a new read-only [List] by populating a [MutableList] using the given [builderAction]\n * and returning a read-only list with the same elements.\n *\n * The list passed as a receiver to the [builderAction] is valid only inside that function.\n * Using it outside of the function produces an unspecified behavior.\n *\n * The returned list is serializable (JVM).\n *\n * [capacity] is used to hint the expected number of elements added in the [builderAction].\n *\n * @throws IllegalArgumentException if the given [capacity] is negative.\n *\n * @sample samples.collections.Builders.Lists.buildListSampleWithCapacity\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Suppress(\"DEPRECATION\")\npublic inline fun <E> buildList(capacity: Int, @BuilderInference builderAction: MutableList<E>.() -> Unit): List<E> {\n   contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n   return buildListInternal(capacity, builderAction)\n}\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\ninternal expect inline fun <E> buildListInternal(capacity: Int, builderAction: MutableList<E>.() -> Unit): List<E>\n\n/**\n * Returns an [IntRange] of the valid indices for this collection.\n * @sample samples.collections.Collections.Collections.indicesOfCollection\n */\npublic val Collection<*>.indices: IntRange\n get() = 0..size - 1\n\n/**\n * Returns the index of the last item in the list or -1 if the list is empty.\n *\n * @sample samples.collections.Collections.Lists.lastIndexOfList\n
 */\npublic val <T> List<T>.lastIndex: Int\n   get() = this.size - 1\n\n/**\n * Returns `true` if the collection is not empty.\n * @sample samples.collections.Collections.Collections.collectionIsNotEmpty\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Collection<T>.isNotEmpty(): Boolean = !isEmpty()\n\n/**\n * Returns `true` if this nullable collection is either null or empty.\n * @sample samples.collections.Collections.Collections.collectionIsNullOrEmpty\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Collection<T>?.isNullOrEmpty():

```
Boolean {\n    contract {\n        returns(false) implies (this@isNullOrEmpty != null)\n    }\n\n    return this == null || this.isEmpty()\n}\n\n/**\n * Returns this Collection if it's not `null` and the empty list otherwise.\n * @sample samples.collections.Collections.Collections.collectionOrEmpty\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Collection<T>?.orEmpty():
 Collection<T> = this ?: emptyList()\n\n/**\n * Returns this List if it's not `null` and the empty list otherwise.\n * @sample samples.collections.Collections.Lists.listOrEmpty\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> List<T>?.orEmpty(): List<T> = this ?: emptyList()\n\n/**\n * Returns this collection if it's not empty\n * or the result of calling [defaultValue] function if the collection is empty.\n *\n * @sample samples.collections.Collections.Collections.collectionIfEmpty\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun <C, R> C.ifEmpty(defaultValue: () -> R): R where C : Collection<*>, C : R =\n    if (isEmpty()) defaultValue() else this\n\n\n/**\n * Checks if all elements in the specified collection are contained in this collection.\n *\n * Allows to overcome type-safety restriction of `containsAll` that requires to pass a collection of type `Collection<E>`.\n * @sample samples.collections.Collections.Collections.collectionContainsAll\n */\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\") // false warning, extension takes precedence in some cases\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes T> Collection<T>.containsAll(elements: Collection<T>): Boolean = this.containsAll(elements)\n\n\n/**\n * Returns a new list with the elements of this list randomly shuffled\n * using the specified [random] instance as the source of randomness.\n */\n@SinceKotlin(\"1.3\")\npublic fun <T> Iterable<T>.shuffled(random: Random): List<T> = toMutableList().apply { shuffle(random) }\n\n\ninternal fun <T> List<T>.optimizeReadOnlyList() = when (size) {\n    0 -> emptyList()\n    1 -> listOf(this[0])\n    else -> this\n}\n\n/**\n * Searches this list or its range for the provided [element] using the binary search algorithm.\n * The list is expected to be sorted into ascending order according to the Comparable natural ordering of its elements,\n * otherwise the result is undefined.\n *\n * If the list contains multiple
 elements equal to the specified [element], there is no guarantee which one will be found.\n *\n * `null` value is considered to be less than any non-null value.\n *\n * @return the index of the element, if it is contained in the list within the specified range;\n * otherwise, the inverted insertion point `(-insertion point - 1)`.\n * The insertion point is defined as the index at which the element should be inserted,\n * so that the list (or the specified subrange of list) still remains sorted.\n * @sample samples.collections.Collections.Lists.binarySearchOnComparable\n * @sample samples.collections.Collections.Lists.binarySearchWithBoundaries\n */\npublic fun <T : Comparable<T>> List<T?>.binarySearch(element: T?, fromIndex: Int = 0, toIndex: Int = size): Int {\n    rangeCheck(size, fromIndex, toIndex)\n\n    var low = fromIndex\n    var high = toIndex - 1\n\n    while (low <= high) {\n        val mid = (low + high).ushr(1) // safe from overflows\n        val midVal = get(mid)\n
        val cmp = compareValues(midVal, element)\n\n        if (cmp < 0)\n            low = mid + 1\n        else if (cmp > 0)\n            high = mid - 1\n        else\n            return mid // key found\n    }\n    return -(low + 1)  // key not found\n}\n\n/**\n * Searches this list or its range for the provided [element] using the binary search algorithm.\n * The list is expected to be sorted into ascending order according to the specified [comparator],\n * otherwise the result is undefined.\n *\n * If the list contains multiple elements equal to the specified [element], there is no guarantee which one will be found.\n *\n * `null` value is considered to be less than any non-null value.\n *\n * @return the index of the element, if it is contained in the list within the specified range;\n * otherwise, the inverted insertion point `(-insertion point - 1)`.\n * The insertion point is defined as the index at which the element should be inserted,\n * so that the list (or the specified
 subrange of list) still remains sorted according to the specified [comparator].\n * @sample samples.collections.Collections.Lists.binarySearchWithComparator\n */\npublic fun <T> List<T>.binarySearch(element: T, comparator: Comparator<in T>, fromIndex: Int = 0, toIndex: Int = size): Int {\n    rangeCheck(size, fromIndex, toIndex)\n\n    var low = fromIndex\n    var high = toIndex - 1\n\n    while (low <= high) {\n        val mid = (low + high).ushr(1) // safe from overflows\n        val midVal = get(mid)\n        val cmp = comparator.compare(midVal, element)\n\n        if (cmp < 0)\n            low = mid + 1\n        else if (cmp > 0)\n
```

high = mid - 1\n else\n return mid // key found\n }\n return -(low + 1) // key not found\n}\n\n/**\n * Searches this list or its range for an element having the key returned by the specified [selector] function\n * equal to the provided [key] value using the binary search algorithm.\n * The list is expected to be sorted
into ascending order according to the Comparable natural ordering of keys of its elements.\n * otherwise the result is undefined.\n *\n * If the list contains multiple elements with the specified [key], there is no guarantee which one will be found.\n *\n * `null` value is considered to be less than any non-null value.\n *\n * @return the index of the element with the specified [key], if it is contained in the list within the specified range;\n * otherwise, the inverted insertion point `(-insertion point - 1)`.\n * The insertion point is defined as the index at which the element should be inserted,\n * so that the list (or the specified subrange of list) still remains sorted.\n * @sample samples.collections.Collections.Lists.binarySearchByKey\n */\npublic inline fun <T, K : Comparable<K>> List<T>.binarySearchBy(\n key: K?,\n fromIndex: Int = 0,\n toIndex: Int = size,\n crossinline selector: (T) -> K?\n): Int =\n binarySearch(fromIndex, toIndex) { compareValues(selector(it),
key) }\n\n// do not introduce this overload --- too rare\n//public fun <T, K> List<T>.binarySearchBy(key: K, comparator: Comparator<K>, fromIndex: Int = 0, toIndex: Int = size(), selector: (T) -> K): Int =\n// binarySearch(fromIndex, toIndex) { comparator.compare(selector(it), key) }\n\n\n/**\n * Searches this list or its range for an element for which the given [comparison] function returns zero using the binary search algorithm.\n *\n * The list is expected to be sorted so that the signs of the [comparison] function's return values ascend on the list elements,\n * i.e. negative values come before zero and zeroes come before positive values.\n * Otherwise, the result is undefined.\n *\n * If the list contains multiple elements for which [comparison] returns zero, there is no guarantee which one will be found.\n *\n * @param comparison function that returns zero when called on the list element being searched.\n * On the elements coming before the target element, the function
must return negative values;\n * on the elements coming after the target element, the function must return positive values.\n *\n * @return the index of the found element, if it is contained in the list within the specified range;\n * otherwise, the inverted insertion point `(-insertion point - 1)`.\n * The insertion point is defined as the index at which the element should be inserted,\n * so that the list (or the specified subrange of list) still remains sorted.\n * @sample samples.collections.Collections.Lists.binarySearchWithComparisonFunction\n */\npublic fun <T> List<T>.binarySearch(fromIndex: Int = 0, toIndex: Int = size, comparison: (T) -> Int): Int {\n rangeCheck(size, fromIndex, toIndex)\n\n var low = fromIndex\n var high = toIndex - 1\n\n while (low <= high) {\n val mid = (low + high).ushr(1) // safe from overflows\n val midVal = get(mid)\n val cmp = comparison(midVal)\n\n if (cmp < 0)\n low = mid + 1\n else if (cmp
> 0)\n high = mid - 1\n else\n return mid // key found\n }\n return -(low + 1) // key not found\n}\n\n/**\n * Checks that `from` and `to` are in\n * the range of [0..size] and throws an appropriate exception, if they aren't.\n */\nprivate fun rangeCheck(size: Int, fromIndex: Int, toIndex: Int) {\n when {\n fromIndex > toIndex -> throw IllegalArgumentException(\"fromIndex ($fromIndex) is greater than toIndex ($toIndex).\")\n fromIndex < 0 -> throw IndexOutOfBoundsException(\"fromIndex ($fromIndex) is less than zero.\")\n toIndex > size -> throw IndexOutOfBoundsException(\"toIndex ($toIndex) is greater than size ($size).\")\n }\n}\n\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\ninternal expect fun checkIndexOverflow(index: Int): Int\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\ninternal expect fun checkCountOverflow(count: Int): Int\n\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\ninternal fun throwIndexOverflow() { throw ArithmeticException(\"Index
overflow has happened.\") }\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\ninternal fun throwCountOverflow() { throw ArithmeticException(\"Count overflow has happened.\") }\n\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"MapsKt\")\n@file:OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n\npackage kotlin.collections\n\nimport kotlin.contracts.*\n\nprivate object EmptyMap : Map<Any?, Nothing>, Serializable {\n private const val serialVersionUID: Long = 8246714829545688274\n\n override fun equals(other: Any?): Boolean = other is Map<*, *> &&

other.isEmpty()\n   override fun hashCode(): Int = 0\n   override fun toString(): String = \"{}\"\n\n   override val size: Int get() = 0\n   override fun isEmpty(): Boolean = true\n\n   override fun containsKey(key: Any?): Boolean = false\n   override fun containsValue(value: Nothing): Boolean = false\n override fun get(key: Any?): Nothing? = null\n   override val entries: Set<Map.Entry<Any?, Nothing>> get() = EmptySet\n   override val keys: Set<Any?> get() = EmptySet\n   override val values: Collection<Nothing> get() = EmptyList\n\n   private fun readResolve(): Any = EmptyMap\n}\n\n/**\n * Returns an empty read-only map of specified type.\n *\n * The returned map is serializable (JVM).\n * @sample samples.collections.Maps.Instantiation.emptyReadOnlyMap\n */\npublic fun <K, V> emptyMap(): Map<K, V> = @Suppress(\"UNCHECKED_CAST\") (EmptyMap as Map<K, V>)\n\n/**\n * Returns a new read-only map with the specified contents, given as a list of pairs\n * where the first value is the key and the second is the value.\n *\n * If multiple pairs have the same key, the resulting map will contain the value from the last of those pairs.\n *\n * Entries of the map are iterated in the order they were specified.\n *\n * The returned map is serializable (JVM).\n *\n * @sample samples.collections.Maps.Instantiation.mapFromPairs\n */\npublic fun <K, V> mapOf(vararg pairs: Pair<K, V>): Map<K, V> =\n   if (pairs.size > 0) pairs.toMap(LinkedHashMap(mapCapacity(pairs.size))) else emptyMap()\n\n/**\n * Returns an empty read-only map.\n *\n * The returned map is serializable (JVM).\n * @sample samples.collections.Maps.Instantiation.emptyReadOnlyMap\n */\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> mapOf(): Map<K, V> = emptyMap()\n\n/**\n * Returns an empty new [MutableMap].\n *\n * The returned map preserves the entry iteration order.\n * @sample samples.collections.Maps.Instantiation.emptyMutableMap\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> mutableMapOf(): MutableMap<K, V> = LinkedHashMap()\n\n/**\n * Returns a new [MutableMap] with the specified contents, given as a list of pairs\n * where the first component is the key and the second is the value.\n *\n * If multiple pairs have the same key, the resulting map will contain the value from the last of those pairs.\n *\n * Entries of the map are iterated in the order they were specified.\n *\n * @sample samples.collections.Maps.Instantiation.mutableMapFromPairs\n * @sample samples.collections.Maps.Instantiation.emptyMutableMap\n */\npublic fun <K, V> mutableMapOf(vararg pairs: Pair<K, V>): MutableMap<K, V> =\n   LinkedHashMap<K, V>(mapCapacity(pairs.size)).apply { putAll(pairs) }\n\n/**\n * Returns an empty new [HashMap].\n *\n * @sample samples.collections.Maps.Instantiation.emptyHashMap\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> hashMapOf(): HashMap<K, V> = HashMap<K, V>()\n\n/**\n * Returns a new [HashMap] with the specified contents, given as a list of pairs\n * where the first component is the key and the second is the value.\n *\n * @sample samples.collections.Maps.Instantiation.hashMapFromPairs\n */\npublic fun <K, V> hashMapOf(vararg pairs: Pair<K, V>): HashMap<K, V> = HashMap<K, V>(mapCapacity(pairs.size)).apply { putAll(pairs) }\n\n/**\n * Returns an empty new [LinkedHashMap].\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> linkedMapOf(): LinkedHashMap<K, V> = LinkedHashMap<K, V>()\n\n/**\n * Returns a new [LinkedHashMap] with the specified contents, given as a list of pairs\n * where the first component is the key and the second is the value.\n *\n * If multiple pairs have the same key, the resulting map will contain the value from the last of those pairs.\n *\n * Entries of the map are iterated in the order they were specified.\n *\n * @sample samples.collections.Maps.Instantiation.linkedMapFromPairs\n */\npublic fun <K, V> linkedMapOf(vararg pairs: Pair<K, V>): LinkedHashMap<K, V> = pairs.toMap(LinkedHashMap(mapCapacity(pairs.size)))\n\n/**\n * Builds a new read-only [Map] by populating a [MutableMap] using the given [builderAction]\n * and returning a read-only map with the same key-value pairs.\n *\n * The map passed as a receiver to the [builderAction] is valid only inside that function.\n * Using it outside of the function produces an unspecified behavior.\n *\n * Entries of the map are iterated in the order they were added by the [builderAction].\n *\n * The returned map is serializable (JVM).\n *\n * @sample samples.collections.Builders.Maps.buildMapSample\n

*/\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Suppress(\"DEPRECATION\")\npublic inline fun <K, V> buildMap(@BuilderInference builderAction: MutableMap<K, V>.() -> Unit): Map<K, V> {\n   contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n   return buildMapInternal(builderAction)\n}\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\ninternal expect inline fun <K, V> buildMapInternal(builderAction: MutableMap<K, V>.() -> Unit): Map<K, V>\n\n/**\n * Builds a new read-only
 [Map] by populating a [MutableMap] using the given [builderAction]\n * and returning a read-only map with the same key-value pairs.\n *\n * The map passed as a receiver to the [builderAction] is valid only inside that function.\n * Using it outside of the function produces an unspecified behavior.\n *\n * [capacity] is used to hint the expected number of pairs added in the [builderAction].\n *\n * Entries of the map are iterated in the order they were added by the [builderAction].\n *\n * The returned map is serializable (JVM).\n *\n * @throws IllegalArgumentException if the given [capacity] is negative.\n *\n * @sample samples.collections.Builders.Maps.buildMapSample\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Suppress(\"DEPRECATION\")\npublic inline fun <K, V> buildMap(capacity: Int, @BuilderInference builderAction: MutableMap<K, V>.() -> Unit): Map<K, V> {\n   contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE)
 }\n   return buildMapInternal(capacity, builderAction)\n}\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\ninternal expect inline fun <K, V> buildMapInternal(capacity: Int, builderAction: MutableMap<K, V>.() -> Unit): Map<K, V>\n\n/**\n * Calculate the initial capacity of a map.\n */\n@PublishedApi\ninternal expect fun mapCapacity(expectedSize: Int): Int\n\n/**\n * Returns `true` if this map is not empty.\n * @sample samples.collections.Maps.Usage.mapIsNotEmpty\n */\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.isNotEmpty(): Boolean = !isEmpty()\n\n/**\n * Returns `true` if this nullable map is either null or empty.\n * @sample samples.collections.Maps.Usage.mapIsNullOrEmpty\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>?.isNullOrEmpty(): Boolean {\n   contract {\n   returns(false) implies (this@isNullOrEmpty != null)\n   }\n\n   return this == null || isEmpty()\n}\n\n/**\n *
 Returns the [Map] if its not `null`, or the empty [Map] otherwise.\n *\n * @sample samples.collections.Maps.Usage.mapOrEmpty\n */\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<K, V>?.orEmpty(): Map<K, V> = this ?: emptyMap()\n\n/**\n * Returns this map if it's not empty\n * or the result of calling [defaultValue] function if the map is empty.\n *\n * @sample samples.collections.Maps.Usage.mapIfEmpty\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun <M, R> M.ifEmpty(defaultValue: () -> R): R where M : Map<*, *>, M : R =\n   if (isEmpty()) defaultValue() else this\n\n/**\n * Checks if the map contains the given key.\n *\n * This method allows to use the `x in map` syntax for checking whether an object is contained in the map.\n *\n * @sample samples.collections.Maps.Usage.containsKey\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <@kotlin.internal.OnlyInputTypes K, V> Map<out K, V>.contains(key: K): Boolean = containsKey(key)\n\n/**\n
 * Returns the value corresponding to the given [key], or `null` if such a key is not present in the map.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <@kotlin.internal.OnlyInputTypes K, V> Map<out K, V>.get(key: K): V? =\n   @Suppress(\"UNCHECKED_CAST\") (this as Map<K, V>).get(key)\n\n/**\n * Allows to use the index operator for storing values in a mutable map.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<K, V>.set(key: K, value: V): Unit {\n   put(key, value)\n}\n\n/**\n * Returns `true` if the map contains the specified [key].\n *\n * Allows to overcome type-safety restriction of `containsKey` that requires to pass a key of type `K`.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes K> Map<out K, *>.containsKey(key: K): Boolean =\n@Suppress(\"UNCHECKED_CAST\") (this as Map<K, *>).containsKey(key)\n\n/**\n * Returns `true` if the map

maps one or more keys to the specified [value].\n *\n *
Allows to overcome type-safety restriction of `containsValue` that requires to pass a value of type `V`.\n *\n * @sample samples.collections.Maps.Usage.containsValue\n */\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\") // false warning, extension takes precedence in some cases\n@kotlin.internal.InlineOnly\npublic inline fun <K, @kotlin.internal.OnlyInputTypes V> Map<K, V>.containsValue(value: V): Boolean = this.containsValue(value)\n\n\n/**\n * Removes the specified key and its corresponding value from this map.\n *\n * @return the previous value associated with the key, or `null` if the key was not present in the map.\n\n * Allows to overcome type-safety restriction of `remove` that requires to pass a key of type `K`.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes K, V> MutableMap<out K, V>.remove(key: K): V? =\n @Suppress(\"UNCHECKED_CAST\") (this as MutableMap<K, V>).remove(key)\n\n/**\n * Returns the key component of the map entry.\n *\n * This
method allows to use destructuring declarations when working with maps, for example:\n * ```\n * for ((key, value) in map) {\n * // do something with the key and the value\n * }\n * ```\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> Map.Entry<K, V>.component1(): K = key\n\n/**\n * Returns the value component of the map entry.\n *\n * This method allows to use destructuring declarations when working with maps, for example:\n * ```\n * for ((key, value) in map) {\n * // do something with the key and the value\n * }\n * ```\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> Map.Entry<K, V>.component2(): V = value\n\n/**\n * Converts entry to [Pair] with key being first component and value being second.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map.Entry<K, V>.toPair(): Pair<K, V> = Pair(key, value)\n\n/**\n * Returns the value for the given [key] if the value is present and not `null`.\n * Otherwise, returns the result of
the [defaultValue] function.\n *\n * @sample samples.collections.Maps.Usage.getOrElse\n */\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<K, V>.getOrElse(key: K, defaultValue: () -> V): V = get(key) ?: defaultValue()\n\n\ninternal inline fun <K, V> Map<K, V>.getOrElseNullable(key: K, defaultValue: () -> V): V {\n val value = get(key)\n if (value == null && !containsKey(key)) {\n return defaultValue()\n } else {\n @Suppress(\"UNCHECKED_CAST\")\n return value as V\n }\n}\n\n/**\n * Returns the value for the given [key] or throws an exception if there is no such key in the map.\n *\n * If the map was created by [withDefault], resorts to its `defaultValue` provider function\n * instead of throwing an exception.\n *\n * @throws NoSuchElementException when the map doesn't contain a value for the specified key and\n * no implicit default value was provided for that map.\n */\n@SinceKotlin(\"1.1\")\npublic fun <K, V> Map<K, V>.getValue(key: K): V = getOrImplicitDefault(key)\n\n/**\n * Returns the value for the given [key] if the value is present and not `null`.\n * Otherwise, calls the [defaultValue] function,\n * puts its result into the map under the given key and returns the call result.\n *\n * Note that the operation is not guaranteed to be atomic if the map is being modified concurrently.\n *\n * @sample samples.collections.Maps.Usage.getOrPut\n */\npublic inline fun <K, V> MutableMap<K, V>.getOrPut(key: K, defaultValue: () -> V): V {\n val value = get(key)\n return if (value == null) {\n val answer = defaultValue()\n put(key, answer)\n answer\n } else {\n value\n }\n}\n\n/**\n * Returns an [Iterator] over the entries in the [Map].\n *\n * @sample samples.collections.Maps.Usage.forOverEntries\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> Map<out K, V>.iterator(): Iterator<Map.Entry<K, V>> = entries.iterator()\n\n/**\n * Returns a [MutableIterator] over the mutable
entries in the [MutableMap].\n *\n */\n@kotlin.jvm.JvmName(\"mutableIterator\")\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<K, V>.iterator(): MutableIterator<MutableMap.MutableEntry<K, V>> = entries.iterator()\n\n/**\n * Populates the given [destination] map with entries having the keys of this map and the values obtained\n * by applying the [transform] function to each entry in this [Map].\n */\npublic inline fun <K, V, R, M : MutableMap<in K, in R>> Map<out K, V>.mapValuesTo(destination: M, transform: (Map.Entry<K, V>) -> R): M {\n return entries.associateByTo(destination, { it.key }, transform)\n}\n\n/**\n * Populates the given [destination] map with entries having the keys obtained\n * by applying the [transform] function to each entry in this [Map] and the values

of this map.\n *\n * In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite\n * the value associated with the former one.\n */\npublic inline fun <K, V, R, M : MutableMap<in R, in V>> Map<out K, V>.mapKeysTo(destination: M, transform: (Map.Entry<K, V>) -> R): M {\n return entries.associateByTo(destination, transform, { it.value })\n}\n\n/**\n * Puts all the given [pairs] into this [MutableMap] with the first component in the pair being the key and the second the value.\n */\npublic fun <K, V> MutableMap<in K, in V>.putAll(pairs: Array<out Pair<K, V>>): Unit {\n for ((key, value) in pairs) {\n put(key, value)\n }\n}\n\n/**\n * Puts all the elements of the given collection into this [MutableMap] with the first component in the pair being the key and the second the value.\n */\npublic fun <K, V> MutableMap<in K, in V>.putAll(pairs: Iterable<Pair<K, V>>): Unit {\n for ((key, value) in pairs) {\n put(key, value)\n }\n}\n\n/**\n * Puts all the elements of the given sequence into this [MutableMap] with the first component in the pair being the key and the second the value.\n */\npublic fun <K, V> MutableMap<in K, in V>.putAll(pairs: Sequence<Pair<K, V>>): Unit {\n for ((key, value) in pairs) {\n put(key, value)\n }\n}\n\n/**\n * Returns a new map with entries having the keys of this map and the values obtained by applying the [transform]\n * function to each entry in this [Map].\n *\n * The returned map preserves the entry iteration order of the original map.\n *\n * @sample samples.collections.Maps.Transformations.mapValues\n */\npublic inline fun <K, V, R> Map<out K, V>.mapValues(transform: (Map.Entry<K, V>) -> R): Map<K, R> {\n return mapValuesTo(LinkedHashMap<K, R>(mapCapacity(size)), transform) // .optimizeReadOnlyMap()\n}\n\n/**\n * Returns a new Map with entries having the keys obtained by applying the [transform] function to each entry in this\n * [Map] and the values of this map.\n *\n * In case if any two entries are mapped to the equal keys, the value of the latter one will overwrite\n * the value associated with the former one.\n *\n * The returned map preserves the entry iteration order of the original map.\n *\n * @sample samples.collections.Maps.Transformations.mapKeys\n */\npublic inline fun <K, V, R> Map<out K, V>.mapKeys(transform: (Map.Entry<K, V>) -> R): Map<R, V> {\n return mapKeysTo(LinkedHashMap<R, V>(mapCapacity(size)), transform) // .optimizeReadOnlyMap()\n}\n\n/**\n * Returns a map containing all key-value pairs with keys matching the given [predicate].\n *\n * The returned map preserves the entry iteration order of the original map.\n * @sample samples.collections.Maps.Filtering.filterKeys\n */\npublic inline fun <K, V> Map<out K, V>.filterKeys(predicate: (K) -> Boolean): Map<K, V> {\n val result = LinkedHashMap<K, V>()\n for (entry in this) {\n if (predicate(entry.key)) {\n result.put(entry.key, entry.value)\n }\n }\n return result\n}\n\n/**\n * Returns a map containing all key-value pairs with values matching the given [predicate].\n *\n * The returned map preserves the entry iteration order of the original map.\n * @sample samples.collections.Maps.Filtering.filterValues\n */\npublic inline fun <K, V> Map<out K, V>.filterValues(predicate: (V) -> Boolean): Map<K, V> {\n val result = LinkedHashMap<K, V>()\n for (entry in this) {\n if (predicate(entry.value)) {\n result.put(entry.key, entry.value)\n }\n }\n return result\n}\n\n/**\n * Appends all entries matching the given [predicate] into the mutable map given as [destination] parameter.\n *\n * @return the destination map.\n * @sample samples.collections.Maps.Filtering.filterTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> Map<out K, V>.filterTo(destination: M, predicate: (Map.Entry<K, V>) -> Boolean): M {\n for (element in this) {\n if (predicate(element)) {\n destination.put(element.key, element.value)\n }\n }\n return destination\n}\n\n/**\n * Returns a new map containing all key-value pairs matching the given [predicate].\n *\n * The returned map preserves the entry iteration order of the original map.\n * @sample samples.collections.Maps.Filtering.filter\n */\npublic inline fun <K, V> Map<out K, V>.filter(predicate: (Map.Entry<K, V>) -> Boolean): Map<K, V> {\n return filterTo(LinkedHashMap<K, V>(), predicate)\n}\n\n/**\n * Appends all entries not matching the given [predicate] into the given [destination].\n *\n * @return the destination map.\n * @sample samples.collections.Maps.Filtering.filterNotTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>> Map<out K, V>.filterNotTo(destination: M, predicate: (Map.Entry<K, V>) -> Boolean): M {\n for (element in this) {\n if (!predicate(element)) {\n destination.put(element.key, element.value)\n }\n }\n return destination\n}\n\n/**\n * Returns a new map containing all key-value pairs not matching the given [predicate].\n *\n * The returned map preserves the entry iteration order of the original

map.\n * @sample
samples.collections.Maps.Filtering.filterNot\n */\npublic inline fun <K, V> Map<out K, V>.filterNot(predicate: (Map.Entry<K, V>) -> Boolean): Map<K, V> {\n return filterNotTo(LinkedHashMap<K, V>(), predicate)\n}\n\n/**\n * Returns a new map containing all key-value pairs from the given collection of pairs.\n *\n * The returned map preserves the entry iteration order of the original collection.\n * If any of two pairs would have the same key the last one gets added to the map.\n */\npublic fun <K, V> Iterable<Pair<K, V>>.toMap(): Map<K, V> {\n if (this is Collection) {\n return when (size) {\n 0 -> emptyMap()\n 1 -> mapOf(if (this is List) this[0] else iterator().next())\n else -> toMap(LinkedHashMap<K, V>(mapCapacity(size)))\n }\n }\n return toMap(LinkedHashMap<K, V>()).optimizeReadOnlyMap()\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs from the given collection of pairs.\n */\npublic fun
<K, V, M : MutableMap<in K, in V>> Iterable<Pair<K, V>>.toMap(destination: M): M =\n destination.apply { putAll(this@toMap) }\n\n/**\n * Returns a new map containing all key-value pairs from the given array of pairs.\n *\n * The returned map preserves the entry iteration order of the original array.\n * If any of two pairs would have the same key the last one gets added to the map.\n */\npublic fun <K, V> Array<out Pair<K, V>>.toMap(): Map<K, V> = when (size) {\n 0 -> emptyMap()\n 1 -> mapOf(this[0])\n else -> toMap(LinkedHashMap<K, V>(mapCapacity(size)))\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs from the given array of pairs.\n */\npublic fun <K, V, M : MutableMap<in K, in V>> Array<out Pair<K, V>>.toMap(destination: M): M =\n destination.apply { putAll(this@toMap) }\n\n/**\n * Returns a new map containing all key-value pairs from the given sequence of pairs.\n *\n * The returned map preserves the entry iteration order of
the original sequence.\n * If any of two pairs would have the same key the last one gets added to the map.\n */\npublic fun <K, V> Sequence<Pair<K, V>>.toMap(): Map<K, V> = toMap(LinkedHashMap<K, V>()).optimizeReadOnlyMap()\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs from the given sequence of pairs.\n */\npublic fun <K, V, M : MutableMap<in K, in V>> Sequence<Pair<K, V>>.toMap(destination: M): M =\n destination.apply { putAll(this@toMap) }\n\n/**\n * Returns a new read-only map containing all key-value pairs from the original map.\n *\n * The returned map preserves the entry iteration order of the original map.\n */\n@SinceKotlin(\"1.1\")\npublic fun <K, V> Map<out K, V>.toMap(): Map<K, V> = when (size) {\n 0 -> emptyMap()\n 1 -> toSingletonMap()\n else -> toMutableMap()\n}\n\n/**\n * Returns a new mutable map containing all key-value pairs from the original map.\n *\n * The returned map preserves the entry iteration order of the original
map.\n */\n@SinceKotlin(\"1.1\")\npublic fun <K, V> Map<out K, V>.toMutableMap(): MutableMap<K, V> = LinkedHashMap(this)\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs from the given map.\n */\n@SinceKotlin(\"1.1\")\npublic fun <K, V, M : MutableMap<in K, in V>> Map<out K, V>.toMap(destination: M): M =\n destination.apply { putAll(this@toMap) }\n\n/**\n * Creates a new read-only map by replacing or adding an entry to this map from a given key-value [pair].\n *\n * The returned map preserves the entry iteration order of the original map.\n * The [pair] is iterated in the end if it has a unique key.\n */\npublic operator fun <K, V> Map<out K, V>.plus(pair: Pair<K, V>): Map<K, V> =\n if (this.isEmpty()) mapOf(pair) else LinkedHashMap(this).apply { put(pair.first, pair.second) }\n\n/**\n * Creates a new read-only map by replacing or adding entries to this map from a given collection of key-value [pairs].\n *\n * The returned map preserves the
entry iteration order of the original map.\n * Those [pairs] with unique keys are iterated in the end in the order of [pairs] collection.\n */\npublic operator fun <K, V> Map<out K, V>.plus(pairs: Iterable<Pair<K, V>>): Map<K, V> =\n if (this.isEmpty()) pairs.toMap() else LinkedHashMap(this).apply { putAll(pairs) }\n\n/**\n * Creates a new read-only map by replacing or adding entries to this map from a given array of key-value [pairs].\n *\n * The returned map preserves the entry iteration order of the original map.\n * Those [pairs] with unique keys are iterated in the end in the order of [pairs] array.\n */\npublic operator fun <K, V> Map<out K, V>.plus(pairs: Array<out Pair<K, V>>): Map<K, V> =\n if (this.isEmpty()) pairs.toMap() else LinkedHashMap(this).apply { putAll(pairs) }\n\n/**\n * Creates a new read-only map by replacing or adding entries to this map from a given sequence of key-value [pairs].\n *\n * The returned map preserves the entry iteration order of the original

map.\n * Those [pairs] with unique keys are iterated in the end in the order of [pairs] sequence.\n */\npublic operator fun <K, V> Map<out K, V>.plus(pairs: Sequence<Pair<K, V>>): Map<K, V> =\n LinkedHashMap(this).apply { putAll(pairs) }.optimizeReadOnlyMap()\n\n/**\n * Creates a new read-only map by replacing or adding entries to this map from another [map].\n *\n * The returned map preserves the entry iteration order of the original map.\n * Those entries of another [map] that are missing in this map are iterated in the end in the order of that [map].\n */\npublic operator fun <K, V> Map<out K, V>.plus(map: Map<out K, V>): Map<K, V> =\n LinkedHashMap(this).apply { putAll(map) }\n\n\n/**\n * Appends or replaces the given [pair] in this mutable map.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<in K, in V>.plusAssign(pair: Pair<K, V>) {\n put(pair.first, pair.second)\n}\n\n/**\n * Appends or replaces all pairs from the given collection of
[pairs] in this mutable map.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<in K, in V>.plusAssign(pairs: Iterable<Pair<K, V>>) {\n putAll(pairs)\n}\n\n/**\n * Appends or replaces all pairs from the given array of [pairs] in this mutable map.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<in K, in V>.plusAssign(pairs: Array<out Pair<K, V>>) {\n putAll(pairs)\n}\n\n/**\n * Appends or replaces all pairs from the given sequence of [pairs] in this mutable map.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<in K, in V>.plusAssign(pairs: Sequence<Pair<K, V>>) {\n putAll(pairs)\n}\n\n/**\n * Appends or replaces all entries from the given [map] in this mutable map.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<in K, in V>.plusAssign(map: Map<K, V>) {\n putAll(map)\n}\n\n/**\n * Returns a map containing all entries of the original map except the entry
with the given [key].\n *\n * The returned map preserves the entry iteration order of the original map.\n */\n@SinceKotlin(\"1.1\")\npublic operator fun <K, V> Map<out K, V>.minus(key: K): Map<K, V> =\n this.toMutableMap().apply { minusAssign(key) }.optimizeReadOnlyMap()\n\n/**\n * Returns a map containing all entries of the original map except those entries\n * the keys of which are contained in the given [keys] collection.\n *\n * The returned map preserves the entry iteration order of the original map.\n */\n@SinceKotlin(\"1.1\")\npublic operator fun <K, V> Map<out K, V>.minus(keys: Iterable<K>): Map<K, V> =\n this.toMutableMap().apply { minusAssign(keys) }.optimizeReadOnlyMap()\n\n/**\n * Returns a map containing all entries of the original map except those entries\n * the keys of which are contained in the given [keys] array.\n *\n * The returned map preserves the entry iteration order of the original map.\n */\n@SinceKotlin(\"1.1\")\npublic operator fun <K, V> Map<out K, V>.minus(keys:
Array<out K>): Map<K, V> =\n this.toMutableMap().apply { minusAssign(keys) }.optimizeReadOnlyMap()\n\n/**\n * Returns a map containing all entries of the original map except those entries\n * the keys of which are contained in the given [keys] sequence.\n *\n * The returned map preserves the entry iteration order of the original map.\n */\n@SinceKotlin(\"1.1\")\npublic operator fun <K, V> Map<out K, V>.minus(keys: Sequence<K>): Map<K, V> =\n this.toMutableMap().apply { minusAssign(keys) }.optimizeReadOnlyMap()\n\n/**\n * Removes the entry with the given [key] from this mutable map.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<K, V>.minusAssign(key: K) {\n remove(key)\n}\n\n/**\n * Removes all entries the keys of which are contained in the given [keys] collection from this mutable map.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<K, V>.minusAssign(keys:
Iterable<K>) {\n this.keys.removeAll(keys)\n}\n\n/**\n * Removes all entries the keys of which are contained in the given [keys] array from this mutable map.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<K, V>.minusAssign(keys: Array<out K>) {\n this.keys.removeAll(keys)\n}\n\n/**\n * Removes all entries from the keys of which are contained in the given [keys] sequence from this mutable map.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline operator fun <K, V> MutableMap<K, V>.minusAssign(keys: Sequence<K>) {\n this.keys.removeAll(keys)\n}\n\n\n// do not expose for now @PublishedApi\ninternal fun <K, V> Map<K,

V>.optimizeReadOnlyMap() = when (size) {\n   0 -> emptyMap()\n   1 -> toSingletonMapOrSelf()\n   else -> this\n}\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found
 in the license/LICENSE.txt file.\n
/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"SetsKt\")\n@file:OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n\npackage kotlin.collections\n\nimport kotlin.contracts.\n\ninternal object EmptySet : Set<Nothing>, Serializable {\n   private const val serialVersionUID: Long = 3406603774387020532\n\n   override fun equals(other: Any?): Boolean = other is Set<*> && other.isEmpty()\n override fun hashCode(): Int = 0\n   override fun toString(): String = \"[]\"\n\n   override val size: Int get() = 0\n override fun isEmpty(): Boolean = true\n   override fun contains(element: Nothing): Boolean = false\n   override fun containsAll(elements: Collection<Nothing>): Boolean = elements.isEmpty()\n\n   override fun iterator(): Iterator<Nothing> = EmptyIterator\n\n   private fun readResolve(): Any = EmptySet\n}\n\n\n/**\n * Returns an empty read-only set. The returned set is serializable (JVM).\n * @sample samples.collections.Collections.Sets.emptyReadOnlySet\n
 */\npublic fun <T> emptySet(): Set<T> = EmptySet\n\n/**\n * Returns a new read-only set with the given elements.\n * Elements of the set are iterated in the order they were specified.\n * The returned set is serializable (JVM).\n * @sample samples.collections.Collections.Sets.readOnlySet\n */\npublic fun <T> setOf(vararg elements: T): Set<T> = if (elements.size > 0) elements.toSet() else emptySet()\n\n/**\n * Returns an empty read-only set. The returned set is serializable (JVM).\n * @sample samples.collections.Collections.Sets.emptyReadOnlySet\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> setOf(): Set<T> = emptySet()\n\n/**\n * Returns an empty new [MutableSet].\n *\n * The returned set preserves the element iteration order.\n * @sample samples.collections.Collections.Sets.emptyMutableSet\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> mutableSetOf(): MutableSet<T> = LinkedHashSet()\n\n/**\n *
 Returns a new [MutableSet] with the given elements.\n * Elements of the set are iterated in the order they were specified.\n * @sample samples.collections.Collections.Sets.mutableSet\n */\npublic fun <T> mutableSetOf(vararg elements: T): MutableSet<T> = elements.toCollection(LinkedHashSet(mapCapacity(elements.size)))\n\n/** Returns an empty new [HashSet]. */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> hashSetOf(): HashSet<T> = HashSet()\n\n/** Returns a new [HashSet] with the given elements. */\npublic fun <T> hashSetOf(vararg elements: T): HashSet<T> = elements.toCollection(HashSet(mapCapacity(elements.size)))\n\n/**\n * Returns an empty new [LinkedHashSet].\n * @sample samples.collections.Collections.Sets.emptyLinkedHashSet\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> linkedSetOf(): LinkedHashSet<T> = LinkedHashSet()\n\n/**\n * Returns a new [LinkedHashSet] with the given elements.\n * Elements of the set are iterated
 in the order they were specified.\n * @sample samples.collections.Collections.Sets.linkedHashSet\n */\npublic fun <T> linkedSetOf(vararg elements: T): LinkedHashSet<T> = elements.toCollection(LinkedHashSet(mapCapacity(elements.size)))\n\n/**\n * Returns a new read-only set either with single given element, if it is not null, or empty set if the element is null.\n * The returned set is serializable (JVM).\n * @sample samples.collections.Collections.Sets.setOfNotNull\n */\n@SinceKotlin(\"1.4\")\npublic fun <T : Any> setOfNotNull(element: T?): Set<T> = if (element != null) setOf(element) else emptySet()\n\n/**\n * Returns a new read-only set only with those given elements, that are not null.\n * Elements of the set are iterated in the order they were specified.\n * The returned set is serializable (JVM).\n * @sample samples.collections.Collections.Sets.setOfNotNull\n */\n@SinceKotlin(\"1.4\")\npublic fun <T : Any> setOfNotNull(vararg elements: T?): Set<T> {\n   return elements.filterNotNullTo(LinkedHashSet())\n}\n\n/**\n
 * Builds a new read-only [Set] by populating a [MutableSet] using the given [builderAction]\n * and returning a read-only set with the same elements.\n *\n * The set passed as a receiver to the [builderAction] is valid only inside that function.\n * Using it outside of the function produces an unspecified behavior.\n *\n * Elements of the set are

```
iterated in the order they were added by the [builderAction].\n *\n * The returned set is serializable (JVM).\n *\n * @sample samples.collections.Builders.Sets.buildSetSample\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Suppress(\"DEPRECATION\")\npublic inline fun <E> buildSet(@BuilderInference builderAction: MutableSet<E>.() -> Unit): Set<E> {\n    contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n    return buildSetInternal(builderAction)\n}\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\ninternal expect
 inline fun <E> buildSetInternal(builderAction: MutableSet<E>.() -> Unit): Set<E>\n\n/**\n * Builds a new read-only [Set] by populating a [MutableSet] using the given [builderAction]\n * and returning a read-only set with the same elements.\n *\n * The set passed as a receiver to the [builderAction] is valid only inside that function.\n * Using it outside of the function produces an unspecified behavior.\n *\n * [capacity] is used to hint the expected number of elements added in the [builderAction].\n *\n * Elements of the set are iterated in the order they were added by the [builderAction].\n *\n * The returned set is serializable (JVM).\n *\n * @throws IllegalArgumentException if the given [capacity] is negative.\n *\n * @sample samples.collections.Builders.Sets.buildSetSample\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Suppress(\"DEPRECATION\")\npublic inline fun <E> buildSet(capacity: Int, @BuilderInference builderAction:
 MutableSet<E>.() -> Unit): Set<E> {\n    contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n    return buildSetInternal(capacity, builderAction)\n}\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\ninternal expect inline fun <E> buildSetInternal(capacity: Int, builderAction: MutableSet<E>.() -> Unit): Set<E>\n\n\n/** Returns this Set if it's not `null` and the empty set otherwise. */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Set<T>?.orEmpty(): Set<T> = this ?: emptySet()\n\ninternal fun <T> Set<T>.optimizeReadOnlySet() = when (size) {\n    0 -> emptySet()\n    1 -> setOf(iterator().next())\n    else -> this\n}\n","/*\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin.ranges\n\n/**\n * A range of values of type `Char`.\n
 */\n@OptIn(ExperimentalStdlibApi::class)\npublic class CharRange(start: Char, endInclusive: Char) : CharProgression(start, endInclusive, 1), ClosedRange<Char>, OpenEndRange<Char> {\n    override val start: Char get() = first\n    override val endInclusive: Char get() = last\n    \n    @Deprecated(\"Can throw an exception when it's impossible to represent the value with Char type, for example, when the range includes MAX_VALUE. It's recommended to use 'endInclusive' property that doesn't throw.\")\n    @SinceKotlin(\"1.7\")\n    @ExperimentalStdlibApi\n    override val endExclusive: Char get() {\n        if (last == Char.MAX_VALUE) error(\"Cannot return the exclusive upper bound of a range that includes MAX_VALUE.\")\n        return last + 1\n    }\n\n    override fun contains(value: Char): Boolean = first <= value && value <= last\n\n    /** \n     * Checks whether the range is empty.\n     *\n     * The range is empty if its start value is greater than the end value.\n     */\n    override
 fun isEmpty(): Boolean = first > last\n\n    override fun equals(other: Any?): Boolean =\n        other is CharRange && (isEmpty() && other.isEmpty() ||\n        first == other.first && last == other.last)\n\n    override fun hashCode(): Int =\n        if (isEmpty()) -1 else (31 * first.code + last.code)\n\n    override fun toString(): String = \"$first..$last\"\n\n    companion object {\n        /** An empty range of values of type Char. */\n        public val EMPTY: CharRange = CharRange(1.toChar(), 0.toChar())\n    }\n}\n\n/**\n * A range of values of type `Int`.\n */\n@OptIn(ExperimentalStdlibApi::class)\npublic class IntRange(start: Int, endInclusive: Int) : IntProgression(start, endInclusive, 1), ClosedRange<Int>, OpenEndRange<Int> {\n    override val start: Int get() = first\n    override val endInclusive: Int get() = last\n    \n    @Deprecated(\"Can throw an exception when it's impossible to represent the value with Int type, for example, when the range includes MAX_VALUE.
 It's recommended to use 'endInclusive' property that doesn't throw.\")\n    @SinceKotlin(\"1.7\")\n    @ExperimentalStdlibApi\n    override val endExclusive: Int get() {\n        if (last == Int.MAX_VALUE)
```

error(\"Cannot return the exclusive upper bound of a range that includes MAX_VALUE.\")\n return last + 1\n }\n\n override fun contains(value: Int): Boolean = first <= value && value <= last\n\n /** \n * Checks whether the range is empty.\n *\n * The range is empty if its start value is greater than the end value.\n */\n override fun isEmpty(): Boolean = first > last\n\n override fun equals(other: Any?): Boolean =\n other is IntRange && (isEmpty() && other.isEmpty() ||\n first == other.first && last == other.last)\n\n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 * first + last)\n\n override fun toString(): String = \"$first..$last\"\n\n companion object {\n /** An empty range of values of type Int. */\n public val EMPTY: IntRange = IntRange(1, 0)\n }\n}\n\n/**\n * A range of values of type `Long`.\n */\n@OptIn(ExperimentalStdlibApi::class)\npublic class LongRange(start: Long, endInclusive: Long) : LongProgression(start, endInclusive, 1), ClosedRange<Long>, OpenEndRange<Long> {\n override val start: Long get() = first\n override val endInclusive: Long get() = last\n \n @Deprecated(\"Can throw an exception when it's impossible to represent the value with Long type, for example, when the range includes MAX_VALUE. It's recommended to use 'endInclusive' property that doesn't throw.\")\n @SinceKotlin(\"1.7\")\n @ExperimentalStdlibApi\n override val endExclusive: Long get() {\n if (last == Long.MAX_VALUE) error(\"Cannot return the exclusive upper bound of a range that includes MAX_VALUE.\")\n return last + 1\n }\n\n override fun contains(value: Long): Boolean = first <= value && value <= last\n\n /** \n * Checks whether the range is empty.\n *\n * The range is empty if its start value is greater than the end value.\n */\n override fun isEmpty(): Boolean = first > last\n\n override fun equals(other: Any?): Boolean =\n other is LongRange && (isEmpty() && other.isEmpty() ||\n first == other.first && last == other.last)\n\n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 * (first xor (first ushr 32)) + (last xor (last ushr 32))).toInt()\n\n override fun toString(): String = \"$first..$last\"\n\n companion object {\n /** An empty range of values of type Long. */\n public val EMPTY: LongRange = LongRange(1, 0)\n }\n}\n\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"StringsKt\")\n@file:Suppress(\"PLATFORM_CLASS_MAPPED_TO_KOTLIN\")\n\npackage kotlin.text\n\n/**\n * Parses the string as a signed [Byte] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n */\n@SinceKotlin(\"1.1\")\npublic fun String.toByteOrNull(): Byte? = toByteOrNull(radix = 10)\n\n/**\n * Parses the string as a signed [Byte] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n */\n@SinceKotlin(\"1.1\")\npublic fun String.toByteOrNull(radix: Int): Byte? {\n val int = this.toIntOrNull(radix) ?: return null\n if (int < Byte.MIN_VALUE || int > Byte.MAX_VALUE) return null\n return int.toByte()\n}\n\n/**\n * Parses the string as a [Short] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n */\n@SinceKotlin(\"1.1\")\npublic fun String.toShortOrNull(): Short? = toShortOrNull(radix = 10)\n\n/**\n * Parses the string as a [Short] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n */\n@SinceKotlin(\"1.1\")\npublic fun String.toShortOrNull(radix: Int): Short? {\n val int = this.toIntOrNull(radix) ?: return null\n if (int < Short.MIN_VALUE || int > Short.MAX_VALUE) return null\n return int.toShort()\n}\n\n/**\n * Parses the string as an [Int] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n */\n@SinceKotlin(\"1.1\")\npublic fun String.toIntOrNull(): Int? = toIntOrNull(radix = 10)\n\n/**\n * Parses the string as an [Int] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n */\n@SinceKotlin(\"1.1\")\npublic fun String.toIntOrNull(radix: Int): Int? {\n checkRadix(radix)\n\n val length = this.length\n if (length == 0) return null\n\n val start: Int\n val isNegative: Boolean\n val limit: Int\n\n val firstChar = this[0]\n if (firstChar < '0') { // Possible leading sign\n if (length == 1) return null // non-digit (possible sign) only, no

```
digits after\n\n        start = 1\n\n        if (firstChar == '-') {\n            isNegative = true\n            limit = Int.MIN_VALUE\n        } else if (firstChar == '+') {\n            isNegative = false\n            limit = -Int.MAX_VALUE\n        } else\n            return null\n    } else {\n        start = 0\n        isNegative = false\n        limit = -Int.MAX_VALUE\n    }\n\n\n    val limitForMaxRadix = (-Int.MAX_VALUE) / 36\n\n    var limitBeforeMul = limitForMaxRadix\n    var result = 0\n    for (i in start until length) {\n        val digit = digitOf(this[i], radix)\n\n        if (digit
 < 0) return null\n        if (result < limitBeforeMul) {\n            if (limitBeforeMul == limitForMaxRadix) {\n                limitBeforeMul = limit / radix\n\n                if (result < limitBeforeMul) {\n                    return null\n                }\n            } else {\n                return null\n            }\n        }\n\n        result *= radix\n\n        if (result < limit + digit) return null\n\n        result -= digit\n    }\n\n    return if (isNegative) result else -result\n}\n\n/**\n * Parses the string as a [Long] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n */\n@SinceKotlin(\"1.1\")\npublic fun String.toLongOrNull(): Long? = toLongOrNull(radix = 10)\n\n/**\n * Parses the string as a [Long] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n */\n@SinceKotlin(\"1.1\")\npublic
 fun String.toLongOrNull(radix: Int): Long? {\n    checkRadix(radix)\n\n    val length = this.length\n    if (length == 0) return null\n\n    val start: Int\n    val isNegative: Boolean\n    val limit: Long\n\n    val firstChar = this[0]\n    if (firstChar < '0') {  // Possible leading sign\n        if (length == 1) return null  // non-digit (possible sign) only, no digits after\n\n        start = 1\n\n        if (firstChar == '-') {\n            isNegative = true\n            limit = Long.MIN_VALUE\n        } else if (firstChar == '+') {\n            isNegative = false\n            limit = -Long.MAX_VALUE\n        } else\n            return null\n    } else {\n        start = 0\n        isNegative = false\n        limit = -Long.MAX_VALUE\n    }\n\n\n    val limitForMaxRadix = (-Long.MAX_VALUE) / 36\n\n    var limitBeforeMul = limitForMaxRadix\n    var result = 0L\n    for (i in start until length) {\n        val digit = digitOf(this[i], radix)\n\n        if
 (digit < 0) return null\n        if (result < limitBeforeMul) {\n            if (limitBeforeMul == limitForMaxRadix) {\n                limitBeforeMul = limit / radix\n\n                if (result < limitBeforeMul) {\n                    return null\n                }\n            } else {\n                return null\n            }\n        }\n\n        result *= radix\n\n        if (result < limit + digit) return null\n\n        result -= digit\n    }\n\n    return if (isNegative) result else -result\n}\n\n\ninternal fun numberFormatError(input: String): Nothing = throw NumberFormatException(\"Invalid number format: '$input'\")\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport kotlin.contracts.*\nimport kotlin.jvm.JvmInline\nimport kotlin.math.*\n\n/**\n * Represents the amount of time
 one instant of time is away from another instant.\n *\n * A negative duration is possible in a situation when the second instant is earlier than the first one.\n *\n * The type can store duration values up to \u00b1146 years with nanosecond precision,\n * and up to \u00b1146 million years with millisecond precision.\n * If a duration-returning operation provided in `kotlin.time` produces a duration value that doesn't fit into the above range,\n * the returned `Duration` is infinite.\n *\n * An infinite duration value [Duration.INFINITE] can be used to represent infinite timeouts.\n *\n * To construct a duration use either the extension function [toDuration],\n * or the extension properties [hours], [minutes], [seconds], and so on,\n * available on [Int], [Long], and [Double] numeric types.\n *\n * To get the value of this duration expressed in a particular [duration units][DurationUnit]\n * use the functions [toInt], [toLong], and [toDouble]\n * or the properties [inWholeHours], [inWholeMinutes],
 [inWholeSeconds], [inWholeNanoseconds], and so on.\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalTime::class)\n@JvmInline\npublic value class Duration internal constructor(private val rawValue: Long) : Comparable<Duration> {\n\n    private val value: Long get() = rawValue shr 1\n    private inline val unitDiscriminator: Int get() = rawValue.toInt() and 1\n    private fun isInNanos() = unitDiscriminator == 0\n    private fun isInMillis() = unitDiscriminator == 1\n    private val storageUnit get() = if (isInNanos()) DurationUnit.NANOSECONDS else DurationUnit.MILLISECONDS\n\n    init
```

{\n if (durationAssertionsEnabled) {\n if (isInNanos()) {\n if (value !in -MAX_NANOS..MAX_NANOS) throw AssertionError(\"$value ns is out of nanoseconds range\")\n } else {\n if (value !in -MAX_MILLIS..MAX_MILLIS) throw AssertionError(\"$value ms is out of milliseconds range\")\n if (value in -MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) throw AssertionError(\"$value ms is denormalized\")\n }\n }\n }\n\n companion object {\n /** The duration equal to exactly 0 seconds. */\n public val ZERO: Duration = Duration(0L)\n\n /** The duration whose value is positive infinity. It is useful for representing timeouts that should never expire. */\n public val INFINITE: Duration = durationOfMillis(MAX_MILLIS)\n internal val NEG_INFINITE: Duration = durationOfMillis(-MAX_MILLIS)\n\n /** Converts the given time duration [value] expressed in the specified [sourceUnit] into the specified [targetUnit]. */\n @ExperimentalTime\n public fun convert(value: Double, sourceUnit: DurationUnit, targetUnit: DurationUnit): Double =\n convertDurationUnit(value, sourceUnit, targetUnit)\n\n // Duration construction extension properties in Duration companion scope\n\n /** Returns a [Duration] equal to this [Int] number of nanoseconds. */\n @kotlin.internal.InlineOnly\n public inline val Int.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)\n\n /** Returns a [Duration] equal to this [Long] number of nanoseconds. */\n @kotlin.internal.InlineOnly\n public inline val Long.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)\n\n /**\n * Returns a [Duration] equal to this [Double] number of nanoseconds.\n *\n * Depending on its magnitude, the value is rounded to an integer number of nanoseconds or milliseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n */\n @kotlin.internal.InlineOnly\n public inline val Double.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)\n\n\n /** Returns a [Duration] equal to this [Int] number of microseconds. */\n @kotlin.internal.InlineOnly\n public inline val Int.microseconds get() = toDuration(DurationUnit.MICROSECONDS)\n\n

/** Returns a [Duration] equal to this [Long] number of microseconds. */\n @kotlin.internal.InlineOnly\n public inline val Long.microseconds get() = toDuration(DurationUnit.MICROSECONDS)\n\n /**\n * Returns a [Duration] equal to this [Double] number of microseconds.\n *\n * Depending on its magnitude, the value is rounded to an integer number of nanoseconds or milliseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n */\n @kotlin.internal.InlineOnly\n public inline val Double.microseconds get() = toDuration(DurationUnit.MICROSECONDS)\n\n\n /** Returns a [Duration] equal to this [Int] number of milliseconds. */\n @kotlin.internal.InlineOnly\n public inline val Int.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n\n /** Returns a [Duration] equal to this [Long] number of milliseconds. */\n @kotlin.internal.InlineOnly\n

public inline val Long.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n\n /**\n * Returns a [Duration] equal to this [Double] number of milliseconds.\n *\n * Depending on its magnitude, the value is rounded to an integer number of nanoseconds or milliseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n */\n @kotlin.internal.InlineOnly\n public inline val Double.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n\n\n /** Returns a [Duration] equal to this [Int] number of seconds. */\n @kotlin.internal.InlineOnly\n public inline val Int.seconds get() = toDuration(DurationUnit.SECONDS)\n\n /** Returns a [Duration] equal to this [Long] number of seconds. */\n @kotlin.internal.InlineOnly\n public inline val Long.seconds get() = toDuration(DurationUnit.SECONDS)\n\n /**\n * Returns a [Duration] equal to this [Double] number of seconds.\n *\n * Depending on its magnitude, the value is rounded to an integer number of nanoseconds or milliseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n */\n @kotlin.internal.InlineOnly\n public inline val Double.seconds get() = toDuration(DurationUnit.SECONDS)\n\n\n /** Returns a [Duration] equal to this [Int] number of minutes. */\n @kotlin.internal.InlineOnly\n public inline val Int.minutes get() = toDuration(DurationUnit.MINUTES)\n\n

/** Returns a [Duration] equal to this [Long] number of minutes. */\n @kotlin.internal.InlineOnly\n public inline val Long.minutes get() = toDuration(DurationUnit.MINUTES)\n\n /**\n * Returns a

```
[Duration] equal to this [Double] number of minutes.\n *\n * Depending on its magnitude, the value is rounded to an integer number of nanoseconds or milliseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n */\n @kotlin.internal.InlineOnly\n public inline val Double.minutes get() = toDuration(DurationUnit.MINUTES)\n\n\n /** Returns a [Duration] equal to this [Int] number of hours. */\n @kotlin.internal.InlineOnly\n public inline val Int.hours get() = toDuration(DurationUnit.HOURS)\n\n /** Returns a [Duration] equal to this [Long] number of hours. */\n @kotlin.internal.InlineOnly\n public inline val Long.hours get() = toDuration(DurationUnit.HOURS)\n\n /**\n * Returns a [Duration] equal to this [Double] number of hours.\n *\n * Depending on its magnitude, the value is rounded to an integer number of nanoseconds or milliseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n */\n @kotlin.internal.InlineOnly\n public inline val Double.hours get() = toDuration(DurationUnit.HOURS)\n\n\n /** Returns a [Duration] equal to this [Int] number of days. */\n @kotlin.internal.InlineOnly\n public inline val Int.days get() = toDuration(DurationUnit.DAYS)\n\n /** Returns a [Duration] equal to this [Long] number of days. */\n @kotlin.internal.InlineOnly\n public inline val Long.days get() = toDuration(DurationUnit.DAYS)\n\n /**\n * Returns a [Duration] equal to this [Double] number of days.\n *\n * Depending on its magnitude, the value is rounded to an integer number of nanoseconds or milliseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n */\n @kotlin.internal.InlineOnly\n public inline val Double.days get() = toDuration(DurationUnit.DAYS)\n\n\n // deprecated static factory functions\n\n /** Returns a [Duration] representing the specified [value] number of nanoseconds. */\n @SinceKotlin(\"1.5\")\n @ExperimentalTime\n @Deprecated(\"Use 'Int.nanoseconds' extension property from Duration.Companion instead.\", ReplaceWith(\"value.nanoseconds\", \"kotlin.time.Duration.Companion.nanoseconds\"))\n @DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n public fun nanoseconds(value: Int): Duration = value.toDuration(DurationUnit.NANOSECONDS)\n\n /** Returns a [Duration] representing the specified [value] number of nanoseconds. */\n @SinceKotlin(\"1.5\")\n @ExperimentalTime\n @Deprecated(\"Use 'Long.nanoseconds' extension property from Duration.Companion instead.\", ReplaceWith(\"value.nanoseconds\", \"kotlin.time.Duration.Companion.nanoseconds\"))\n @DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n public fun nanoseconds(value: Long): Duration = value.toDuration(DurationUnit.NANOSECONDS)\n\n /**\n * Returns a [Duration] representing the specified\n [value] number of nanoseconds.\n *\n * @throws IllegalArgumentException if the provided `Double` [value] is `NaN`.\n */\n @SinceKotlin(\"1.5\")\n @ExperimentalTime\n @Deprecated(\"Use 'Double.nanoseconds' extension property from Duration.Companion instead.\", ReplaceWith(\"value.nanoseconds\", \"kotlin.time.Duration.Companion.nanoseconds\"))\n @DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n public fun nanoseconds(value: Double): Duration = value.toDuration(DurationUnit.NANOSECONDS)\n\n\n /** Returns a [Duration] representing the specified [value] number of microseconds. */\n @SinceKotlin(\"1.5\")\n @ExperimentalTime\n @Deprecated(\"Use 'Int.microseconds' extension property from Duration.Companion instead.\", ReplaceWith(\"value.microseconds\", \"kotlin.time.Duration.Companion.microseconds\"))\n @DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n public fun microseconds(value: Int): Duration = value.toDuration(DurationUnit.MICROSECONDS)\n\n /** Returns a [Duration] representing the specified [value] number of microseconds. */\n @SinceKotlin(\"1.5\")\n @ExperimentalTime\n @Deprecated(\"Use 'Long.microseconds' extension property from Duration.Companion instead.\", ReplaceWith(\"value.microseconds\", \"kotlin.time.Duration.Companion.microseconds\"))\n @DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n public fun microseconds(value: Long): Duration = value.toDuration(DurationUnit.MICROSECONDS)\n\n /**\n * Returns a [Duration] representing the
```

```
specified [value] number of microseconds.\n         *\n         * @throws IllegalArgumentException if the provided
`Double` [value] is `NaN`.\n         */\n         @SinceKotlin(\"1.5\")\n         @ExperimentalTime\n
@Deprecated(\"Use 'Double.microseconds' extension property from Duration.Companion instead.\",
 ReplaceWith(\"value.microseconds\", \"kotlin.time.Duration.Companion.microseconds\"))\n
@DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n         public fun microseconds(value:
Double): Duration = value.toDuration(DurationUnit.MICROSECONDS)\n\n\n         /** Returns a [Duration]
representing the specified [value] number of milliseconds. */\n         @SinceKotlin(\"1.5\")\n
@ExperimentalTime\n         @Deprecated(\"Use 'Int.milliseconds' extension property from Duration.Companion
instead.\", ReplaceWith(\"value.milliseconds\", \"kotlin.time.Duration.Companion.milliseconds\"))\n
@DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n         public fun milliseconds(value: Int):
Duration = value.toDuration(DurationUnit.MILLISECONDS)\n\n         /** Returns a [Duration] representing the
specified [value] number of milliseconds. */\n         @SinceKotlin(\"1.5\")\n         @ExperimentalTime\n
@Deprecated(\"Use 'Long.milliseconds' extension
 property from Duration.Companion instead.\", ReplaceWith(\"value.milliseconds\",
\"kotlin.time.Duration.Companion.milliseconds\"))\n         @DeprecatedSinceKotlin(warningSince = \"1.6\",
errorSince = \"1.8\")\n         public fun milliseconds(value: Long): Duration =
value.toDuration(DurationUnit.MILLISECONDS)\n\n         /**\n         * Returns a [Duration] representing the
specified [value] number of milliseconds.\n         *\n         * @throws IllegalArgumentException if the provided
`Double` [value] is `NaN`.\n         */\n         @SinceKotlin(\"1.5\")\n         @ExperimentalTime\n
@Deprecated(\"Use 'Double.milliseconds' extension property from Duration.Companion instead.\",
ReplaceWith(\"value.milliseconds\", \"kotlin.time.Duration.Companion.milliseconds\"))\n
@DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n         public fun milliseconds(value:
Double): Duration = value.toDuration(DurationUnit.MILLISECONDS)\n\n\n         /** Returns a [Duration]
 representing the specified [value] number of seconds. */\n         @SinceKotlin(\"1.5\")\n         @ExperimentalTime\n
     @Deprecated(\"Use 'Int.seconds' extension property from Duration.Companion instead.\",
ReplaceWith(\"value.seconds\", \"kotlin.time.Duration.Companion.seconds\"))\n
@DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n         public fun seconds(value: Int):
Duration = value.toDuration(DurationUnit.SECONDS)\n\n         /** Returns a [Duration] representing the specified
[value] number of seconds. */\n         @SinceKotlin(\"1.5\")\n         @ExperimentalTime\n         @Deprecated(\"Use
'Long.seconds' extension property from Duration.Companion instead.\", ReplaceWith(\"value.seconds\",
\"kotlin.time.Duration.Companion.seconds\"))\n         @DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince =
\"1.8\")\n         public fun seconds(value: Long): Duration = value.toDuration(DurationUnit.SECONDS)\n\n
/**\n         * Returns a [Duration]
 representing the specified [value] number of seconds.\n         *\n         * @throws IllegalArgumentException if the
provided `Double` [value] is `NaN`.\n         */\n         @SinceKotlin(\"1.5\")\n         @ExperimentalTime\n
@Deprecated(\"Use 'Double.seconds' extension property from Duration.Companion instead.\",
ReplaceWith(\"value.seconds\", \"kotlin.time.Duration.Companion.seconds\"))\n
@DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n         public fun seconds(value: Double):
Duration = value.toDuration(DurationUnit.SECONDS)\n\n\n         /** Returns a [Duration] representing the
specified [value] number of minutes. */\n         @SinceKotlin(\"1.5\")\n         @ExperimentalTime\n
@Deprecated(\"Use 'Int.minutes' extension property from Duration.Companion instead.\",
ReplaceWith(\"value.minutes\", \"kotlin.time.Duration.Companion.minutes\"))\n
@DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n         public fun minutes(value:
 Int): Duration = value.toDuration(DurationUnit.MINUTES)\n\n         /** Returns a [Duration] representing the
specified [value] number of minutes. */\n         @SinceKotlin(\"1.5\")\n         @ExperimentalTime\n
@Deprecated(\"Use 'Long.minutes' extension property from Duration.Companion instead.\",
ReplaceWith(\"value.minutes\", \"kotlin.time.Duration.Companion.minutes\"))\n
@DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n         public fun minutes(value: Long):
```

```
Duration = value.toDuration(DurationUnit.MINUTES)\n\n        /**\n         * Returns a [Duration] representing the specified [value] number of minutes.\n         *\n         * @throws IllegalArgumentException if the provided `Double` [value] is `NaN`.\n         */\n        @SinceKotlin(\"1.5\")\n        @ExperimentalTime\n        @Deprecated(\"Use 'Double.minutes' extension property from Duration.Companion instead.\", ReplaceWith(\"value.minutes\", \"kotlin.time.Duration.Companion.minutes\"))\n
    @DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n        public fun minutes(value: Double): Duration = value.toDuration(DurationUnit.MINUTES)\n\n\n        /** Returns a [Duration] representing the specified [value] number of hours. */\n        @SinceKotlin(\"1.5\")\n        @ExperimentalTime\n        @Deprecated(\"Use 'Int.hours' extension property from Duration.Companion instead.\", ReplaceWith(\"value.hours\", \"kotlin.time.Duration.Companion.hours\"))\n        @DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n        public fun hours(value: Int): Duration = value.toDuration(DurationUnit.HOURS)\n\n        /** Returns a [Duration] representing the specified [value] number of hours. */\n        @SinceKotlin(\"1.5\")\n        @ExperimentalTime\n        @Deprecated(\"Use 'Long.hours' extension property from Duration.Companion instead.\", ReplaceWith(\"value.hours\", \"kotlin.time.Duration.Companion.hours\"))\n        @DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n        public fun hours(value: Long): Duration = value.toDuration(DurationUnit.HOURS)\n\n        /**\n         * Returns a [Duration] representing the specified [value] number of hours.\n         *\n         * @throws IllegalArgumentException if the provided `Double` [value] is `NaN`.\n         */\n        @SinceKotlin(\"1.5\")\n        @ExperimentalTime\n        @Deprecated(\"Use 'Double.hours' extension property from Duration.Companion instead.\", ReplaceWith(\"value.hours\", \"kotlin.time.Duration.Companion.hours\"))\n        @DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n        public fun hours(value: Double): Duration = value.toDuration(DurationUnit.HOURS)\n\n\n        /** Returns a [Duration] representing the specified [value] number of days. */\n        @SinceKotlin(\"1.5\")\n        @ExperimentalTime\n        @Deprecated(\"Use 'Int.days' extension property from Duration.Companion instead.\", ReplaceWith(\"value.days\", \"kotlin.time.Duration.Companion.days\"))\n        @DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n        public fun days(value: Int): Duration = value.toDuration(DurationUnit.DAYS)\n\n        /** Returns a [Duration] representing the specified [value] number of days. */\n        @SinceKotlin(\"1.5\")\n        @ExperimentalTime\n        @Deprecated(\"Use 'Long.days' extension property from Duration.Companion instead.\", ReplaceWith(\"value.days\", \"kotlin.time.Duration.Companion.days\"))\n        @DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n        public fun days(value: Long): Duration = value.toDuration(DurationUnit.DAYS)\n\n        /**\n         * Returns a [Duration] representing the specified [value] number of days.\n         *\n         * @throws IllegalArgumentException if the provided `Double` [value] is `NaN`.\n         */\n        @SinceKotlin(\"1.5\")\n        @ExperimentalTime\n        @Deprecated(\"Use 'Double.days' extension property from Duration.Companion instead.\", ReplaceWith(\"value.days\", \"kotlin.time.Duration.Companion.days\"))\n        @DeprecatedSinceKotlin(warningSince = \"1.6\", errorSince = \"1.8\")\n        public fun days(value: Double): Duration = value.toDuration(DurationUnit.DAYS)\n\n        /**\n         * Parses a string that represents a duration and returns the parsed [Duration] value.\n         *\n         * The following formats are accepted:\n         *\n         * - ISO-8601 Duration format, e.g. `P1DT2H3M4.058S`, see [toIsoString] and [parseIsoString].\n         * - The format of string returned by the default [Duration.toString] and `toString` in a specific unit,\n         *   e.g. `10s`, `1h 30m` or `-(1h 30m)`.\n         *\n         * @throws IllegalArgumentException if the string doesn't represent a duration in any of the supported formats.\n         * @sample samples.time.Durations.parse\n         */\n        public fun parse(value: String): Duration = try {\n            parseDuration(value, strictIso = false)\n        } catch (e: IllegalArgumentException) {\n            throw IllegalArgumentException(\"Invalid duration string format: '$value'.\", e)\n        }\n\n        /**\n         * Parses a string that represents a duration in a restricted ISO-8601 composite representation\n         * and returns the parsed [Duration] value.\n         * Composite representation is a relaxed version of ISO-8601 duration format that supports\n         * negative durations and negative values of individual components.\n         *\n         * The following
```

```
restrictions are imposed:\n *\n * - The only allowed non-time designator is days (`D`). `Y` (years), `W` (weeks), and `M` (months) are not supported.\n * - Day is considered to be exactly 24 hours (24-hour clock time scale).\n * - Alternative week-based representation `[\"P\"][number][\"W\"]` is not supported.\n *\n * @throws IllegalArgumentException if the string doesn't represent a duration in ISO-8601 format.\n * @sample samples.time.Durations.parseIsoString\n */\n public fun parseIsoString(value: String): Duration = try {\n parseDuration(value, strictIso = true)\n } catch (e: IllegalArgumentException) {\n throw IllegalArgumentException(\"Invalid ISO duration string format: '$value'.\", e)\n }\n\n /**\n * Parses a string that represents a duration and returns the parsed [Duration] value,\n * or `null` if the string doesn't represent a duration in any of the supported formats.\n *\n * The following formats are accepted:\n *\n * - Restricted ISO-8601 duration composite representation, e.g. `P1DT2H3M4.058S`, see [toIsoString] and [parseIsoString].\n * - The format of string returned by the default [Duration.toString] and `toString` in a specific unit,\n * e.g. `10s`, `1h 30m` or `-(1h 30m)`.\n * @sample samples.time.Durations.parse\n */\n public fun parseOrNull(value: String): Duration? = try {\n parseDuration(value, strictIso = false)\n } catch (e: IllegalArgumentException) {\n null\n }\n\n /**\n * Parses a string that represents a duration in restricted ISO-8601 composite representation\n * and returns the parsed [Duration] value or `null` if the string doesn't represent a duration in the format\n * acceptable by [parseIsoString].\n *\n * @sample samples.time.Durations.parseIsoString\n */\n public fun parseIsoStringOrNull(value: String): Duration? = try {\n parseDuration(value, strictIso = true)\n } catch (e: IllegalArgumentException) {\n null\n }\n }\n\n // arithmetic operators\n\n /** Returns the negative of this value. */\n public operator fun unaryMinus(): Duration = durationOf(-value, unitDiscriminator)\n\n\n /**\n * Returns a duration whose value is the sum of this and [other] duration values.\n *\n * @throws IllegalArgumentException if the operation results in an undefined value for the given arguments,\n * e.g. when adding infinite durations of different sign.\n */\n public operator fun plus(other: Duration): Duration {\n when {\n this.isInfinite() -> {\n if (other.isFinite() || (this.rawValue xor other.rawValue >= 0))\n return this\n else\n throw IllegalArgumentException(\"Summing infinite durations of different signs yields an undefined result.\")\n }\n other.isInfinite() -> return other\n }\n\n return when {\n this.unitDiscriminator == other.unitDiscriminator -> {\n val result = this.value + other.value // never overflows long, but can overflow long63\n when {\n isInNanos() ->\n durationOfNanosNormalized(result)\n else ->\n durationOfMillisNormalized(result)\n }\n }\n this.isInMillis() ->\n addValuesMixedRanges(this.value, other.value)\n else ->\n addValuesMixedRanges(other.value, this.value)\n }\n }\n\n private fun addValuesMixedRanges(thisMillis: Long, otherNanos: Long): Duration {\n val otherMillis = nanosToMillis(otherNanos)\n val resultMillis = thisMillis + otherMillis\n return if (resultMillis in -MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) {\n val otherNanoRemainder = otherNanos - millisToNanos(otherMillis)\n durationOfNanos(millisToNanos(resultMillis) + otherNanoRemainder)\n } else {\n durationOfMillis(resultMillis.coerceIn(-MAX_MILLIS, MAX_MILLIS))\n }\n }\n\n /**\n * Returns a duration whose value is the difference between this and [other] duration values.\n *\n * @throws IllegalArgumentException if the operation results in an undefined value for the given arguments,\n * e.g. when subtracting infinite durations of the same sign.\n */\n public operator fun minus(other: Duration): Duration = this + (-other)\n\n /**\n * Returns a duration whose value is this duration value multiplied by the given [scale] number.\n *\n * @throws IllegalArgumentException if the operation results in an undefined value for the given arguments,\n * e.g. when multiplying an infinite duration by zero.\n */\n public operator fun times(scale: Int): Duration {\n if (isInfinite()) {\n return when {\n scale == 0 -> throw IllegalArgumentException(\"Multiplying infinite duration by zero yields an undefined result.\")\n scale > 0 -> this\n else -> -this\n }\n }\n if (scale == 0)\n return ZERO\n\n
```

```
    val value = value\n        val result = value * scale\n        return if (isInNanos()) {\n            if (value in (MAX_NANOS / Int.MIN_VALUE)..(-MAX_NANOS / Int.MIN_VALUE)) {\n                // can't overflow nanos range for any scale\n                durationOfNanos(result)\n            } else {\n                if (result / scale == value) {\n                    durationOfNanosNormalized(result)\n                } else {\n                    val millis = nanosToMillis(value)\n                    val remNanos = value - millisToNanos(millis)\n                    val resultMillis = millis * scale\n                    val totalMillis = resultMillis + nanosToMillis(remNanos * scale)\n                    if (resultMillis / scale == millis && totalMillis xor resultMillis >= 0) {\n                        durationOfMillis(totalMillis.coerceIn(-MAX_MILLIS..MAX_MILLIS))\n                    } else {\n                        if (value.sign * scale.sign > 0) INFINITE else NEG_INFINITE\n                    }\n                }\n            }\n        } else {\n            if (result / scale == value) {\n                durationOfMillis(result.coerceIn(-MAX_MILLIS..MAX_MILLIS))\n            } else {\n                if (value.sign * scale.sign > 0) INFINITE else NEG_INFINITE\n            }\n        }\n    }\n\n    /**\n     * Returns a duration whose value is this duration value multiplied by the given [scale] number.\n     *\n     * The operation may involve rounding when the result cannot be represented exactly with a [Double] number.\n     *\n     * @throws IllegalArgumentException if the operation results in an undefined value for the given arguments,\n     * e.g. when multiplying an infinite duration by zero.\n     */\n    public operator fun times(scale: Double): Duration {\n        val intScale = scale.roundToInt()\n        if (intScale.toDouble() == scale) {\n            return times(intScale)\n        }\n\n        val unit = storageUnit\n        val result = toDouble(unit) * scale\n        return result.toDuration(unit)\n    }\n\n    /**\n     * Returns a duration whose value is this duration value divided by the given [scale] number.\n     *\n     * @throws IllegalArgumentException if the operation results in an undefined value for the given arguments,\n     * e.g. when dividing zero duration by zero.\n     */\n    public operator fun div(scale: Int): Duration {\n        if (scale == 0) {\n            return when {\n                isPositive() -> INFINITE\n                isNegative() -> NEG_INFINITE\n                else -> throw IllegalArgumentException(\"Dividing zero duration by zero yields an undefined result.\")\n            }\n        }\n        if (isInNanos()) {\n            return durationOfNanos(value / scale)\n        } else {\n            if (isInfinite())\n                return this * scale.sign\n\n            val result = value / scale\n\n            if (result in -MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) {\n                val rem = millisToNanos(value - (result * scale)) / scale\n                return durationOfNanos(millisToNanos(result) + rem)\n            }\n            return durationOfMillis(result)\n        }\n    }\n\n    /**\n     * Returns a duration whose value is this duration value divided by the given [scale] number.\n     *\n     * @throws IllegalArgumentException if the operation results in an undefined value for the given arguments,\n     * e.g. when dividing an infinite duration by infinity or zero duration by zero.\n     */\n    public operator fun div(scale: Double): Duration {\n        val intScale = scale.roundToInt()\n        if (intScale.toDouble() == scale && intScale != 0) {\n            return div(intScale)\n        }\n\n        val unit = storageUnit\n        val result = toDouble(unit) / scale\n        return result.toDuration(unit)\n    }\n\n    /** Returns a number that is the ratio of this and [other] duration values. */\n    public operator fun div(other: Duration): Double {\n        val coarserUnit = maxOf(this.storageUnit, other.storageUnit)\n        return this.toDouble(coarserUnit) / other.toDouble(coarserUnit)\n    }\n\n    /** Returns true, if the duration value is less than zero. */\n    public fun isNegative(): Boolean = rawValue < 0\n\n    /** Returns true, if the duration value is greater than zero. */\n    public fun isPositive(): Boolean = rawValue > 0\n\n    /** Returns true, if the duration value is infinite. */\n    public fun isInfinite(): Boolean = rawValue == INFINITE.rawValue || rawValue == NEG_INFINITE.rawValue\n\n    /** Returns true, if the duration value is finite. */\n    public fun isFinite(): Boolean = !isInfinite()\n\n    /** Returns the absolute value of this value. The returned value is always non-negative. */\n    public val absoluteValue: Duration get() = if (isNegative()) -this else this\n\n    override fun compareTo(other: Duration): Int {\n        val compareBits = this.rawValue xor other.rawValue\n        if (compareBits < 0 || compareBits.toInt() and 1 == 0) // different signs or same sign/same range\n            return this.rawValue.compareTo(other.rawValue)\n        // same sign/different ranges\n        val r = this.unitDiscriminator - other.unitDiscriminator // compare ranges\n        return if (isNegative()) -r else r\n    }\n\n    // splitting to
```

```
components\n\n    /**\n     * Splits this duration into days, hours, minutes, seconds, and nanoseconds and executes the given [action] with these components.\n     * The result of [action] is returned as the result of this function.\n     *\n     * - `nanoseconds` represents the whole number of nanoseconds in this duration, and its absolute value is less than 1_000_000_000;\n     * - `seconds` represents the whole number of seconds in this duration, and its absolute value is less than 60;\n     * - `minutes` represents the whole number of minutes in this duration, and its absolute value is less than 60;\n     * - `hours` represents the whole number of hours in this duration, and its absolute value is less than 24;\n     * - `days` represents the whole number of days in this duration.\n     *\n     *  Infinite durations are represented as either [Long.MAX_VALUE] days, or [Long.MIN_VALUE] days (depending on the sign of infinity),\n     *  and zeroes in the lower components.\n     */\n    public inline fun <T> toComponents(action: (days: Long, hours: Int, minutes: Int, seconds: Int, nanoseconds: Int) -> T): T {\n        contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE) }\n        return action(inWholeDays, hoursComponent, minutesComponent, secondsComponent, nanosecondsComponent)\n    }\n\n    /**\n     * Splits this duration into hours, minutes, seconds, and nanoseconds and executes the given [action] with these components.\n     * The result of [action] is returned as the result of this function.\n     *\n     * - `nanoseconds` represents the whole number of nanoseconds in this duration, and its absolute value is less than 1_000_000_000;\n     * - `seconds` represents the whole number of seconds in this duration, and its absolute value is less than 60;\n     * - `minutes` represents the whole number of minutes in this duration, and its absolute value is less than 60;\n     * - `hours` represents the whole number of hours in this duration.\n     *\n     *  Infinite durations are represented as either [Long.MAX_VALUE] hours, or [Long.MIN_VALUE] hours (depending on the sign of infinity),\n     *  and zeroes in the lower components.\n     */\n    public inline fun <T> toComponents(action: (hours: Long, minutes: Int, seconds: Int, nanoseconds: Int) -> T): T {\n        contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE) }\n        return action(inWholeHours, minutesComponent, secondsComponent, nanosecondsComponent)\n    }\n\n    /**\n     * Splits this duration into minutes, seconds, and nanoseconds and executes the given [action] with these components.\n     * The result of [action] is returned as the result of this function.\n     *\n     * - `nanoseconds` represents the whole number of nanoseconds in this duration, and its absolute value is less than 1_000_000_000;\n     * - `seconds` represents the whole number of seconds in this duration, and its absolute value is less than 60;\n     * - `minutes` represents the whole number of minutes in this duration.\n     *\n     *  Infinite durations are represented as either [Long.MAX_VALUE] minutes, or [Long.MIN_VALUE] minutes (depending on the sign of infinity),\n     *  and zeroes in the lower components.\n     */\n    public inline fun <T> toComponents(action: (minutes: Long, seconds: Int, nanoseconds: Int) -> T): T {\n        contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE) }\n        return action(inWholeMinutes, secondsComponent, nanosecondsComponent)\n    }\n\n    /**\n     * Splits this duration into seconds, and nanoseconds and executes the given [action] with these components.\n     * The result of [action] is returned as the result of this function.\n     *\n     * - `nanoseconds` represents the whole number of nanoseconds in this duration, and its absolute value is less than 1_000_000_000;\n     * - `seconds` represents the whole number of seconds in this duration.\n     *\n     *  Infinite durations are represented as either [Long.MAX_VALUE] seconds, or [Long.MIN_VALUE] seconds (depending on the sign of infinity),\n     *  and zero nanoseconds.\n     */\n    public inline fun <T> toComponents(action: (seconds: Long, nanoseconds: Int) -> T): T {\n        contract { callsInPlace(action, InvocationKind.EXACTLY_ONCE) }\n        return action(inWholeSeconds, nanosecondsComponent)\n    }\n\n    @PublishedApi\n    internal val hoursComponent: Int\n        get() = if (isInfinite()) 0 else (inWholeHours % 24).toInt()\n\n    @PublishedApi\n    internal val minutesComponent: Int\n        get() = if (isInfinite()) 0 else (inWholeMinutes % 60).toInt()\n\n    @PublishedApi\n    internal val secondsComponent: Int\n        get() = if (isInfinite()) 0 else (inWholeSeconds % 60).toInt()\n\n    @PublishedApi\n    internal val nanosecondsComponent: Int\n        get() = when {\n            isInfinite() -> 0\n            isInMillis() -> millisToNanos(value % 1_000).toInt()\n            else -> (value % 1_000_000_000).toInt()\n        }\n\n\n    // conversion to units\n\n    /**\n     * Returns the value of this duration expressed as a [Double] number of the specified [unit].\n     *\n     * The operation may involve rounding when the result cannot be represented exactly with
```

```
a [Double] number.\n     *\n     * An infinite duration value is converted either to [Double.POSITIVE_INFINITY] or [Double.NEGATIVE_INFINITY] depending on its sign.\n     */\n    public fun toDouble(unit: DurationUnit): Double {\n        return when (rawValue) {\n            INFINITE.rawValue -> Double.POSITIVE_INFINITY\n            NEG_INFINITE.rawValue -> Double.NEGATIVE_INFINITY\n            else -> {\n                // TODO: whether it's ok to convert to Double before scaling\n                convertDurationUnit(value.toDouble(), storageUnit, unit)\n            }\n        }\n    }\n\n    /**\n     * Returns the value of this duration expressed as a [Long] number of the specified [unit].\n     *\n     * If the result doesn't fit in the range of [Long] type, it is coerced into that range:\n     * - [Long.MIN_VALUE] is returned if it's less than `Long.MIN_VALUE`,\n     * - [Long.MAX_VALUE] is returned if it's greater than `Long.MAX_VALUE`.\n     *\n     * An infinite duration value is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending on its sign.\n     */\n    public fun toLong(unit: DurationUnit): Long {\n        return when (rawValue) {\n            INFINITE.rawValue -> Long.MAX_VALUE\n            NEG_INFINITE.rawValue -> Long.MIN_VALUE\n            else -> convertDurationUnit(value, storageUnit, unit)\n        }\n    }\n\n    /**\n     * Returns the value of this duration\n     expressed as an [Int] number of the specified [unit].\n     *\n     * If the result doesn't fit in the range of [Int] type, it is coerced into that range:\n     * - [Int.MIN_VALUE] is returned if it's less than `Int.MIN_VALUE`,\n     * - [Int.MAX_VALUE] is returned if it's greater than `Int.MAX_VALUE`.\n     *\n     * An infinite duration value is converted either to [Int.MAX_VALUE] or [Int.MIN_VALUE] depending on its sign.\n     */\n    public fun toInt(unit: DurationUnit): Int =\n        toLong(unit).coerceIn(Int.MIN_VALUE.toLong(), Int.MAX_VALUE.toLong()).toInt()\n\n    /** The value of this duration expressed as a [Double] number of days. */\n    @ExperimentalTime\n    @Deprecated(\"Use inWholeDays property instead or convert toDouble(DAYS) if a double value is required.\", ReplaceWith(\"toDouble(DurationUnit.DAYS)\"))\n    @DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\n    public val inDays: Double get() = toDouble(DurationUnit.DAYS)\n\n    /** The value of\n     this duration expressed as a [Double] number of hours. */\n    @ExperimentalTime\n    @Deprecated(\"Use inWholeHours property instead or convert toDouble(HOURS) if a double value is required.\", ReplaceWith(\"toDouble(DurationUnit.HOURS)\"))\n    @DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\n    public val inHours: Double get() = toDouble(DurationUnit.HOURS)\n\n    /** The value of this duration expressed as a [Double] number of minutes. */\n    @ExperimentalTime\n    @Deprecated(\"Use inWholeMinutes property instead or convert toDouble(MINUTES) if a double value is required.\", ReplaceWith(\"toDouble(DurationUnit.MINUTES)\"))\n    @DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\n    public val inMinutes: Double get() = toDouble(DurationUnit.MINUTES)\n\n    /** The value of this duration expressed as a [Double] number of seconds. */\n    @ExperimentalTime\n    @Deprecated(\"Use inWholeSeconds property instead or convert toDouble(SECONDS) if\n     a double value is required.\", ReplaceWith(\"toDouble(DurationUnit.SECONDS)\"))\n    @DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\n    public val inSeconds: Double get() = toDouble(DurationUnit.SECONDS)\n\n    /** The value of this duration expressed as a [Double] number of milliseconds. */\n    @ExperimentalTime\n    @Deprecated(\"Use inWholeMilliseconds property instead or convert toDouble(MILLISECONDS) if a double value is required.\", ReplaceWith(\"toDouble(DurationUnit.MILLISECONDS)\"))\n    @DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\n    public val inMilliseconds: Double get() = toDouble(DurationUnit.MILLISECONDS)\n\n    /** The value of this duration expressed as a [Double] number of microseconds. */\n    @ExperimentalTime\n    @Deprecated(\"Use inWholeMicroseconds property instead or convert toDouble(MICROSECONDS) if a double value is required.\", ReplaceWith(\"toDouble(DurationUnit.MICROSECONDS)\"))\n    @DeprecatedSinceKotlin(warningSince\n     = \"1.5\", errorSince = \"1.8\")\n    public val inMicroseconds: Double get() = toDouble(DurationUnit.MICROSECONDS)\n\n    /** The value of this duration expressed as a [Double] number of nanoseconds. */\n    @ExperimentalTime\n    @Deprecated(\"Use inWholeNanoseconds property instead or convert
```

```
toDouble(NANOSECONDS) if a double value is required.\",
ReplaceWith(\"toDouble(DurationUnit.NANOSECONDS)\"))\n    @DeprecatedSinceKotlin(warningSince =
\"1.5\", errorSince = \"1.8\")\n    public val inNanoseconds: Double get() =
toDouble(DurationUnit.NANOSECONDS)\n\n\n    /**\n     * The value of this duration expressed as a [Long]
number of days.\n     *\n     * An infinite duration value is converted either to [Long.MAX_VALUE] or
[Long.MIN_VALUE] depending on its sign.\n     */\n    public val inWholeDays: Long\n        get() =
toLong(DurationUnit.DAYS)\n\n    /**\n     * The value of this duration expressed as a [Long] number of hours.\n
*\n     * An infinite duration value
 is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending on its sign.\n     */\n    public val
inWholeHours: Long\n        get() = toLong(DurationUnit.HOURS)\n\n    /**\n     * The value of this duration
expressed as a [Long] number of minutes.\n     *\n     * An infinite duration value is converted either to
[Long.MAX_VALUE] or [Long.MIN_VALUE] depending on its sign.\n     */\n    public val inWholeMinutes:
Long\n        get() = toLong(DurationUnit.MINUTES)\n\n    /**\n     * The value of this duration expressed as a
[Long] number of seconds.\n     *\n     * An infinite duration value is converted either to [Long.MAX_VALUE] or
[Long.MIN_VALUE] depending on its sign.\n     */\n    public val inWholeSeconds: Long\n        get() =
toLong(DurationUnit.SECONDS)\n\n    /**\n     * The value of this duration expressed as a [Long] number of
milliseconds.\n     *\n     * An infinite duration value is converted either to [Long.MAX_VALUE] or
[Long.MIN_VALUE] depending on its sign.\n
     */\n    public val inWholeMilliseconds: Long\n        get() {\n            return if (isInMillis() && isFinite()) value
else toLong(DurationUnit.MILLISECONDS)\n        }\n\n    /**\n     * The value of this duration expressed as a
[Long] number of microseconds.\n     *\n     * If the result doesn't fit in the range of [Long] type, it is coerced into
that range:\n     * - [Long.MIN_VALUE] is returned if it's less than `Long.MIN_VALUE`,\n     * -
[Long.MAX_VALUE] is returned if it's greater than `Long.MAX_VALUE`.\n     *\n     * An infinite duration value
is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending on its sign.\n     */\n    public val
inWholeMicroseconds: Long\n        get() = toLong(DurationUnit.MICROSECONDS)\n\n    /**\n     * The value of
this duration expressed as a [Long] number of nanoseconds.\n     *\n     * If the result doesn't fit in the range of
[Long] type, it is coerced into that range:\n     * - [Long.MIN_VALUE] is returned if it's less than
 `Long.MIN_VALUE`,\n     * - [Long.MAX_VALUE] is returned if it's greater than `Long.MAX_VALUE`.\n     *\n
   * An infinite duration value is converted either to [Long.MAX_VALUE] or [Long.MIN_VALUE] depending on
its sign.\n     */\n    public val inWholeNanoseconds: Long\n        get() {\n            val value = value\n            return
when {\n                isInNanos() -> value\n                value > Long.MAX_VALUE / NANOS_IN_MILLIS ->
Long.MAX_VALUE\n                value < Long.MIN_VALUE / NANOS_IN_MILLIS -> Long.MIN_VALUE\n
      else -> millisToNanos(value)\n            }\n        }\n\n    // shortcuts\n\n    /**\n     * Returns the value of this
duration expressed as a [Long] number of nanoseconds.\n     *\n     * If the value doesn't fit in the range of [Long]
type, it is coerced into that range, see the conversion [Double.toLong] for details.\n     *\n     * The range of
durations that can be expressed as a `Long` number of nanoseconds is approximately \u00b1292
 years.\n     */\n    @ExperimentalTime\n    @Deprecated(\"Use inWholeNanoseconds property instead.\",
ReplaceWith(\"this.inWholeNanoseconds\"))\n    @DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince =
\"1.8\")\n    public fun toLongNanoseconds(): Long = inWholeNanoseconds\n\n    /**\n     * Returns the value of
this duration expressed as a [Long] number of milliseconds.\n     *\n     * The value is coerced to the range of [Long]
type, if it doesn't fit in that range, see the conversion [Double.toLong] for details.\n     *\n     * The range of
durations that can be expressed as a `Long` number of milliseconds is approximately \u00b1292 million years.\n
*/\n    @ExperimentalTime\n    @Deprecated(\"Use inWholeMilliseconds property instead.\",
ReplaceWith(\"this.inWholeMilliseconds\"))\n    @DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince =
\"1.8\")\n    public fun toLongMilliseconds(): Long = inWholeMilliseconds\n\n    /**\n     * Returns a string
representation of this
 duration value\n     * expressed as a combination of numeric components, each in its own unit.\n     *\n     * Each
component is a number followed by the unit abbreviated name: `d`, `h`, `m`, `s`:\n     * `5h`, `1d 12h`, `1h 0m
```

```
30.340s`.\n     * The last component, usually seconds, can be a number with a fractional part.\n     *\n     * If the duration is less than a second, it is represented as a single number\n     * with one of sub-second units: `ms` (milliseconds), `us` (microseconds), or `ns` (nanoseconds):\n     * `140.884ms`, `500us`, `24ns`.\n     *\n     * A negative duration is prefixed with `-` sign and, if it consists of multiple components, surrounded with parentheses:\n     * `-12m` and `-(1h 30m)`.\n     *\n     * Special cases:\n     *  - an infinite duration is formatted as `\"Infinity\"` or `\"-Infinity\"` without a unit.\n     *\n     * It's recommended to use [toIsoString] that uses more strict ISO-8601 format instead of this `toString`\n     * when you want to
     convert a duration to a string in cases of serialization, interchange, etc.\n     *\n     * @sample samples.time.Durations.toStringDefault\n     */\n    override fun toString(): String = when (rawValue) {\n        0L -> \"0s\"\n        INFINITE.rawValue -> \"Infinity\"\n        NEG_INFINITE.rawValue -> \"-Infinity\"\n        else -> {\n            val isNegative = isNegative()\n            buildString {\n                if (isNegative) append('-')\n                absoluteValue.toComponents { days, hours, minutes, seconds, nanoseconds ->\n                    val hasDays = days != 0L\n                    val hasHours = hours != 0\n                    val hasMinutes = minutes != 0\n                    val hasSeconds = seconds != 0 || nanoseconds != 0\n                    var components = 0\n                    if (hasDays) {\n                        append(days).append('d')\n                        components++\n                    }\n                    if (hasHours || (hasDays && (hasMinutes
     || hasSeconds))) {\n                        if (components++ > 0) append(' ')\n                        append(hours).append('h')\n                    }\n                    if (hasMinutes || (hasSeconds && (hasHours || hasDays))) {\n                        if (components++ > 0) append(' ')\n                        append(minutes).append('m')\n                    }\n                    if (hasSeconds) {\n                        if (components++ > 0) append(' ')\n                        when {\n                            seconds != 0 || hasDays || hasHours || hasMinutes ->\n                                appendFractional(seconds, nanoseconds, 9, \"s\", isZeroes = false)\n                            nanoseconds >= 1_000_000 ->\n                                appendFractional(nanoseconds / 1_000_000, nanoseconds % 1_000_000, 6, \"ms\", isZeroes = false)\n                            nanoseconds >= 1_000 ->\n                                appendFractional(nanoseconds / 1_000, nanoseconds
     % 1_000, 3, \"us\", isZeroes = false)\n                            else ->\n                                append(nanoseconds).append(\"ns\")\n                        }\n                    }\n                    if (isNegative && components > 1) insert(1, '(').append(')')\n                }\n            }\n        }\n    }\n\n    private fun StringBuilder.appendFractional(whole: Int, fractional: Int, fractionalSize: Int, unit: String, isZeroes: Boolean) {\n        append(whole)\n        if (fractional != 0) {\n            append('.')\n            val fracString = fractional.toString().padStart(fractionalSize, '0')\n            val nonZeroDigits = fracString.indexOfLast { it != '0' } + 1\n            when {\n                !isZeroes && nonZeroDigits < 3 -> appendRange(fracString, 0, nonZeroDigits)\n                else -> appendRange(fracString, 0, ((nonZeroDigits + 2) / 3) * 3)\n            }\n        }\n        append(unit)\n    }\n\n    /**\n     * Returns a string representation
     of this duration value expressed in the given [unit]\n     * and formatted with the specified [decimals] number of digits after decimal point.\n     *\n     * Special cases:\n     *  - an infinite duration is formatted as `\"Infinity\"` or `\"-Infinity\"` without a unit.\n     *\n     * @param decimals the number of digits after decimal point to show. The value must be non-negative.\n     * No more than 12 decimals will be shown, even if a larger number is requested.\n     *\n     * @return the value of duration in the specified [unit] followed by that unit abbreviated name: `d`, `h`, `m`, `s`, `ms`, `us`, or `ns`.\n     *\n     * @throws IllegalArgumentException if [decimals] is less than zero.\n     *\n     * @sample samples.time.Durations.toStringDecimals\n     */\n    public fun toString(unit: DurationUnit, decimals: Int = 0): String {\n        require(decimals >= 0) { \"decimals must be not negative, but was $decimals\" }\n        val number = toDouble(unit)\n        if (number.isInfinite()) return number.toString()\n        return formatToExactDecimals(number, decimals.coerceAtMost(12)) + unit.shortName()\n    }\n\n    /**\n     * Returns an ISO-8601 based string representation of this duration.\n     *\n     * The returned value is presented in the format `PThHmMs.fS`, where `h`, `m`, `s` are the integer components of this duration (see [toComponents])\n     * and `f` is a fractional part of second. Depending on the roundness of the value the fractional part can be formatted with either\n     * 0, 3, 6, or 9 decimal digits.\n     *\n     * The infinite duration is represented as `\"PT9999999999999H\"` which is larger than any
```

```
possible finite duration in Kotlin.\n     *\n     * Negative durations are indicated with the sign `-` in the beginning of the returned string, for example, `\"-PT5M30S\"`.\n     *\n     * @sample samples.time.Durations.toIsoString\n     */\n    public fun toIsoString(): String = buildString {\n        if (isNegative()) append('-')\n        append(\"PT\")\n        this@Duration.absoluteValue.toComponents { hours, minutes, seconds, nanoseconds ->\n            @Suppress(\"NAME_SHADOWING\")\n            var hours = hours\n            if (isInfinite()) {\n                // use large enough value instead of Long.MAX_VALUE\n                hours = 9_999_999_999_999\n            }\n            val hasHours = hours != 0L\n            val hasSeconds = seconds != 0 || nanoseconds != 0\n            val hasMinutes = minutes != 0 || (hasSeconds && hasHours)\n            if (hasHours) {\n                append(hours).append('H')\n            }\n            if (hasMinutes) {\n                append(minutes).append('M')\n            }\n            if (hasSeconds || (!hasHours && !hasMinutes)) {\n                appendFractional(seconds, nanoseconds, 9, \"S\", isOzeroes = true)\n            }\n        }\n    }\n\n}\n\n// constructing from number of units\n// extension functions\n\n/** Returns a [Duration] equal to this
 [Int] number of the specified [unit].
*/\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalTime::class)\npublic fun Int.toDuration(unit: DurationUnit): Duration {\n    return if (unit <= DurationUnit.SECONDS) {\n        durationOfNanos(convertDurationUnitOverflow(this.toLong(), unit, DurationUnit.NANOSECONDS))\n    } else\n        toLong().toDuration(unit)\n}\n\n/** Returns a [Duration] equal to this [Long] number of the specified [unit].
*/\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalTime::class)\npublic fun Long.toDuration(unit: DurationUnit): Duration {\n    val maxNsInUnit = convertDurationUnitOverflow(MAX_NANOS, DurationUnit.NANOSECONDS, unit)\n    if (this in -maxNsInUnit..maxNsInUnit) {\n        return durationOfNanos(convertDurationUnitOverflow(this, unit, DurationUnit.NANOSECONDS))\n    } else {\n        val millis = convertDurationUnit(this, unit, DurationUnit.MILLISECONDS)\n        return durationOfMillis(millis.coerceIn(-MAX_MILLIS, MAX_MILLIS))\n    }\n}\n\n/**\n
 * Returns a [Duration] equal to this [Double] number of the specified [unit].\n *\n * Depending on its magnitude, the value is rounded to an integer number of nanoseconds or milliseconds.\n *\n * @throws IllegalArgumentException if this `Double` value is `NaN`.\n
*/\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalTime::class)\npublic fun Double.toDuration(unit: DurationUnit): Duration {\n    val valueInNs = convertDurationUnit(this, unit, DurationUnit.NANOSECONDS)\n    require(!valueInNs.isNaN()) { \"Duration value cannot be NaN.\" }\n    val nanos = valueInNs.roundToLong()\n    return if (nanos in -MAX_NANOS..MAX_NANOS) {\n        durationOfNanos(nanos)\n    } else {\n        val millis = convertDurationUnit(this, unit, DurationUnit.MILLISECONDS).roundToLong()\n        durationOfMillisNormalized(millis)\n    }\n}\n\n// constructing from number of units\n// deprecated extension properties\n\n/** Returns a [Duration] equal to this [Int] number of nanoseconds.
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use
 'Int.nanoseconds' extension property from Duration.Companion instead.\", ReplaceWith(\"this.nanoseconds\", \"kotlin.time.Duration.Companion.nanoseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Int.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)\n\n/** Returns a [Duration] equal to this [Long] number of nanoseconds.
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Long.nanoseconds' extension property from Duration.Companion instead.\", ReplaceWith(\"this.nanoseconds\", \"kotlin.time.Duration.Companion.nanoseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Long.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)\n\n/**\n * Returns a [Duration] equal to this [Double] number of nanoseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use
 'Double.nanoseconds' extension property from Duration.Companion instead.\", ReplaceWith(\"this.nanoseconds\", \"kotlin.time.Duration.Companion.nanoseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Double.nanoseconds get() = toDuration(DurationUnit.NANOSECONDS)\n\n\n/** Returns a [Duration] equal to this [Int] number of microseconds.
```

*/\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Int.microseconds' extension property from Duration.Companion instead.\", ReplaceWith(\"this.microseconds\", \"kotlin.time.Duration.Companion.microseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Int.microseconds get() = toDuration(DurationUnit.MICROSECONDS)\n\n/** Returns a [Duration] equal to this [Long] number of microseconds. */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Long.microseconds' extension property from Duration.Companion instead.\", ReplaceWith(\"this.microseconds\", \"kotlin.time.Duration.Companion.microseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Long.microseconds get() = toDuration(DurationUnit.MICROSECONDS)\n\n/**\n * Returns a [Duration] equal to this [Double] number of microseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Double.microseconds' extension property from Duration.Companion instead.\", ReplaceWith(\"this.microseconds\", \"kotlin.time.Duration.Companion.microseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Double.microseconds get() = toDuration(DurationUnit.MICROSECONDS)\n\n\n/** Returns a [Duration] equal to this [Int] number of milliseconds. */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Int.milliseconds' extension property from Duration.Companion instead.\", ReplaceWith(\"this.milliseconds\", \"kotlin.time.Duration.Companion.milliseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Int.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n\n/** Returns a [Duration] equal to this [Long] number of milliseconds. */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Long.milliseconds' extension property from Duration.Companion instead.\", ReplaceWith(\"this.milliseconds\", \"kotlin.time.Duration.Companion.milliseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Long.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n\n/**\n * Returns a [Duration] equal to this [Double] number of milliseconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Double.milliseconds' extension property from Duration.Companion instead.\", ReplaceWith(\"this.milliseconds\", \"kotlin.time.Duration.Companion.milliseconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Double.milliseconds get() = toDuration(DurationUnit.MILLISECONDS)\n\n\n/** Returns a [Duration] equal to this [Int] number of seconds. */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Int.seconds' extension property from Duration.Companion instead.\", ReplaceWith(\"this.seconds\", \"kotlin.time.Duration.Companion.seconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Int.seconds get() = toDuration(DurationUnit.SECONDS)\n\n/** Returns a [Duration] equal to this [Long] number of seconds. */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Long.seconds' extension property from Duration.Companion instead.\", ReplaceWith(\"this.seconds\", \"kotlin.time.Duration.Companion.seconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Long.seconds get() = toDuration(DurationUnit.SECONDS)\n\n/**\n * Returns a [Duration] equal to this\n [Double] number of seconds.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Double.seconds' extension property from Duration.Companion instead.\", ReplaceWith(\"this.seconds\", \"kotlin.time.Duration.Companion.seconds\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Double.seconds get() = toDuration(DurationUnit.SECONDS)\n\n\n/** Returns a [Duration] equal to this [Int] number of minutes. */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Int.minutes' extension property from Duration.Companion instead.\", ReplaceWith(\"this.minutes\", \"kotlin.time.Duration.Companion.minutes\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince =

\"1.8\")\npublic val Int.minutes get() = toDuration(DurationUnit.MINUTES)\n\n/** Returns a [Duration] equal to this [Long] number of minutes. */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Long.minutes' extension property from Duration.Companion instead.\", ReplaceWith(\"this.minutes\", \"kotlin.time.Duration.Companion.minutes\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Long.minutes get() = toDuration(DurationUnit.MINUTES)\n\n/**\n * Returns a [Duration] equal to this [Double] number of minutes.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Double.minutes' extension property from Duration.Companion instead.\", ReplaceWith(\"this.minutes\", \"kotlin.time.Duration.Companion.minutes\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Double.minutes get() = toDuration(DurationUnit.MINUTES)\n\n\n/** Returns a [Duration] equal to this [Int] number of hours. */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Int.hours' extension property from Duration.Companion instead.\", ReplaceWith(\"this.hours\", \"kotlin.time.Duration.Companion.hours\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Int.hours get() = toDuration(DurationUnit.HOURS)\n\n/** Returns a [Duration] equal to this [Long] number of hours. */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Long.hours' extension property from Duration.Companion instead.\", ReplaceWith(\"this.hours\", \"kotlin.time.Duration.Companion.hours\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Long.hours get() = toDuration(DurationUnit.HOURS)\n\n/**\n * Returns a [Duration] equal to this [Double] number of hours.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Double.hours' extension property from Duration.Companion instead.\", ReplaceWith(\"this.hours\", \"kotlin.time.Duration.Companion.hours\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic
val Double.hours get() = toDuration(DurationUnit.HOURS)\n\n\n/** Returns a [Duration] equal to this [Int] number of days. */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Int.days' extension property from Duration.Companion instead.\", ReplaceWith(\"this.days\", \"kotlin.time.Duration.Companion.days\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Int.days get() = toDuration(DurationUnit.DAYS)\n\n/** Returns a [Duration] equal to this [Long] number of days. */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Long.days' extension property from Duration.Companion instead.\", ReplaceWith(\"this.days\", \"kotlin.time.Duration.Companion.days\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Long.days get() = toDuration(DurationUnit.DAYS)\n\n/**\n * Returns a [Duration] equal to this [Double] number of days.\n *\n * @throws IllegalArgumentException if this [Double] value is `NaN`.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Use 'Double.days' extension property from Duration.Companion instead.\", ReplaceWith(\"this.days\", \"kotlin.time.Duration.Companion.days\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\", errorSince = \"1.8\")\npublic val Double.days get() = toDuration(DurationUnit.DAYS)\n\n\n/** Returns a duration whose value is the specified [duration] value multiplied by this number.
*/\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalTime::class)\n@kotlin.internal.InlineOnly\npublic inline operator fun Int.times(duration: Duration): Duration = duration * this\n\n/**\n * Returns a duration whose value is the specified [duration] value multiplied by this number.\n *\n * The operation may involve rounding when the result cannot be represented exactly with a [Double] number.\n *\n * @throws IllegalArgumentException if the operation results in a `NaN` value.\n
*/\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalTime::class)\n@kotlin.internal.InlineOnly\npublic
inline operator fun Double.times(duration: Duration): Duration = duration * this\n\n\n\nprivate fun parseDuration(value: String, strictIso: Boolean): Duration {\n   var length = value.length\n   if (length == 0) throw IllegalArgumentException(\"The string is empty\")\n   var index = 0\n   var result = Duration.ZERO\n   val infinityString = \"Infinity\"\n   when (value[index]) {\n   '+', '-' -> index++\n   }\n   val hasSign = index > 0\n

```
val isNegative = hasSign && value.startsWith('-')\n    when {\n        length <= index ->\n            throw IllegalArgumentException(\"No components\")\n        value[index] == 'P' -> {\n            if (++index == length) throw IllegalArgumentException()\n            val nonDigitSymbols = \"+-.\"\n            var isTimeComponent = false\n            var prevUnit: DurationUnit? = null\n            while (index < length) {\n                if (value[index] == 'T') {\n                    if (isTimeComponent\n                        || ++index == length) throw IllegalArgumentException()\n                    isTimeComponent = true\n                    continue\n                }\n                val component = value.substringWhile(index) { it in '0'..'9' || it in nonDigitSymbols }\n                if (component.isEmpty()) throw IllegalArgumentException()\n                index += component.length\n                val unitChar = value.getOrElse(index) { throw IllegalArgumentException(\"Missing unit for value $component\") }\n                index++\n                val unit = durationUnitByIsoChar(unitChar, isTimeComponent)\n                if (prevUnit != null && prevUnit <= unit) throw IllegalArgumentException(\"Unexpected order of duration components\")\n                prevUnit = unit\n                val dotIndex = component.indexOf('.')\n                if (unit == DurationUnit.SECONDS && dotIndex > 0) {\n                    val whole = component.substring(0, dotIndex)\n                    result += parseOverLongIsoComponent(whole).toDuration(unit)\n                    result += component.substring(dotIndex).toDouble().toDuration(unit)\n                } else {\n                    result += parseOverLongIsoComponent(component).toDuration(unit)\n                }\n            }\n        }\n        strictIso ->\n            throw IllegalArgumentException()\n        value.regionMatches(index, infinityString, 0, length = maxOf(length - index, infinityString.length), ignoreCase = true) -> {\n            result = Duration.INFINITE\n        }\n        else -> {\n            // parse default string format\n            var prevUnit: DurationUnit? = null\n            var afterFirst = false\n            var allowSpaces = !hasSign\n            if (hasSign && value[index] == '(' && value.last() == ')') {\n                allowSpaces = true\n                if (++index == --length) throw IllegalArgumentException(\"No components\")\n            }\n            while (index < length) {\n                if (afterFirst && allowSpaces) {\n                    index = value.skipWhile(index) { it == ' ' }\n                }\n                afterFirst = true\n                val component = value.substringWhile(index) { it in '0'..'9' || it == '.' }\n                if (component.isEmpty()) throw IllegalArgumentException()\n                index += component.length\n                val unitName = value.substringWhile(index) { it in 'a'..'z' }\n                index += unitName.length\n                val unit = durationUnitByShortName(unitName)\n                if (prevUnit != null && prevUnit <= unit) throw IllegalArgumentException(\"Unexpected order of duration components\")\n                prevUnit = unit\n                val dotIndex = component.indexOf('.')\n                if (dotIndex > 0) {\n                    val whole = component.substring(0, dotIndex)\n                    result += whole.toLong().toDuration(unit)\n                    result += component.substring(dotIndex).toDouble().toDuration(unit)\n                    if (index < length) throw IllegalArgumentException(\"Fractional component must be last\")\n                } else {\n                    result += component.toLong().toDuration(unit)\n                }\n            }\n        }\n    }\n    return if (isNegative) -result else result\n}\n\n\nprivate fun parseOverLongIsoComponent(value: String): Long {\n    val length = value.length\n    var startIndex = 0\n    if (length > 0 && value[0] in \"+-\") startIndex++\n    if ((length - startIndex) > 16 && (startIndex..value.lastIndex).all { value[it] in '0'..'9' }) {\n        // all chars are digits, but more than ceiling(log10(MAX_MILLIS / 1000)) of them\n        return if (value[0] == '-') Long.MIN_VALUE else Long.MAX_VALUE\n    }\n    // TODO: replace with just toLong after min JDK becomes 8\n    return if (value.startsWith(\"+\")) value.drop(1).toLong() else value.toLong()\n}\n\n\n\nprivate inline fun String.substringWhile(startIndex: Int, predicate: (Char) -> Boolean): String =\n    substring(startIndex, skipWhile(startIndex, predicate))\n\nprivate inline fun String.skipWhile(startIndex: Int, predicate: (Char) -> Boolean): Int {\n    var i = startIndex\n    while (i < length && predicate(this[i])) i++\n    return i\n}\n\n\n\n\n\n// The ranges are chosen so that they are:\n// - symmetric relative to zero: this greatly simplifies operations with sign, e.g. unaryMinus and minus.\n// - non-overlapping, but adjacent: the first value that doesn't fit in nanos range, can be exactly represented in millis.\n\ninternal const val NANOS_IN_MILLIS = 1_000_000\n// maximum number duration can store in nanosecond range\ninternal const val MAX_NANOS = Long.MAX_VALUE / 2 / NANOS_IN_MILLIS * NANOS_IN_MILLIS - 1 // ends in ..._999_999\n// maximum number duration can store in millisecond range, also encodes an infinite value\ninternal
```

const val MAX_MILLIS = Long.MAX_VALUE / 2\n// MAX_NANOS expressed in milliseconds\nprivate const val MAX_NANOS_IN_MILLIS
= MAX_NANOS / NANOS_IN_MILLIS\n\nprivate fun nanosToMillis(nanos: Long): Long = nanos / NANOS_IN_MILLIS\nprivate fun millisToNanos(millis: Long): Long = millis * NANOS_IN_MILLIS\n\nprivate fun durationOfNanos(normalNanos: Long) = Duration(normalNanos shl 1)\nprivate fun durationOfMillis(normalMillis: Long) = Duration((normalMillis shl 1) + 1)\nprivate fun durationOf(normalValue: Long, unitDiscriminator: Int) = Duration((normalValue shl 1) + unitDiscriminator)\nprivate fun durationOfNanosNormalized(nanos: Long) =\n if (nanos in -MAX_NANOS..MAX_NANOS) {\n durationOfNanos(nanos)\n } else {\n durationOfMillis(nanosToMillis(nanos))\n }\n\nprivate fun durationOfMillisNormalized(millis: Long) =\n if (millis in -MAX_NANOS_IN_MILLIS..MAX_NANOS_IN_MILLIS) {\n durationOfNanos(millisToNanos(millis))\n } else {\n durationOfMillis(millis.coerceIn(-MAX_MILLIS, MAX_MILLIS))\n }\n\ninternal expect val durationAssertionsEnabled: Boolean\n\ninternal expect
fun formatToExactDecimals(value: Double, decimals: Int): String\ninternal expect fun formatUpToDecimals(value: Double, decimals: Int): String\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n@file:kotlin.jvm.JvmName(\"UnsignedKt\")\npackage kotlin\n\n@PublishedApi\ninternal fun uintCompare(v1: Int, v2: Int): Int = (v1 xor Int.MIN_VALUE).compareTo(v2 xor Int.MIN_VALUE)\n@PublishedApi\ninternal fun ulongCompare(v1: Long, v2: Long): Int = (v1 xor Long.MIN_VALUE).compareTo(v2 xor Long.MIN_VALUE)\n\n@PublishedApi\ninternal fun uintDivide(v1: UInt, v2: UInt): UInt = (v1.toLong() / v2.toLong()).toUInt()\n@PublishedApi\ninternal fun uintRemainder(v1: UInt, v2: UInt): UInt = (v1.toLong() % v2.toLong()).toUInt()\n\n// Division and remainder are based on Guava's UnsignedLongs implementation\n// Copyright 2011 The Guava Authors\n\n@PublishedApi\ninternal
fun ulongDivide(v1: ULong, v2: ULong): ULong {\n val dividend = v1.toLong()\n val divisor = v2.toLong()\n if (divisor < 0) { // i.e., divisor >= 2^63:\n return if (v1 < v2) ULong(0) else ULong(1)\n }\n\n // Optimization - use signed division if both dividend and divisor < 2^63\n if (dividend >= 0) {\n return ULong(dividend / divisor)\n }\n\n // Otherwise, approximate the quotient, check, and correct if necessary.\n val quotient = ((dividend ushr 1) / divisor) shl 1\n val rem = dividend - quotient * divisor\n return ULong(quotient + if (ULong(rem) >= ULong(divisor)) 1 else 0)\n\n}\n\n@PublishedApi\ninternal fun ulongRemainder(v1: ULong, v2: ULong): ULong {\n val dividend = v1.toLong()\n val divisor = v2.toLong()\n if (divisor < 0) { // i.e., divisor >= 2^63:\n return if (v1 < v2) {\n v1 // dividend < divisor\n } else {\n v1 - v2 // dividend >= divisor\n }\n }\n\n
// Optimization - use signed modulus if both dividend and divisor < 2^63\n if (dividend >= 0) {\n return ULong(dividend % divisor)\n }\n\n // Otherwise, approximate the quotient, check, and correct if necessary.\n val quotient = ((dividend ushr 1) / divisor) shl 1\n val rem = dividend - quotient * divisor\n return ULong(rem - if (ULong(rem) >= ULong(divisor)) divisor else 0)\n}\n\n@PublishedApi\ninternal fun doubleToUInt(v: Double): UInt = when {\n v.isNaN() -> 0u\n v <= UInt.MIN_VALUE.toDouble() -> UInt.MIN_VALUE\n v >= UInt.MAX_VALUE.toDouble() -> UInt.MAX_VALUE\n v <= Int.MAX_VALUE -> v.toInt().toUInt()\n else -> (v - Int.MAX_VALUE).toInt().toUInt() + Int.MAX_VALUE.toUInt() // Int.MAX_VALUE < v < UInt.MAX_VALUE\n}\n\n@PublishedApi\ninternal fun doubleToULong(v: Double): ULong = when {\n v.isNaN() -> 0u\n v <= ULong.MIN_VALUE.toDouble() -> ULong.MIN_VALUE\n v >= ULong.MAX_VALUE.toDouble() -> ULong.MAX_VALUE\n v < Long.MAX_VALUE
-> v.toLong().toULong()\n\n // Real values from Long.MAX_VALUE to (Long.MAX_VALUE + 1) are not representable in Double, so don't handle them.\n else -> (v - 9223372036854775808.0).toLong().toULong() + 9223372036854775808uL // Long.MAX_VALUE + 1 < v < ULong.MAX_VALUE\n}\n\n\n@PublishedApi\ninternal fun uintToDouble(v: Int): Double = (v and Int.MAX_VALUE).toDouble() + (v ushr 31 shl 30).toDouble() * 2\n\n@PublishedApi\ninternal fun

ulongToDouble(v: Long): Double = (v ushr 11).toDouble() * 2048 + (v and 2047)\n\n\ninternal fun ulongToString(v: Long): String = ulongToString(v, 10)\n\ninternal fun ulongToString(v: Long, base: Int): String {\n if (v >= 0) return v.toString(base)\n\n var quotient = ((v ushr 1) / base) shl 1\n var rem = v - quotient * base\n if (rem >= base) {\n rem -= base\n quotient += 1\n }\n return quotient.toString(base) + rem.toString(base)\n}\n\n","/*\n * Copyright 2010-2016 JetBrains s.r.o.\n *\n * Licensed under the Apache License, Version 2.0 (the \"License\");\n * you may not use this file except in compliance with the License.\n * You may obtain a copy of the License at\n *\n * http://www.apache.org/licenses/LICENSE-2.0\n *\n * Unless required by applicable law or agreed to in writing, software\n * distributed under the License is distributed on an \"AS IS\" BASIS,\n * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.\n * See the License for the specific language governing permissions and\n * limitations under the License.\n */\n\npackage kotlin.internal\n\n/**\n * Specifies that the corresponding type parameter is not used for unsafe operations such as casts or 'is' checks\n * That means it's completely safe to use generic types as argument for such parameter.\n */\n@Target(AnnotationTarget.TYPE_PARAMETER)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class PureReifiable\n\n/**\n * Specifies that the corresponding built-in method exists depending on platform.\n * Current implementation for JVM looks whether method with same JVM descriptor exists in the module JDK.\n * For example MutableMap.remove(K, V) available only if corresponding\n * method 'java/util/Map.remove(Ljava/lang/Object;Ljava/lang/Object;)Z' is defined in JDK (i.e. for major versions >= 8)\n */\n@Target(AnnotationTarget.FUNCTION)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class PlatformDependent\n\n/**\n * When applied to a function or property, enables a compiler optimization that evaluates that function or property\n * at compile-time and replaces calls to it with the computed result.\n */\n@Target(AnnotationTarget.CONSTRUCTOR, AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY)\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin(\"1.7\")\ninternal annotation class IntrinsicConstEvaluation\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"CollectionsKt\")\n\npackage kotlin.collections\n\n/**\n * Given an [iterator] function constructs an [Iterable] instance that returns values through the [Iterator]\n * provided by that function.\n * @sample samples.collections.Iterables.Building.iterable\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable(crossinline iterator: () -> Iterator<T>): Iterable<T> = object : Iterable<T> {\n override fun iterator(): Iterator<T> = iterator()\n}\n\n/**\n * A wrapper over another [Iterable] (or any other object that can produce an [Iterator]) that returns\n * an indexing iterator.\n */\ninternal class IndexingIterable<out T>(private val iteratorFactory: () -> Iterator<T>) : Iterable<IndexedValue<T>> {\n override fun iterator(): Iterator<IndexedValue<T>> = IndexingIterator(iteratorFactory())\n}\n\n\n/**\n * Returns the size of this iterable if it is known, or `null` otherwise.\n */\n@PublishedApi\ninternal fun <T> Iterable<T>.collectionSizeOrNull(): Int? = if (this is Collection<*>) this.size else null\n\n/**\n * Returns the size of this iterable if it is known, or the specified [default] value otherwise.\n */\n@PublishedApi\ninternal fun <T> Iterable<T>.collectionSizeOrDefault(default: Int): Int = if (this is Collection<*>) this.size else default\n\n\n/**\n * Returns a single list of all elements from all collections in the given collection.\n * @sample samples.collections.Iterables.Operations.flattenIterable\n */\npublic fun <T> Iterable<Iterable<T>>.flatten(): List<T> {\n val result = ArrayList<T>()\n for (element in this) {\n result.addAll(element)\n }\n return result\n}\n\n/**\n * Returns a pair of lists, where\n * *first* list is built from the first values of each pair from this collection,\n * *second* list is built from the second values of each pair from this collection.\n * @sample samples.collections.Iterables.Operations.unzipIterable\n */\npublic fun <T, R> Iterable<Pair<T, R>>.unzip(): Pair<List<T>, List<R>> {\n val expectedSize = collectionSizeOrDefault(10)\n val listT = ArrayList<T>(expectedSize)\n val listR = ArrayList<R>(expectedSize)\n for (pair in this) {\n listT.add(pair.first)\n listR.add(pair.second)\n }\n

return listT to listR\n}\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"SequencesKt\")\n\npackage kotlin.sequences\n\nimport kotlin.random.Random\n\n/**\n * Given an [iterator] function constructs a [Sequence] that returns values through the [Iterator]\n * provided by that function.\n * The values are evaluated lazily, and the sequence is potentially infinite.\n *\n * @sample samples.collections.Sequences.Building.sequenceFromIterator\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence(crossinline iterator: () -> Iterator<T>): Sequence<T> = object : Sequence<T> {\n override fun iterator(): Iterator<T> = iterator()\n}\n\n/**\n * Creates a sequence that returns all elements from this iterator. The sequence is constrained to be iterated only once.\n *\n * @sample samples.collections.Sequences.Building.sequenceFromIterator\n */\npublic fun <T> Iterator<T>.asSequence(): Sequence<T> = Sequence { this }.constrainOnce()\n\n/**\n * Creates a sequence that returns the specified values.\n *\n * @sample samples.collections.Sequences.Building.sequenceOfValues\n */\npublic fun <T> sequenceOf(vararg elements: T): Sequence<T> = if (elements.isEmpty()) emptySequence() else elements.asSequence()\n\n/**\n * Returns an empty sequence.\n */\npublic fun <T> emptySequence(): Sequence<T> = EmptySequence\n\nprivate object EmptySequence : Sequence<Nothing>, DropTakeSequence<Nothing> {\n override fun iterator(): Iterator<Nothing> = EmptyIterator\n override fun drop(n: Int) = EmptySequence\n override fun take(n: Int) = EmptySequence\n}\n\n/**\n * Returns this sequence if it's not `null` and the empty sequence otherwise.\n * @sample samples.collections.Sequences.Usage.sequenceOrEmpty\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>?.orEmpty(): Sequence<T> = this ?: emptySequence()\n\n\n/**\n * Returns a sequence that iterates through the elements either of this sequence\n * or, if this sequence turns out to be empty, of the sequence returned by [defaultValue] function.\n *\n * @sample samples.collections.Sequences.Usage.sequenceIfEmpty\n */\n@SinceKotlin(\"1.3\")\npublic fun <T> Sequence<T>.ifEmpty(defaultValue: () -> Sequence<T>): Sequence<T> = sequence {\n val iterator = this@ifEmpty.iterator()\n if (iterator.hasNext()) {\n yieldAll(iterator)\n } else {\n yieldAll(defaultValue())\n }\n}\n\n/**\n * Returns a sequence of all elements from all sequences in this sequence.\n *\n * The operation is _intermediate_ and _stateless_.\n *\n * @sample samples.collections.Sequences.Transformations.flattenSequenceOfSequences\n */\npublic fun <T> Sequence<Sequence<T>>.flatten(): Sequence<T> = flatten { it.iterator() }\n\n/**\n * Returns a sequence of all elements from all iterables in this sequence.\n *\n * The operation is _intermediate_ and _stateless_.\n *\n * @sample samples.collections.Sequences.Transformations.flattenSequenceOfLists\n */\n@kotlin.jvm.JvmName(\"flattenSequenceOfIterable\")\npublic fun <T> Sequence<Iterable<T>>.flatten(): Sequence<T> = flatten { it.iterator() }\n\nprivate fun <T, R> Sequence<T>.flatten(iterator: (T) -> Iterator<R>): Sequence<R> {\n if (this is TransformingSequence<*, *>) {\n return (this as TransformingSequence<*, T>).flatten(iterator)\n }\n return FlatteningSequence(this, { it }, iterator)\n}\n\n/**\n * Returns a pair of lists, where\n * *first* list is built from the first values of each pair from this sequence,\n * *second* list is built from the second values of each pair from this sequence.\n *\n * The operation is _terminal_.\n *\n * @sample samples.collections.Sequences.Transformations.unzip\n */\npublic fun <T, R> Sequence<Pair<T, R>>.unzip(): Pair<List<T>, List<R>> {\n val listT = ArrayList<T>()\n val listR = ArrayList<R>()\n for (pair in this) {\n listT.add(pair.first)\n listR.add(pair.second)\n }\n return listT to listR\n}\n\n/**\n * Returns a sequence that yields elements of this sequence randomly shuffled.\n *\n * Note that every iteration of the sequence returns elements in a different order.\n *\n * The operation is _intermediate_ and _stateful_.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T> Sequence<T>.shuffled(): Sequence<T> = shuffled(Random)\n\n/**\n * Returns a sequence that yields elements of this sequence randomly shuffled\n * using the specified [random] instance as the source of randomness.\n

```
 *\n * Note that every iteration of the sequence returns elements in a different order.\n *\n * The operation is
_intermediate_ and _stateful_.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T> Sequence<T>.shuffled(random:
Random): Sequence<T> = sequence<T> {\n    val buffer = toMutableList()\n    while (buffer.isNotEmpty()) {\n
val j = random.nextInt(buffer.size)\n        val last = buffer.removeLast()\n        val value = if (j < buffer.size)
buffer.set(j, last) else last\n        yield(value)\n    }\n}\n\n\n/**\n * A sequence that returns the values from the
underlying [sequence] that either match or do not match\n * the specified [predicate].\n *\n * @param sendWhen If
`true`, values for which the predicate returns `true` are returned. Otherwise,\n * values for which the predicate
returns `false` are returned\n */\ninternal class FilteringSequence<T>(\n    private val sequence: Sequence<T>,\n
private val sendWhen: Boolean = true,\n    private val predicate: (T) -> Boolean\n)
 : Sequence<T> {\n\n    override fun iterator(): Iterator<T> = object : Iterator<T> {\n        val iterator =
sequence.iterator()\n        var nextState: Int = -1 // -1 for unknown, 0 for done, 1 for continue\n        var nextItem: T?
= null\n\n        private fun calcNext() {\n            while (iterator.hasNext()) {\n                val item = iterator.next()\n
            if (predicate(item) == sendWhen) {\n                    nextItem = item\n                    nextState = 1\n
return\n                }\n            }\n            nextState = 0\n        }\n\n        override fun next(): T {\n            if (nextState
== -1)\n                calcNext()\n            if (nextState == 0)\n                throw NoSuchElementException()\n            val
result = nextItem\n            nextItem = null\n            nextState = -1\n            @Suppress(\"UNCHECKED_CAST\")\n
            return result as T\n        }\n\n        override fun hasNext(): Boolean {\n
            if (nextState == -1)\n                calcNext()\n            return nextState == 1\n        }\n    }\n}\n\n/**\n * A
sequence which returns the results of applying the given [transformer] function to the values\n * in the underlying
[sequence].\n */\n\ninternal class TransformingSequence<T, R>\nconstructor(private val sequence: Sequence<T>,
private val transformer: (T) -> R) : Sequence<R> {\n    override fun iterator(): Iterator<R> = object : Iterator<R> {\n
        val iterator = sequence.iterator()\n        override fun next(): R {\n            return transformer(iterator.next())\n
}\n\n        override fun hasNext(): Boolean {\n            return iterator.hasNext()\n        }\n    }\n\n    internal fun <E>
flatten(iterator: (R) -> Iterator<E>): Sequence<E> {\n        return FlatteningSequence<T, R, E>(sequence,
transformer, iterator)\n    }\n}\n\n/**\n * A sequence which returns the results of applying the given [transformer]
function to the values\n * in the underlying
 [sequence], where the transformer function takes the index of the value in the underlying\n * sequence along with
the value itself.\n */\ninternal class TransformingIndexedSequence<T, R>\nconstructor(private val sequence:
Sequence<T>, private val transformer: (Int, T) -> R) : Sequence<R> {\n    override fun iterator(): Iterator<R> =
object : Iterator<R> {\n        val iterator = sequence.iterator()\n        var index = 0\n        override fun next(): R {\n
        return transformer(checkIndexOverflow(index++), iterator.next())\n        }\n\n        override fun hasNext():
Boolean {\n            return iterator.hasNext()\n        }\n    }\n}\n\n/**\n * A sequence which combines values from
the underlying [sequence] with their indices and returns them as\n * [IndexedValue] objects.\n */\ninternal class
IndexingSequence<T>\nconstructor(private val sequence: Sequence<T>) : Sequence<IndexedValue<T>> {\n
override fun iterator(): Iterator<IndexedValue<T>> = object : Iterator<IndexedValue<T>>
 {\n        val iterator = sequence.iterator()\n        var index = 0\n        override fun next(): IndexedValue<T> {\n
return IndexedValue(checkIndexOverflow(index++), iterator.next())\n        }\n\n        override fun hasNext():
Boolean {\n            return iterator.hasNext()\n        }\n    }\n}\n\n/**\n * A sequence which takes the values from
two parallel underlying sequences, passes them to the given\n * [transform] function and returns the values returned
by that function. The sequence stops returning\n * values as soon as one of the underlying sequences stops returning
values.\n */\ninternal class MergingSequence<T1, T2, V>\nconstructor(\n    private val sequence1:
Sequence<T1>,\n    private val sequence2: Sequence<T2>,\n    private val transform: (T1, T2) -> V\n) :
Sequence<V> {\n    override fun iterator(): Iterator<V> = object : Iterator<V> {\n        val iterator1 =
sequence1.iterator()\n        val iterator2 = sequence2.iterator()\n        override
 fun next(): V {\n            return transform(iterator1.next(), iterator2.next())\n        }\n\n        override fun hasNext():
Boolean {\n            return iterator1.hasNext() && iterator2.hasNext()\n        }\n    }\n}\n\ninternal class
FlatteningSequence<T, R, E>\nconstructor(\n    private val sequence: Sequence<T>,\n    private val transformer: (T)
-> R,\n    private val iterator: (R) -> Iterator<E>\n) : Sequence<E> {\n    override fun iterator(): Iterator<E> = object
```

```
: Iterator<E> {\n        val iterator = sequence.iterator()\n        var itemIterator: Iterator<E>? = null\n\n        override fun next(): E {\n            if (!ensureItemIterator())\n                throw NoSuchElementException()\n            return itemIterator!!.next()\n        }\n\n        override fun hasNext(): Boolean {\n            return ensureItemIterator()\n        }\n\n        private fun ensureItemIterator(): Boolean {\n            if (itemIterator?.hasNext() == false)\n                itemIterator = null\n\n            while (itemIterator == null) {\n                if (!iterator.hasNext()) {\n                    return false\n                } else {\n                    val element = iterator.next()\n                    val nextItemIterator = iterator(transformer(element))\n                    if (nextItemIterator.hasNext()) {\n                        itemIterator = nextItemIterator\n                        return true\n                    }\n                }\n            }\n            return true\n        }\n    }\n}\n\ninternal fun <T, C, R> flatMapIndexed(source: Sequence<T>, transform: (Int, T) -> C, iterator: (C) -> Iterator<R>): Sequence<R> =\n    sequence {\n        var index = 0\n        for (element in source) {\n            val result = transform(checkIndexOverflow(index++), element)\n            yieldAll(iterator(result))\n        }\n    }\n\n/**\n * A sequence that supports drop(n) and take(n) operations\n */\ninternal interface DropTakeSequence<T> : Sequence<T> {\n
    fun drop(n: Int): Sequence<T>\n    fun take(n: Int): Sequence<T>\n}\n\n/**\n * A sequence that skips [startIndex] values from the underlying [sequence]\n * and stops returning values right before [endIndex], i.e. stops at `endIndex - 1`\n */\ninternal class SubSequence<T>(\n    private val sequence: Sequence<T>,\n    private val startIndex: Int,\n    private val endIndex: Int\n) : Sequence<T>, DropTakeSequence<T> {\n\n    init {\n        require(startIndex >= 0) { \"startIndex should be non-negative, but is $startIndex\" }\n        require(endIndex >= 0) { \"endIndex should be non-negative, but is $endIndex\" }\n        require(endIndex >= startIndex) { \"endIndex should be not less than startIndex, but was $endIndex < $startIndex\" }\n    }\n\n    private val count: Int get() = endIndex - startIndex\n\n    override fun drop(n: Int): Sequence<T> = if (n >= count) emptySequence() else SubSequence(sequence, startIndex + n, endIndex)\n    override fun take(n: Int): Sequence<T> = if (n >= count) this else SubSequence(sequence, startIndex, startIndex + n)\n\n    override fun iterator() = object : Iterator<T> {\n\n        val iterator = sequence.iterator()\n        var position = 0\n\n        // Shouldn't be called from constructor to avoid premature iteration\n        private fun drop() {\n            while (position < startIndex && iterator.hasNext()) {\n                iterator.next()\n                position++\n            }\n        }\n\n        override fun hasNext(): Boolean {\n            drop()\n            return (position < endIndex) && iterator.hasNext()\n        }\n\n        override fun next(): T {\n            drop()\n            if (position >= endIndex)\n                throw NoSuchElementException()\n            position++\n            return iterator.next()\n        }\n    }\n}\n\n/**\n * A sequence that returns at most [count] values from the underlying [sequence], and stops returning values\n * as soon as that count is reached.\n */\ninternal class TakeSequence<T>(\n    private val sequence: Sequence<T>,\n    private val count: Int\n) : Sequence<T>, DropTakeSequence<T> {\n\n    init {\n        require(count >= 0) { \"count must be non-negative, but was $count.\" }\n    }\n\n    override fun drop(n: Int): Sequence<T> = if (n >= count) emptySequence() else SubSequence(sequence, n, count)\n    override fun take(n: Int): Sequence<T> = if (n >= count) this else TakeSequence(sequence, n)\n\n    override fun iterator(): Iterator<T> = object : Iterator<T> {\n        var left = count\n        val iterator = sequence.iterator()\n\n        override fun next(): T {\n            if (left == 0)\n                throw NoSuchElementException()\n            left--\n            return iterator.next()\n        }\n\n        override fun hasNext(): Boolean {\n            return left > 0 && iterator.hasNext()\n        }\n    }\n}\n\n/**\n * A sequence that returns values from the underlying [sequence] while the [predicate] function returns\n * `true`, and stops returning values once the function returns `false` for the next element.\n */\ninternal class TakeWhileSequence<T>\nconstructor(\n    private val sequence: Sequence<T>,\n    private val predicate: (T) -> Boolean\n) : Sequence<T> {\n    override fun iterator(): Iterator<T> = object : Iterator<T> {\n        val iterator = sequence.iterator()\n        var nextState: Int = -1 // -1 for unknown, 0 for done, 1 for continue\n        var nextItem: T? = null\n\n        private fun calcNext() {\n            if (iterator.hasNext()) {\n                val item = iterator.next()\n                if (predicate(item)) {\n                    nextState = 1\n                    nextItem = item\n                    return\n                }\n            }\n            nextState = 0\n        }\n\n        override fun next(): T {\n            if (nextState == -1)\n
```

```
calcNext() // will change nextState\n            if (nextState == 0)\n                throw NoSuchElementException()\n            @Suppress(\"UNCHECKED_CAST\")\n            val result = nextItem as T\n\n            // Clean next to avoid keeping reference on yielded instance\n            nextItem = null\n            nextState = -1\n            return result\n        }\n\n        override fun hasNext(): Boolean {\n            if (nextState == -1)\n                calcNext() // will change nextState\n            return nextState == 1\n        }\n    }\n}\n\n/**\n * A sequence that skips the specified number of values from the underlying [sequence] and returns\n * all values after that.\n */\ninternal class DropSequence<T>(\n  private val sequence: Sequence<T>,\n    private val count: Int\n) : Sequence<T>, DropTakeSequence<T> {\n    init {\n        require(count >= 0) { \"count must be non-negative, but was $count.\" }\n    }\n\n    override fun drop(n: Int): Sequence<T> = (count + n).let { n1 -> if (n1 < 0) DropSequence(this, n) else DropSequence(sequence, n1) }\n    override fun take(n: Int): Sequence<T>
    = (count + n).let { n1 -> if (n1 < 0) TakeSequence(this, n) else SubSequence(sequence, count, n1) }\n\n    override fun iterator(): Iterator<T> = object : Iterator<T> {\n        val iterator = sequence.iterator()\n        var left = count\n\n        // Shouldn't be called from constructor to avoid premature iteration\n        private fun drop() {\n            while (left > 0 && iterator.hasNext()) {\n                iterator.next()\n                left--\n            }\n        }\n\n        override fun next(): T {\n            drop()\n            return iterator.next()\n        }\n\n        override fun hasNext(): Boolean {\n            drop()\n            return iterator.hasNext()\n        }\n    }\n}\n\n/**\n * A sequence that skips the values from the underlying [sequence] while the given [predicate] returns `true` and returns\n * all values after that.\n */\ninternal class DropWhileSequence<T>\nconstructor(\n    private val sequence: Sequence<T>,\n    private val predicate: (T) -> Boolean\n) : Sequence<T> {\n\n    override fun iterator(): Iterator<T> = object : Iterator<T> {\n        val iterator = sequence.iterator()\n        var dropState: Int = -1 // -1 for not dropping, 1 for nextItem, 0 for normal iteration\n        var nextItem: T? = null\n\n        private fun drop() {\n            while (iterator.hasNext()) {\n                val item = iterator.next()\n                if (!predicate(item)) {\n                    nextItem = item\n                    dropState = 1\n                    return\n                }\n            }\n            dropState = 0\n        }\n\n        override fun next(): T {\n            if (dropState == -1)\n                drop()\n\n            if (dropState == 1) {\n                @Suppress(\"UNCHECKED_CAST\")\n                val result = nextItem as T\n                nextItem = null\n                dropState = 0\n                return result\n            }\n            return iterator.next()\n        }\n\n        override fun hasNext(): Boolean {\n            if (dropState == -1)\n                drop()\n            return dropState == 1 || iterator.hasNext()\n        }\n    }\n}\n\ninternal class DistinctSequence<T, K>(private val source: Sequence<T>, private val keySelector: (T) -> K) : Sequence<T> {\n    override fun iterator(): Iterator<T> = DistinctIterator(source.iterator(), keySelector)\n}\n\nprivate class DistinctIterator<T, K>(private val source: Iterator<T>, private val keySelector: (T) -> K) : AbstractIterator<T>() {\n    private val observed = HashSet<K>()\n\n    override fun computeNext() {\n        while (source.hasNext()) {\n            val next = source.next()\n            val key = keySelector(next)\n\n            if (observed.add(key)) {\n                setNext(next)\n                return\n            }\n        }\n\n        done()\n    }\n}\n\n\nprivate class GeneratorSequence<T : Any>(private val getInitialValue: () -> T?, private val getNextValue: (T) -> T?) : Sequence<T>
{\n    override fun iterator(): Iterator<T> = object : Iterator<T> {\n        var nextItem: T? = null\n        var nextState: Int = -2 // -2 for initial unknown, -1 for next unknown, 0 for done, 1 for continue\n\n        private fun calcNext() {\n            nextItem = if (nextState == -2) getInitialValue() else getNextValue(nextItem!!)\n            nextState = if (nextItem == null) 0 else 1\n        }\n\n        override fun next(): T {\n            if (nextState < 0)\n                calcNext()\n\n            if (nextState == 0)\n                throw NoSuchElementException()\n            val result = nextItem as T\n            // Do not clean nextItem (to avoid keeping reference on yielded instance) -- need to keep state for getNextValue\n            nextState = -1\n            return result\n        }\n\n        override fun hasNext(): Boolean {\n            if (nextState < 0)\n                calcNext()\n            return nextState == 1\n        }\n    }\n}\n\n/**\n * Returns a
 wrapper sequence that provides values of this sequence, but ensures it can be iterated only one time.\n *\n * The operation is _intermediate_ and _stateless_.\n *\n * [IllegalStateException] is thrown on iterating the returned sequence for the second time and the following times.\n *\n */\npublic fun <T> Sequence<T>.constrainOnce(): Sequence<T> {\n    // as? does not work in js\n    //return this as? ConstrainedOnceSequence<T> ?:
```

```
ConstrainedOnceSequence(this)\n    return if (this is ConstrainedOnceSequence<T>) this else ConstrainedOnceSequence(this)\n}\n\n\n/**\n * Returns a sequence which invokes the function to calculate the next value on each iteration until the function returns `null`.\n *\n * The returned sequence is constrained to be iterated only once.\n *\n * @see constrainOnce\n * @see kotlin.sequences.sequence\n *\n * @sample samples.collections.Sequences.Building.generateSequence\n */\npublic fun <T : Any> generateSequence(nextFunction: () -> T?): Sequence<T> {\n    return GeneratorSequence(nextFunction, { nextFunction() }).constrainOnce()\n}\n\n/**\n * Returns a sequence defined by the starting value [seed] and the function [nextFunction],\n * which is invoked to calculate the next value based on the previous one on each iteration.\n *\n * The sequence produces values until it encounters first `null` value.\n * If [seed] is `null`, an empty sequence is produced.\n *\n * The sequence can be iterated multiple times, each time starting with [seed].\n *\n * @see kotlin.sequences.sequence\n *\n * @sample samples.collections.Sequences.Building.generateSequenceWithSeed\n */\n@kotlin.internal.LowPriorityInOverloadResolution\npublic fun <T : Any> generateSequence(seed: T?, nextFunction: (T) -> T?): Sequence<T> =\n    if (seed == null)\n        EmptySequence\n    else\n        GeneratorSequence({ seed }, nextFunction)\n\n/**\n * Returns a sequence defined by the function [seedFunction], which is invoked to produce the starting value,\n * and the [nextFunction], which is invoked to calculate the next value based on the previous one on each iteration.\n *\n * The sequence produces values until it encounters first `null` value.\n * If [seedFunction] returns `null`, an empty sequence is produced.\n *\n * The sequence can be iterated multiple times.\n *\n * @see kotlin.sequences.sequence\n *\n * @sample samples.collections.Sequences.Building.generateSequenceWithLazySeed\n */\npublic fun <T : Any> generateSequence(seedFunction: () -> T?, nextFunction: (T) -> T?): Sequence<T> =\n    GeneratorSequence(seedFunction, nextFunction)\n\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"PreconditionsKt\")\n\npackage kotlin\n\nimport kotlin.contracts.contract\n\n/**\n * Throws an [IllegalArgumentException] if the [value] is false.\n *\n * @sample samples.misc.Preconditions.failRequireWithLazyMessage\n */\n@kotlin.internal.InlineOnly\npublic inline fun require(value: Boolean): Unit {\n    contract {\n        returns() implies value\n    }\n    require(value) { \"Failed requirement.\" }\n}\n\n/**\n * Throws an [IllegalArgumentException] with the result of calling [lazyMessage] if the [value] is false.\n *\n * @sample samples.misc.Preconditions.failRequireWithLazyMessage\n */\n@kotlin.internal.InlineOnly\npublic inline fun require(value: Boolean, lazyMessage: () -> Any): Unit {\n    contract {\n        returns() implies value\n    }\n    if (!value) {\n        val message = lazyMessage()\n        throw IllegalArgumentException(message.toString())\n    }\n}\n\n/**\n * Throws an [IllegalArgumentException] if the [value] is null. Otherwise returns the not null value.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T : Any> requireNotNull(value: T?): T {\n    contract {\n        returns() implies (value != null)\n    }\n    return requireNotNull(value) { \"Required value was null.\" }\n}\n\n/**\n * Throws an [IllegalArgumentException] with the result of calling [lazyMessage] if the [value] is null. Otherwise\n * returns the not null value.\n *\n * @sample samples.misc.Preconditions.failRequireNotNullWithLazyMessage\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T : Any> requireNotNull(value: T?, lazyMessage: () -> Any): T {\n    contract {\n        returns() implies (value != null)\n    }\n\n    if (value == null) {\n        val message = lazyMessage()\n        throw IllegalArgumentException(message.toString())\n    } else {\n        return value\n    }\n}\n\n/**\n * Throws an [IllegalStateException] if the [value] is false.\n *\n * @sample samples.misc.Preconditions.failCheckWithLazyMessage\n */\n@kotlin.internal.InlineOnly\npublic inline fun check(value: Boolean): Unit {\n    contract {\n        returns() implies value\n    }\n    check(value) { \"Check failed.\" }\n}\n\n/**\n * Throws an [IllegalStateException] with the result of calling [lazyMessage] if the [value] is false.\n *\n * @sample samples.misc.Preconditions.failCheckWithLazyMessage\n */\n@kotlin.internal.InlineOnly\npublic inline fun
```

check(value: Boolean, lazyMessage: () -> Any): Unit {\n   contract {\n   returns() implies value\n   }\n   if (!value) {\n   val message = lazyMessage()\n   throw IllegalStateException(message.toString())\n   }\n}\n\n/**\n * Throws an [IllegalStateException] if the [value] is null. Otherwise\n * returns the not null value.\n *\n * @sample samples.misc.Preconditions.failCheckWithLazyMessage\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T : Any> checkNotNull(value: T?): T {\n   contract {\n   returns() implies (value != null)\n   }\n   return checkNotNull(value) { \"Required value was null.\" }\n}\n\n/**\n * Throws an [IllegalStateException] with the result of calling [lazyMessage] if the [value] is null. Otherwise\n * returns the not null value.\n *\n * @sample samples.misc.Preconditions.failCheckWithLazyMessage\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T : Any> checkNotNull(value: T?, lazyMessage: () -> Any): T {\n   contract {\n   returns() implies (value != null)\n   }\n\n   if (value == null) {\n   val message = lazyMessage()\n   throw IllegalStateException(message.toString())\n   } else {\n   return value\n   }\n}\n\n\n/**\n * Throws an [IllegalStateException] with the given [message].\n *\n * @sample samples.misc.Preconditions.failWithError\n */\n@kotlin.internal.InlineOnly\npublic inline fun error(message: Any): Nothing = throw IllegalStateException(message.toString())\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n//
 See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.js.*\nimport primitiveArrayConcat\nimport withType\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun <T> Array<out T>.elementAt(index: Int): T {\n   return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size\") }\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun ByteArray.elementAt(index: Int): Byte {\n   return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size\") }\n}\n\n/**\n * Returns an element
 at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun ShortArray.elementAt(index: Int): Short {\n   return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size\") }\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun IntArray.elementAt(index: Int): Int {\n   return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size\") }\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun LongArray.elementAt(index: Int): Long {\n
    return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size\") }\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun FloatArray.elementAt(index: Int): Float {\n   return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size\") }\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun DoubleArray.elementAt(index: Int): Double {\n   return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size\") }\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this

```
 array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun BooleanArray.elementAt(index: Int): Boolean {\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size}\") }\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun CharArray.elementAt(index: Int): Char {\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size}\") }\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic actual fun <T> Array<out T>.asList(): List<T> {\n    return ArrayList<T>(this.unsafeCast<Array<Any?>>())\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun ByteArray.asList(): List<Byte> {\n return this.unsafeCast<Array<Byte>>().asList()\n}\n\n/**\n
 * Returns a [List] that wraps the original array.\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun ShortArray.asList(): List<Short> {\n    return this.unsafeCast<Array<Short>>().asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun IntArray.asList(): List<Int> {\n    return this.unsafeCast<Array<Int>>().asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun LongArray.asList(): List<Long> {\n    return this.unsafeCast<Array<Long>>().asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun FloatArray.asList(): List<Float> {\n    return this.unsafeCast<Array<Float>>().asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun DoubleArray.asList(): List<Double>
 {\n    return this.unsafeCast<Array<Double>>().asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun BooleanArray.asList(): List<Boolean> {\n    return this.unsafeCast<Array<Boolean>>().asList()\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\npublic actual fun CharArray.asList(): List<Char> {\n    return object : AbstractList<Char>(), RandomAccess {\n      override val size: Int get() = this@asList.size\n        override fun isEmpty(): Boolean = this@asList.isEmpty()\n    override fun contains(element: Char): Boolean = this@asList.contains(element)\n        override fun get(index: Int): Char {\n            AbstractList.checkElementIndex(index, size)\n            return this@asList[index]\n        }\n        override fun indexOf(element: Char): Int {\n            @Suppress(\"USELESS_CAST\")\n            if ((element as Any?) !is Char) return -1\n            return this@asList.indexOf(element)\n
        }\n        override fun lastIndexOf(element: Char): Int {\n            @Suppress(\"USELESS_CAST\")\n            if ((element as Any?) !is Char) return -1\n            return this@asList.lastIndexOf(element)\n        }\n    }\n}\n\n/**\n * Returns `true` if the two specified arrays are *deeply* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * If two corresponding elements are nested arrays, they are also compared deeply.\n * If any of arrays contains itself on any nesting level the behavior is undefined.\n * \n * The elements of other types are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic actual infix fun <T> Array<out T>.contentDeepEquals(other: Array<out T>): Boolean {\n    return this.contentDeepEquals(other)\n}\n\n/**\n
 * Returns `true` if the two specified arrays are *deeply* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The specified arrays are also considered deeply equal if both are `null`.\n * \n * If two corresponding elements are nested arrays, they are also compared deeply.\n * If any of arrays contains itself on any nesting level the behavior is undefined.\n * \n * The elements of other types are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayDeepEquals\")\npublic actual infix fun <T> Array<out T>?.contentDeepEquals(other: Array<out T>?): Boolean {\n    definedExternally\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n * Nested arrays are treated as lists too.\n * \n * If any of arrays contains itself on any nesting
```

level the behavior is undefined.\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic actual fun <T> Array<out T>.contentDeepHashCode(): Int {\n   return this.contentDeepHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n * Nested arrays are treated as lists too.\n * \n * If any of arrays contains itself on any nesting level the behavior is undefined.\n
*/\n@SinceKotlin(\"1.4\")\n@library(\"arrayDeepHashCode\")\npublic actual fun <T> Array<out T>?.contentDeepHashCode(): Int {\n   definedExternally\n}\n\n/**\n * Returns a string representation of the contents of this array as if it is a [List].\n * Nested arrays are treated as lists too.\n * \n * If any of arrays contains itself on any nesting level that reference\n * is rendered as `\"[...]\"` to prevent recursion.\n * \n * @sample samples.collections.Arrays.ContentOperations.contentDeepToString\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.LowPriorityInOverloadResolution\npublic actual fun <T> Array<out T>.contentDeepToString(): String {\n   return this.contentDeepToString()\n}\n\n/**\n * Returns a string representation of the contents of this array as if it is a [List].\n * Nested arrays are treated as lists too.\n * \n * If any of arrays contains itself on any nesting level that reference\n * is rendered as `\"[...]\"` to prevent recursion.\n * \n * @sample samples.collections.Arrays.ContentOperations.contentDeepToString\n
*/\n@SinceKotlin(\"1.4\")\n@library(\"arrayDeepToString\")\npublic actual fun <T> Array<out T>?.contentDeepToString(): String {\n   definedExternally\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*/\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual infix fun <T> Array<out T>.contentEquals(other: Array<out T>): Boolean {\n   return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*/\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual infix fun ByteArray.contentEquals(other: ByteArray): Boolean {\n   return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n *
 i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual infix fun ShortArray.contentEquals(other: ShortArray): Boolean {\n   return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n
*/\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince
 = \"1.4\")\npublic actual infix fun IntArray.contentEquals(other: IntArray): Boolean {\n   return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual infix fun LongArray.contentEquals(other: LongArray): Boolean {\n   return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the

same elements in the same order.\n * \n * The elements are compared for equality with
the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual infix fun FloatArray.contentEquals(other: FloatArray): Boolean {\n return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual infix fun DoubleArray.contentEquals(other: DoubleArray): Boolean {\n
return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual infix fun BooleanArray.contentEquals(other: BooleanArray): Boolean {\n return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and
`-0.0` is not equal to `0.0`.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual infix fun CharArray.contentEquals(other: CharArray): Boolean {\n return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayEquals\")\npublic actual infix fun <T> Array<out T>?.contentEquals(other: Array<out T>?): Boolean {\n definedExternally\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n
* The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayEquals\")\npublic actual infix fun ByteArray?.contentEquals(other: ByteArray?): Boolean {\n definedExternally\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayEquals\")\npublic actual infix fun ShortArray?.contentEquals(other: ShortArray?): Boolean {\n definedExternally\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of
the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayEquals\")\npublic actual infix fun IntArray?.contentEquals(other: IntArray?): Boolean {\n definedExternally\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayEquals\")\npublic actual infix fun LongArray?.contentEquals(other:

LongArray?): Boolean {\n   definedExternally\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n
 * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayEquals\")\npublic actual infix fun FloatArray?.contentEquals(other: FloatArray?): Boolean {\n   definedExternally\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayEquals\")\npublic actual infix fun DoubleArray?.contentEquals(other: DoubleArray?): Boolean {\n   definedExternally\n}\n\n/**\n * Returns `true` if the two specified
 arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayEquals\")\npublic actual infix fun BooleanArray?.contentEquals(other: BooleanArray?): Boolean {\n   definedExternally\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n * \n * The elements are compared for equality with the [equals][Any.equals] function.\n * For floating point numbers it means that `NaN` is equal to itself and `-0.0` is not equal to `0.0`.\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayEquals\")\npublic actual infix fun CharArray?.contentEquals(other: CharArray?): Boolean {\n   definedExternally\n}\n\n/**\n
 * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun <T> Array<out T>.contentHashCode(): Int {\n   return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun ByteArray.contentHashCode(): Int {\n   return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun ShortArray.contentHashCode(): Int {\n   return this.contentHashCode()\n}\n\n/**\n
 * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun IntArray.contentHashCode(): Int {\n   return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun LongArray.contentHashCode(): Int {\n   return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun FloatArray.contentHashCode(): Int {\n   return this.contentHashCode()\n}\n\n/**\n * Returns a hash code
 based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun DoubleArray.contentHashCode(): Int {\n   return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun BooleanArray.contentHashCode(): Int {\n   return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid

deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun CharArray.contentHashCode(): Int {\n   return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents
 of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayHashCode\")\npublic actual fun <T> Array<out T>?.contentHashCode(): Int {\n   definedExternally\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayHashCode\")\npublic actual fun ByteArray?.contentHashCode(): Int {\n   definedExternally\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayHashCode\")\npublic actual fun ShortArray?.contentHashCode(): Int {\n   definedExternally\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayHashCode\")\npublic actual fun IntArray?.contentHashCode(): Int {\n   definedExternally\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayHashCode\")\npublic
 actual fun LongArray?.contentHashCode(): Int {\n   definedExternally\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayHashCode\")\npublic actual fun FloatArray?.contentHashCode(): Int {\n   definedExternally\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayHashCode\")\npublic actual fun DoubleArray?.contentHashCode(): Int {\n   definedExternally\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayHashCode\")\npublic actual fun BooleanArray?.contentHashCode(): Int {\n   definedExternally\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayHashCode\")\npublic actual fun CharArray?.contentHashCode(): Int {\n   definedExternally\n}\n\n/**\n
 * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun <T> Array<out T>.contentToString(): String {\n   return this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun ByteArray.contentToString(): String {\n   return this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n
 */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun ShortArray.contentToString(): String {\n   return this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun IntArray.contentToString(): String {\n   return this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince
 = \"1.4\")\npublic actual fun LongArray.contentToString(): String {\n   return this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun FloatArray.contentToString(): String {\n   return this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to

avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun DoubleArray.contentToString(): String {\n   return this.contentToString()\n}\n\n/**\n * Returns a string representation of
the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun BooleanArray.contentToString(): String {\n   return this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.1\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\npublic actual fun CharArray.contentToString(): String {\n   return this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayToString\")\npublic
actual fun <T> Array<out T>?.contentToString(): String {\n   definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayToString\")\npublic actual fun ByteArray?.contentToString(): String {\n   definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayToString\")\npublic actual fun ShortArray?.contentToString(): String {\n   definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayToString\")\npublic
actual fun IntArray?.contentToString(): String {\n   definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayToString\")\npublic actual fun LongArray?.contentToString(): String {\n   definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayToString\")\npublic actual fun FloatArray?.contentToString(): String {\n   definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayToString\")\npublic actual fun DoubleArray?.contentToString():
String {\n   definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayToString\")\npublic actual fun BooleanArray?.contentToString():
String {\n   definedExternally\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\n@library(\"arrayToString\")\npublic actual fun CharArray?.contentToString(): String {\n   definedExternally\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n *
@param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array

starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual inline fun <T> Array<out T>.copyInto(destination: Array<T>, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): Array<T> {\n arrayCopy(this, destination, destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual inline fun ByteArray.copyInto(destination: ByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ByteArray {\n arrayCopy(this.unsafeCast<Array<Byte>>(), destination.unsafeCast<Array<Byte>>(), destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual inline fun ShortArray.copyInto(destination: ShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ShortArray {\n arrayCopy(this.unsafeCast<Array<Short>>(), destination.unsafeCast<Array<Short>>(), destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual inline fun IntArray.copyInto(destination: IntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): IntArray {\n arrayCopy(this.unsafeCast<Array<Int>>(), destination.unsafeCast<Array<Int>>(), destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into

the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex]
or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
*/\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual inline fun LongArray.copyInto(destination: LongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): LongArray {\n arrayCopy(this.unsafeCast<Array<Long>>(), destination.unsafeCast<Array<Long>>(), destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination
range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
*/\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual inline fun FloatArray.copyInto(destination: FloatArray, destinationOffset: Int = 0, startIndex:
Int = 0, endIndex: Int = size): FloatArray {\n arrayCopy(this.unsafeCast<Array<Float>>(), destination.unsafeCast<Array<Float>>(), destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange
doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n
*/\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual inline fun DoubleArray.copyInto(destination: DoubleArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): DoubleArray {\n arrayCopy(this.unsafeCast<Array<Double>>(), destination.unsafeCast<Array<Double>>(), destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy
to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this

array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual inline fun BooleanArray.copyInto(destination: BooleanArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): BooleanArray {\n arrayCopy(this.unsafeCast<Array<Boolean>>(), destination.unsafeCast<Array<Boolean>>(), destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual inline fun CharArray.copyInto(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): CharArray {\n arrayCopy(this.unsafeCast<Array<Char>>(), destination.unsafeCast<Array<Char>>(), destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\n@Suppress(\"ACTUAL_WITHOUT_EXPECT\", \"NOTHING_TO_INLINE\")\npublic actual inline fun <T> Array<out T>.copyOf(): Array<T> {\n return this.asDynamic().slice()\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline fun ByteArray.copyOf(): ByteArray {\n return this.asDynamic().slice()\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline fun ShortArray.copyOf(): ShortArray {\n return this.asDynamic().slice()\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline fun IntArray.copyOf(): IntArray {\n return this.asDynamic().slice()\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\npublic actual fun LongArray.copyOf(): LongArray {\n return withType(\"LongArray\", this.asDynamic().slice())\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline fun FloatArray.copyOf(): FloatArray {\n return this.asDynamic().slice()\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline fun DoubleArray.copyOf(): DoubleArray {\n return this.asDynamic().slice()\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\npublic actual fun BooleanArray.copyOf(): BooleanArray {\n return withType(\"BooleanArray\", this.asDynamic().slice())\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\npublic actual fun CharArray.copyOf(): CharArray {\n return withType(\"CharArray\", this.asDynamic().slice())\n}\n\n/**\n * Returns new array which is a copy of

the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n */\npublic actual fun ByteArray.copyOf(newSize: Int): ByteArray {\n   require(newSize >= 0) { \"Invalid new array size: $newSize.\" }\n   return fillFrom(this, ByteArray(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size
of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n */\npublic actual fun ShortArray.copyOf(newSize: Int): ShortArray {\n   require(newSize >= 0) { \"Invalid new array size: $newSize.\" }\n   return fillFrom(this, ShortArray(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n */\npublic actual fun IntArray.copyOf(newSize: Int): IntArray {\n   require(newSize >= 0) { \"Invalid new array size:
$newSize.\" }\n   return fillFrom(this, IntArray(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n */\npublic actual fun LongArray.copyOf(newSize: Int): LongArray {\n   require(newSize >= 0) { \"Invalid new array size: $newSize.\" }\n   return withType(\"LongArray\", arrayCopyResize(this, newSize, 0L))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize]
is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n */\npublic actual fun FloatArray.copyOf(newSize: Int): FloatArray {\n   require(newSize >= 0) { \"Invalid new array size: $newSize.\" }\n   return fillFrom(this, FloatArray(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n */\npublic actual fun DoubleArray.copyOf(newSize: Int): DoubleArray {\n   require(newSize >= 0) { \"Invalid new array size: $newSize.\" }\n   return fillFrom(this, DoubleArray(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with `false` values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `false` values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n */\npublic actual fun BooleanArray.copyOf(newSize: Int): BooleanArray {\n   require(newSize >= 0) { \"Invalid new array size: $newSize.\" }\n   return withType(\"BooleanArray\", arrayCopyResize(this, newSize, false))\n}\n\n/**\n * Returns new array which is a copy

of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with null char (`\\u0000`) values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with null char (`\\u0000`) values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizedPrimitiveCopyOf\n */\npublic actual fun CharArray.copyOf(newSize: Int): CharArray {\n require(newSize >= 0) { \"Invalid new array size: $newSize.\" }\n return withType(\"CharArray\", fillFrom(this, CharArray(newSize)))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with `null` values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated
to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with `null` values.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.resizingCopyOf\n */\n@Suppress(\"ACTUAL_WITHOUT_EXPECT\")\npublic actual fun <T> Array<out T>.copyOf(newSize: Int): Array<T?> {\n require(newSize >= 0) { \"Invalid new array size: $newSize.\" }\n return arrayCopyResize(this, newSize, null)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@Suppress(\"ACTUAL_WITHOUT_EXPECT\")\npublic actual fun <T> Array<out T>.copyOfRange(fromIndex:
Int, toIndex: Int): Array<T> {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n return this.asDynamic().slice(fromIndex, toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic actual fun ByteArray.copyOfRange(fromIndex: Int, toIndex: Int): ByteArray {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n return this.asDynamic().slice(fromIndex, toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive)
to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic actual fun ShortArray.copyOfRange(fromIndex: Int, toIndex: Int): ShortArray {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n return this.asDynamic().slice(fromIndex, toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic actual fun IntArray.copyOfRange(fromIndex: Int, toIndex: Int): IntArray {\n AbstractList.checkRangeIndexes(fromIndex,
toIndex, size)\n return this.asDynamic().slice(fromIndex, toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic actual fun LongArray.copyOfRange(fromIndex: Int, toIndex: Int): LongArray {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n return withType(\"LongArray\", this.asDynamic().slice(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param

toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException
if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws
IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic actual fun
FloatArray.copyOfRange(fromIndex: Int, toIndex: Int): FloatArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n   return this.asDynamic().slice(fromIndex,
toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n *
@param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to
copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the
size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic
actual fun DoubleArray.copyOfRange(fromIndex: Int, toIndex: Int): DoubleArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n   return this.asDynamic().slice(fromIndex,
toIndex)\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n *
@param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to
copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the
size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\npublic
actual fun BooleanArray.copyOfRange(fromIndex: Int, toIndex: Int): BooleanArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n   return withType(\"BooleanArray\",
this.asDynamic().slice(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the specified
range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param
toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is
less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n */\npublic actual fun CharArray.copyOfRange(fromIndex: Int, toIndex: Int): CharArray {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n   return withType(\"CharArray\",
this.asDynamic().slice(fromIndex, toIndex))\n}\n\n/**\n * Fills this array or its subrange with the specified
[element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param
toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws
IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n *
@throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*/\n@SinceKotlin(\"1.3\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic
actual fun <T> Array<T>.fill(element: T, fromIndex: Int = 0, toIndex: Int =
size): Unit {\n   AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n   nativeFill(element, fromIndex,
toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param
fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive)
to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater
than [toIndex].\n
*/\n@SinceKotlin(\"1.3\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic
actual fun ByteArray.fill(element: Byte, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n   nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n
* Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of
the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this
array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is
greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*/\n@SinceKotlin(\"1.3\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic
actual fun ShortArray.fill(element: Short, fromIndex: Int = 0, toIndex: Int = size): Unit {\n
AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n   nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n
* Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range

(inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun IntArray.fill(element: Int, fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun LongArray.fill(element: Long, fromIndex:
Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun FloatArray.fill(element: Float, fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param
fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun DoubleArray.fill(element: Double, fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero
or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun BooleanArray.fill(element: Boolean, fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun CharArray.fill(element:
Char, fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n nativeFill(element, fromIndex, toIndex);\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@Suppress(\"ACTUAL_WITHOUT_EXPECT\", \"NOTHING_TO_INLINE\")\npublic actual inline operator fun <T> Array<out T>.plus(element: T): Array<T> {\n return this.asDynamic().concat(arrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the

original array and then the given [element].\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline operator fun ByteArray.plus(element: Byte): ByteArray {\n   return plus(byteArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline operator fun ShortArray.plus(element: Short): ShortArray {\n   return plus(shortArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline operator fun IntArray.plus(element: Int): IntArray {\n   return plus(intArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline operator fun LongArray.plus(element: Long): LongArray {\n   return plus(longArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline operator fun FloatArray.plus(element: Float): FloatArray {\n   return plus(floatArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline operator fun DoubleArray.plus(element:
Double): DoubleArray {\n   return plus(doubleArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline operator fun BooleanArray.plus(element: Boolean): BooleanArray {\n   return plus(booleanArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline operator fun CharArray.plus(element: Char): CharArray {\n   return plus(charArrayOf(element))\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\n@Suppress(\"ACTUAL_WITHOUT_EXPECT\")\npublic actual operator fun <T> Array<out T>.plus(elements: Collection<T>): Array<T> {\n   return arrayPlusCollection(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original
array and then all elements of the given [elements] collection.\n */\npublic actual operator fun ByteArray.plus(elements: Collection<Byte>): ByteArray {\n   return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun ShortArray.plus(elements: Collection<Short>): ShortArray {\n   return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun IntArray.plus(elements: Collection<Int>): IntArray {\n return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun LongArray.plus(elements: Collection<Long>): LongArray {\n   return arrayPlusCollection(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun FloatArray.plus(elements: Collection<Float>): FloatArray {\n   return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun DoubleArray.plus(elements: Collection<Double>): DoubleArray {\n   return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun BooleanArray.plus(elements: Collection<Boolean>): BooleanArray {\n   return arrayPlusCollection(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\npublic actual operator fun CharArray.plus(elements: Collection<Char>): CharArray {\n   return fillFromCollection(this.copyOf(size + elements.size), this.size, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress(\"ACTUAL_WITHOUT_EXPECT\",

\"NOTHING_TO_INLINE\")\npublic actual inline operator fun <T> Array<out T>.plus(elements: Array<out T>): Array<T> {\n   return this.asDynamic().concat(elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline operator fun ByteArray.plus(elements: ByteArray): ByteArray {\n   return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline operator fun ShortArray.plus(elements: ShortArray): ShortArray {\n   return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline operator fun IntArray.plus(elements: IntArray): IntArray {\n   return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline operator fun LongArray.plus(elements: LongArray): LongArray {\n   return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline operator fun FloatArray.plus(elements: FloatArray): FloatArray {\n   return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline operator fun DoubleArray.plus(elements: DoubleArray): DoubleArray {\n   return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline operator fun BooleanArray.plus(elements: BooleanArray): BooleanArray {\n   return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline operator fun CharArray.plus(elements: CharArray): CharArray {\n   return primitiveArrayConcat(this, elements)\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@Suppress(\"ACTUAL_WITHOUT_EXPECT\", \"NOTHING_TO_INLINE\")\npublic actual inline fun <T> Array<out T>.plusElement(element: T): Array<T> {\n   return this.asDynamic().concat(arrayOf(element))\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\n@library(\"primitiveArraySort\")\npublic actual fun IntArray.sort(): Unit {\n   definedExternally\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\npublic actual fun LongArray.sort(): Unit {\n   @Suppress(\"DEPRECATION\")\n   if (size > 1) sort { a: Long, b: Long -> a.compareTo(b) }\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\n@library(\"primitiveArraySort\")\npublic actual fun ByteArray.sort(): Unit {\n   definedExternally\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\n@library(\"primitiveArraySort\")\npublic actual fun ShortArray.sort(): Unit {\n   definedExternally\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\n@library(\"primitiveArraySort\")\npublic actual fun DoubleArray.sort(): Unit {\n   definedExternally\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\n@library(\"primitiveArraySort\")\npublic actual fun FloatArray.sort(): Unit {\n   definedExternally\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\n@library(\"primitiveArraySort\")\npublic actual fun CharArray.sort(): Unit {\n   definedExternally\n}\n\n/**\n * Sorts the array in-place according to the natural order of its elements.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @sample samples.collections.Arrays.Sorting.sortArrayOfComparable\n */\npublic actual fun <T : Comparable<T>> Array<out T>.sort(): Unit {\n   if (size > 1) sortArray(this)\n}\n\n/**\n * Sorts the array in-

place according to the order specified by the given [comparison] function.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\n@Deprecated(\"Use sortWith instead\", ReplaceWith(\"this.sortWith(Comparator(comparison))\"))\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\npublic fun <T> Array<out T>.sort(comparison: (a: T, b: T) -> Int): Unit {\n   if (size > 1) sortArrayWith(this, comparison)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArrayOfComparable\n */\n@SinceKotlin(\"1.4\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun <T : Comparable<T>> Array<out T>.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n   AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n   sortArrayWith(this, fromIndex, toIndex, naturalOrder())\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun ByteArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n   AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n   val subarray = this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<ByteArray>()\n   subarray.sort()\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun ShortArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n   AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n   val subarray = this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<ShortArray>()\n   subarray.sort()\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun IntArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n   AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n   val subarray = this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<IntArray>()\n   subarray.sort()\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic

actual fun LongArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n sortArrayWith(this.unsafeCast<Array<Long>>(), fromIndex, toIndex, naturalOrder())\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun FloatArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val subarray = this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<FloatArray>()\n subarray.sort()\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun DoubleArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val subarray = this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<DoubleArray>()\n subarray.sort()\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun CharArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n val subarray = this.asDynamic().subarray(fromIndex, toIndex).unsafeCast<CharArray>()\n subarray.sort()\n}\n\n/**\n * Sorts the array in-place according to the order specified by the given [comparison] function.\n */\n@Deprecated(\"Use other sorting functions from the Standard Library\")\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.sort(noinline comparison: (a: Byte, b: Byte) -> Int): Unit {\n nativeSort(comparison)\n}\n\n/**\n * Sorts the array in-place according to the order specified by the given [comparison] function.\n */\n@Deprecated(\"Use other sorting functions from the Standard Library\")\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.sort(noinline comparison: (a: Short, b: Short) -> Int): Unit {\n nativeSort(comparison)\n}\n\n/**\n * Sorts the array in-place according to the order specified by the given [comparison] function.\n */\n@Deprecated(\"Use other sorting functions from the Standard Library\")\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.sort(noinline comparison: (a: Int, b: Int) -> Int): Unit {\n nativeSort(comparison)\n}\n\n/**\n * Sorts the array in-place according to the order specified by the given [comparison] function.\n */\n@Deprecated(\"Use other sorting functions from the Standard Library\")\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.sort(noinline comparison: (a: Long, b: Long) -> Int): Unit {\n nativeSort(comparison)\n}\n\n/**\n * Sorts the array in-place according to the order specified by the given [comparison] function.\n */\n@Deprecated(\"Use other sorting functions from the Standard

Library\")\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n@kotlin.internal.InlineOnly\npublic inline fun FloatArray.sort(noinline comparison: (a: Float, b: Float) -> Int): Unit {\n nativeSort(comparison)\n}\n\n/**\n * Sorts the array in-place according to the order specified by the given [comparison] function.\n */\n@Deprecated(\"Use other sorting functions from the Standard Library\")\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleArray.sort(noinline comparison: (a: Double, b: Double) -> Int): Unit {\n nativeSort(comparison)\n}\n\n/**\n * Sorts the array in-place according to the order specified by the given [comparison] function.\n */\n@Deprecated(\"Use other sorting functions from the Standard Library\")\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n@kotlin.internal.InlineOnly\npublic inline fun CharArray.sort(noinline comparison: (a: Char, b: Char) -> Int): Unit {\n nativeSort(comparison)\n}\n\n/**\n * Sorts the array in-place according to the order specified by the given [comparator].\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic actual fun <T> Array<out T>.sortWith(comparator: Comparator<in T>): Unit {\n if (size > 1) sortArrayWith(this, comparator)\n}\n\n/**\n * Sorts a range in the array in-place with the given [comparator].\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun <T> Array<out T>.sortWith(comparator: Comparator<in T>, fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n sortArrayWith(this, fromIndex, toIndex, comparator)\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic actual fun ByteArray.toTypedArray(): Array<Byte> {\n return js(\"[]\").slice.call(this)\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic actual fun ShortArray.toTypedArray(): Array<Short> {\n return js(\"[]\").slice.call(this)\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic actual fun IntArray.toTypedArray(): Array<Int> {\n return js(\"[]\").slice.call(this)\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic actual fun LongArray.toTypedArray(): Array<Long> {\n return js(\"[]\").slice.call(this)\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic actual fun FloatArray.toTypedArray(): Array<Float> {\n return js(\"[]\").slice.call(this)\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic actual fun DoubleArray.toTypedArray(): Array<Double> {\n return js(\"[]\").slice.call(this)\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic actual fun BooleanArray.toTypedArray(): Array<Boolean> {\n return js(\"[]\").slice.call(this)\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\npublic actual fun CharArray.toTypedArray(): Array<Char> {\n return Array(size) { index -> this[index] }\n}\n\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n@file:kotlin.jvm.JvmName(\"ComparisonsKt\")\n@file:kotlin.jvm.JvmMultifileClass\n\npackage kotlin.comparisons\n\n/**\n * Compares two values using the specified functions [selectors] to calculate the result of the comparison.\n * The functions are called sequentially, receive the given values [a] and [b] and return [Comparable]\n * objects. As soon as the [Comparable] instances returned by a function for [a] and [b] values do not\n * compare as equal, the result of that comparison is returned.\n *\n * @sample samples.comparisons.Comparisons.compareValuesByWithSelectors\n

```
*/\npublic fun <T> compareValuesBy(a: T, b: T, vararg selectors: (T) -> Comparable<*>?): Int {\n require(selectors.size > 0)\n    return compareValuesByImpl(a, b, selectors)\n}\n\nprivate fun <T> compareValuesByImpl(a: T, b: T, selectors: Array<out (T) -> Comparable<*>?>): Int {\n    for (fn in selectors) {\n        val v1 = fn(a)\n        val v2 = fn(b)\n        val diff = compareValues(v1, v2)\n        if (diff != 0) return diff\n    }\n return 0\n}\n\n/**\n * Compares two values using the specified [selector] function to calculate the result of the comparison.\n * The function is applied to the given values [a] and [b] and return [Comparable] objects.\n * The result of comparison of these [Comparable] instances is returned.\n *\n * @sample samples.comparisons.Comparisons.compareValuesByWithSingleSelector\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> compareValuesBy(a: T, b: T, selector: (T) -> Comparable<*>?): Int {\n return compareValues(selector(a), selector(b))\n}\n\n/**\n * Compares two values using the specified [selector] function to calculate the result of the comparison.\n * The function is applied to the given values [a] and [b] and return objects of type K which are then being\n * compared with the given [comparator].\n *\n * @sample samples.comparisons.Comparisons.compareValuesByWithComparator\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T, K> compareValuesBy(a: T, b: T, comparator: Comparator<in K>, selector: (T) -> K): Int {\n    return comparator.compare(selector(a), selector(b))\n}\n\n//// Not so useful without type inference for receiver of expression\n//// compareValuesWith(v1, v2, compareBy { it.prop1 } thenByDescending { it.prop2 })\n///**\n// * Compares two values using the specified [comparator].\n// */\n//@Suppress(\"NOTHING_TO_INLINE\")\n//public inline fun <T> compareValuesWith(a: T, b: T, comparator: Comparator<T>): Int = comparator.compare(a, b)\n//\n\n\n/**\n * Compares
 two nullable [Comparable] values. Null is considered less than any value.\n *\n * @sample samples.comparisons.Comparisons.compareValues\n */\npublic fun <T : Comparable<*>> compareValues(a: T?, b: T?): Int {\n    if (a === b) return 0\n    if (a == null) return -1\n    if (b == null) return 1\n\n @Suppress(\"UNCHECKED_CAST\")\n    return (a as Comparable<Any>).compareTo(b)\n}\n\n/**\n * Creates a comparator using the sequence of functions to calculate a result of comparison.\n * The functions are called sequentially, receive the given values `a` and `b` and return [Comparable]\n * objects. As soon as the [Comparable] instances returned by a function for `a` and `b` values do not\n * compare as equal, the result of that comparison is returned from the [Comparator].\n *\n * @sample samples.comparisons.Comparisons.compareByWithSelectors\n */\npublic fun <T> compareBy(vararg selectors: (T) -> Comparable<*>?): Comparator<T> {\n require(selectors.size > 0)\n    return Comparator
 { a, b -> compareValuesByImpl(a, b, selectors) }\n}\n\n\n/**\n * Creates a comparator using the function to transform value to a [Comparable] instance for comparison.\n *\n * @sample samples.comparisons.Comparisons.compareByWithSingleSelector\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> compareBy(crossinline selector: (T) -> Comparable<*>?): Comparator<T> =\n    Comparator { a, b -> compareValuesBy(a, b, selector) }\n\n/**\n * Creates a comparator using the [selector] function to transform values being compared and then applying\n * the specified [comparator] to compare transformed values.\n *\n * @sample samples.comparisons.Comparisons.compareByWithComparator\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T, K> compareBy(comparator: Comparator<in K>, crossinline selector: (T) -> K): Comparator<T> =\n Comparator { a, b -> compareValuesBy(a, b, comparator, selector) }\n\n/**\n * Creates a descending comparator using the function to transform value to a [Comparable]
 instance for comparison.\n *\n * @sample samples.comparisons.Comparisons.compareByDescendingWithSingleSelector\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> compareByDescending(crossinline selector: (T) -> Comparable<*>?): Comparator<T> =\n    Comparator { a, b -> compareValuesBy(b, a, selector) }\n\n/**\n * Creates a descending comparator using the [selector] function to transform values being compared and then applying\n * the specified [comparator] to compare transformed values.\n *\n * Note that an order of [comparator] is reversed by this wrapper.\n *\n * @sample samples.comparisons.Comparisons.compareByDescendingWithComparator\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T, K> compareByDescending(comparator: Comparator<in K>,
```

crossinline selector: (T) -> K): Comparator<T> =\n   Comparator { a, b -> compareValuesBy(b, a, comparator, selector) }\n\n/**\n * Creates a comparator comparing values after the primary comparator defined them equal. It uses\n * the function
 to transform value to a [Comparable] instance for comparison.\n *\n * @sample samples.comparisons.Comparisons.thenBy\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Comparator<T>.thenBy(crossinline selector: (T) -> Comparable<*>?): Comparator<T> =\n   Comparator { a, b ->\n   val previousCompare = this@thenBy.compare(a, b)\n   if (previousCompare != 0) previousCompare else compareValuesBy(a, b, selector)\n   }\n\n/**\n * Creates a comparator comparing values after the primary comparator defined them equal. It uses\n * the [selector] function to transform values and then compares them with the given [comparator].\n *\n * @sample samples.comparisons.Comparisons.thenByWithComparator\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T, K> Comparator<T>.thenBy(comparator: Comparator<in K>, crossinline selector: (T) -> K): Comparator<T> =\n   Comparator { a, b ->\n   val previousCompare = this@thenBy.compare(a, b)\n   if (previousCompare != 0) previousCompare
 else compareValuesBy(a, b, comparator, selector)\n   }\n\n/**\n * Creates a descending comparator using the primary comparator and\n * the function to transform value to a [Comparable] instance for comparison.\n *\n * @sample samples.comparisons.Comparisons.thenByDescending\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Comparator<T>.thenByDescending(crossinline selector: (T) -> Comparable<*>?): Comparator<T> =\n   Comparator { a, b ->\n   val previousCompare = this@thenByDescending.compare(a, b)\n   if (previousCompare != 0) previousCompare else compareValuesBy(b, a, selector)\n   }\n\n/**\n * Creates a descending comparator comparing values after the primary comparator defined them equal. It uses\n * the [selector] function to transform values and then compares them with the given [comparator].\n *\n * @sample samples.comparisons.Comparisons.thenByDescendingWithComparator\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T, K> Comparator<T>.thenByDescending(comparator:
 Comparator<in K>, crossinline selector: (T) -> K): Comparator<T> =\n   Comparator { a, b ->\n   val previousCompare = this@thenByDescending.compare(a, b)\n   if (previousCompare != 0) previousCompare else compareValuesBy(b, a, comparator, selector)\n   }\n\n\n/**\n * Creates a comparator using the primary comparator and function to calculate a result of comparison.\n *\n * @sample samples.comparisons.Comparisons.thenComparator\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Comparator<T>.thenComparator(crossinline comparison: (a: T, b: T) -> Int): Comparator<T> =\n   Comparator { a, b ->\n   val previousCompare = this@thenComparator.compare(a, b)\n   if (previousCompare != 0) previousCompare else comparison(a, b)\n   }\n\n/**\n * Combines this comparator and the given [comparator] such that the latter is applied only\n * when the former considered values equal.\n *\n * @sample samples.comparisons.Comparisons.then\n */\npublic infix
 fun <T> Comparator<T>.then(comparator: Comparator<in T>): Comparator<T> =\n   Comparator { a, b ->\n   val previousCompare = this@then.compare(a, b)\n   if (previousCompare != 0) previousCompare else comparator.compare(a, b)\n   }\n\n/**\n * Combines this comparator and the given [comparator] such that the latter is applied only\n * when the former considered values equal.\n *\n * @sample samples.comparisons.Comparisons.thenDescending\n */\npublic infix fun <T> Comparator<T>.thenDescending(comparator: Comparator<in T>): Comparator<T> =\n   Comparator<T> { a, b ->\n   val previousCompare = this@thenDescending.compare(a, b)\n   if (previousCompare != 0) previousCompare else comparator.compare(b, a)\n   }\n\n// Not so useful without type inference for receiver of expression\n/**\n * Extends the given [comparator] of non-nullable values to a comparator of nullable values\n * considering `null` value less than any other value.\n * Non-null values are compared with
 the provided [comparator].\n *\n * @sample samples.comparisons.Comparisons.nullsFirstLastWithComparator\n */\npublic fun <T : Any> nullsFirst(comparator: Comparator<in T>): Comparator<T?> =\n   Comparator { a, b ->\n   when {\n   a === b -> 0\n   a == null -> -1\n   b == null -> 1\n   else -> comparator.compare(a, b)\n   }\n   }\n\n/**\n * Provides a comparator of nullable [Comparable] values\n * considering `null` value less than any other value.\n * Non-null values are compared according to their [natural

order][naturalOrder].\n *\n * @sample samples.comparisons.Comparisons.nullsFirstLastComparator\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T : Comparable<T>> nullsFirst(): Comparator<T?> = nullsFirst(naturalOrder())\n\n/**\n * Extends the given [comparator] of non-nullable values to a comparator of nullable values\n * considering `null` value greater than any other value.\n * Non-null values are compared with the provided [comparator].\n
*\n * @sample samples.comparisons.Comparisons.nullsFirstLastWithComparator\n */\npublic fun <T : Any> nullsLast(comparator: Comparator<in T>): Comparator<T?> =\n Comparator { a, b ->\n when {\n a === b -> 0\n a == null -> 1\n b == null -> -1\n else -> comparator.compare(a, b)\n }\n }\n\n/**\n * Provides a comparator of nullable [Comparable] values\n * considering `null` value greater than any other value.\n * Non-null values are compared according to their [natural order][naturalOrder].\n *\n * @sample samples.comparisons.Comparisons.nullsFirstLastComparator\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T : Comparable<T>> nullsLast(): Comparator<T?> = nullsLast(naturalOrder())\n\n/**\n * Returns a comparator that compares [Comparable] objects in natural order.\n *\n * The natural order of a `Comparable` type here means the order established by its `compareTo` function.\n *\n * @sample samples.comparisons.Comparisons.naturalOrderComparator\n
*/\npublic fun <T : Comparable<T>> naturalOrder(): Comparator<T> = @Suppress(\"UNCHECKED_CAST\") (NaturalOrderComparator as Comparator<T>)\n\n/**\n * Returns a comparator that compares [Comparable] objects in reversed natural order.\n *\n * The natural order of a `Comparable` type here means the order established by its `compareTo` function.\n *\n * @sample samples.comparisons.Comparisons.nullsFirstLastWithComparator\n */\npublic fun <T : Comparable<T>> reverseOrder(): Comparator<T> = @Suppress(\"UNCHECKED_CAST\") (ReverseOrderComparator as Comparator<T>)\n\n/**\n * Returns a comparator that imposes the reverse ordering of this comparator.\n *\n * @sample samples.comparisons.Comparisons.reversed\n */\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\")\npublic fun <T> Comparator<T>.reversed(): Comparator<T> = when (this) {\n is ReversedComparator -> this.comparator\n NaturalOrderComparator -> @Suppress(\"UNCHECKED_CAST\") (ReverseOrderComparator
as Comparator<T>)\n ReverseOrderComparator -> @Suppress(\"UNCHECKED_CAST\") (NaturalOrderComparator as Comparator<T>)\n else -> ReversedComparator(this)\n}\n\n\nprivate class ReversedComparator<T>(public val comparator: Comparator<T>) : Comparator<T> {\n override fun compare(a: T, b: T): Int = comparator.compare(b, a)\n @Suppress(\"VIRTUAL_MEMBER_HIDDEN\")\n fun reversed(): Comparator<T> = comparator\n}\n\nprivate object NaturalOrderComparator : Comparator<Comparable<Any>> {\n override fun compare(a: Comparable<Any>, b: Comparable<Any>): Int = a.compareTo(b)\n @Suppress(\"VIRTUAL_MEMBER_HIDDEN\")\n fun reversed(): Comparator<Comparable<Any>> = ReverseOrderComparator\n}\n\nprivate object ReverseOrderComparator : Comparator<Comparable<Any>> {\n override fun compare(a: Comparable<Any>, b: Comparable<Any>): Int = b.compareTo(a)\n @Suppress(\"VIRTUAL_MEMBER_HIDDEN\")\n fun reversed(): Comparator<Comparable<Any>> = NaturalOrderComparator\n}\n","/*\n * Copyright
2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"StandardKt\")\npackage kotlin\n\nimport kotlin.contracts.*\n\n/**\n * An exception is thrown to indicate that a method body remains to be implemented.\n */\npublic class NotImplementedError(message: String = \"An operation is not implemented.\") : Error(message)\n\n/**\n * Always throws [NotImplementedError] stating that operation is not implemented.\n */\n\n@kotlin.internal.InlineOnly\npublic inline fun TODO(): Nothing = throw NotImplementedError()\n\n/**\n * Always throws [NotImplementedError] stating that operation is not implemented.\n *\n * @param reason a string explaining why the implementation is missing.\n */\n@kotlin.internal.InlineOnly\npublic inline fun TODO(reason: String): Nothing = throw NotImplementedError(\"An
operation is not implemented: $reason\")\n\n\n\n/**\n * Calls the specified function [block] and returns its result.\n *\n * For detailed usage information see the documentation for [scope

functions](https://kotlinlang.org/docs/reference/scope-functions.html#run).\n */\n@kotlin.internal.InlineOnly\npublic inline fun <R> run(block: () -> R): R {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n return block()\n}\n\n/**\n * Calls the specified function [block] with `this` value as its receiver and returns its result.\n *\n * For detailed usage information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#run).\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> T.run(block: T.() -> R): R {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n return block()\n}\n\n/**\n * Calls the specified function [block] with the given [receiver] as its receiver and returns its result.\n *\n * For detailed usage information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#with).\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> with(receiver: T, block: T.() -> R): R {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n return receiver.block()\n}\n\n/**\n * Calls the specified function [block] with `this` value as its receiver and returns `this` value.\n *\n * For detailed usage information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#apply).\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> T.apply(block: T.() -> Unit): T {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n block()\n return this\n}\n\n/**\n * Calls the specified function [block] with `this` value as its argument and returns `this` value.\n *\n * For detailed usage information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#also).\n */\n@kotlin.internal.InlineOnly\n@SinceKotlin(\"1.1\")\npublic inline fun <T> T.also(block: (T) -> Unit): T {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n block(this)\n return this\n}\n\n/**\n * Calls the specified function [block] with `this` value as its argument and returns its result.\n *\n * For detailed usage information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#let).\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> T.let(block: (T) -> R): R {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n return block(this)\n}\n\n/**\n * Returns `this` value if it satisfies the given [predicate] or `null`, if it doesn't.\n *\n * For detailed usage information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#takeif-and-takeunless).\n */\n@kotlin.internal.InlineOnly\n@SinceKotlin(\"1.1\")\npublic inline fun <T> T.takeIf(predicate: (T) -> Boolean): T? {\n contract {\n callsInPlace(predicate, InvocationKind.EXACTLY_ONCE)\n }\n return if (predicate(this)) this else null\n}\n\n/**\n * Returns `this` value if it _does not_ satisfy the given [predicate] or `null`, if it does.\n *\n * For detailed usage information see the documentation for [scope functions](https://kotlinlang.org/docs/reference/scope-functions.html#takeif-and-takeunless).\n */\n@kotlin.internal.InlineOnly\n@SinceKotlin(\"1.1\")\npublic inline fun <T> T.takeUnless(predicate: (T) -> Boolean): T? {\n contract {\n callsInPlace(predicate, InvocationKind.EXACTLY_ONCE)\n }\n return if (!predicate(this)) this else null\n}\n\n/**\n * Executes the given function [action] specified number of [times].\n *\n * A zero-based index of current iteration is passed as a parameter to [action].\n *\n * @sample samples.misc.ControlFlow.repeat\n */\n@kotlin.internal.InlineOnly\npublic inline fun repeat(times: Int, action: (Int) -> Unit) {\n contract { callsInPlace(action) }\n\n for (index in 0 until times) {\n action(index)\n }\n}\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.comparisons\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.js.*\n\n/**\n * Returns the greater of two values.\n * \n * If values are equal, returns the first one.\n */\n@SinceKotlin(\"1.1\")\npublic actual fun <T : Comparable<T>> maxOf(a: T, b: T): T {\n return if (a >= b) a else b\n}\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic

```
 actual inline fun maxOf(a: Byte, b: Byte): Byte {\n    return maxOf(a.toInt(),
b.toInt()).unsafeCast<Byte>()\n}\n\n/**\n * Returns the greater of two values.\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun maxOf(a: Short, b: Short): Short
{\n    return maxOf(a.toInt(), b.toInt()).unsafeCast<Short>()\n}\n\n/**\n * Returns the greater of two values.\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun maxOf(a: Int, b: Int): Int {\n
return JsMath.max(a, b)\n}\n\n/**\n * Returns the greater of two values.\n
*/\n@SinceKotlin(\"1.1\")\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline fun maxOf(a: Long, b:
Long): Long {\n    return if (a >= b) a else b\n}\n\n/**\n * Returns the greater of two values.\n * \n * If either value
is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun
maxOf(a: Float, b: Float): Float {\n    return JsMath.max(a, b)\n}\n\n/**\n * Returns the
 greater of two values.\n * \n * If either value is `NaN`, returns `NaN`.\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun maxOf(a: Double, b: Double):
Double {\n    return JsMath.max(a, b)\n}\n\n/**\n * Returns the greater of three values.\n * \n * If there are multiple
equal maximal values, returns the first of them.\n */\n@SinceKotlin(\"1.1\")\npublic actual fun <T :
Comparable<T>> maxOf(a: T, b: T, c: T): T {\n    return maxOf(a, maxOf(b, c))\n}\n\n/**\n * Returns the greater
of three values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun maxOf(a: Byte,
b: Byte, c: Byte): Byte {\n    return JsMath.max(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Byte>()\n}\n\n/**\n *
Returns the greater of three values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline
fun maxOf(a: Short, b: Short, c: Short): Short {\n    return JsMath.max(a.toInt(), b.toInt(),
c.toInt()).unsafeCast<Short>()\n}\n\n/**\n
 * Returns the greater of three values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual
inline fun maxOf(a: Int, b: Int, c: Int): Int {\n    return JsMath.max(a, b, c)\n}\n\n/**\n * Returns the greater of three
values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun maxOf(a: Long, b: Long,
c: Long): Long {\n    return maxOf(a, maxOf(b, c))\n}\n\n/**\n * Returns the greater of three values.\n * \n * If any
value is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun
maxOf(a: Float, b: Float, c: Float): Float {\n    return JsMath.max(a, b, c)\n}\n\n/**\n * Returns the greater of three
values.\n * \n * If any value is `NaN`, returns `NaN`.\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun maxOf(a: Double, b: Double, c:
Double): Double {\n    return JsMath.max(a, b, c)\n}\n\n/**\n * Returns the greater of the given values.\n * \n * If
there are
 multiple equal maximal values, returns the first of them.\n */\n@SinceKotlin(\"1.4\")\npublic actual fun <T :
Comparable<T>> maxOf(a: T, vararg other: T): T {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n
return max\n}\n\n/**\n * Returns the greater of the given values.\n */\n@SinceKotlin(\"1.4\")\npublic actual fun
maxOf(a: Byte, vararg other: Byte): Byte {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return
max\n}\n\n/**\n * Returns the greater of the given values.\n */\n@SinceKotlin(\"1.4\")\npublic actual fun maxOf(a:
Short, vararg other: Short): Short {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return
max\n}\n\n/**\n * Returns the greater of the given values.\n */\n@SinceKotlin(\"1.4\")\npublic actual fun maxOf(a:
Int, vararg other: Int): Int {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return max\n}\n\n/**\n *
Returns the greater of the given values.\n */\n@SinceKotlin(\"1.4\")\npublic actual fun maxOf(a:
 Long, vararg other: Long): Long {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return
max\n}\n\n/**\n * Returns the greater of the given values.\n * \n * If any value is `NaN`, returns `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\npublic actual fun maxOf(a: Float, vararg other: Float): Float {\n    var max = a\n    for (e
in other) max = maxOf(max, e)\n    return max\n}\n\n/**\n * Returns the greater of the given values.\n * \n * If any
value is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic actual fun maxOf(a: Double, vararg other:
Double): Double {\n    var max = a\n    for (e in other) max = maxOf(max, e)\n    return max\n}\n\n/**\n * Returns
the smaller of two values.\n * \n * If values are equal, returns the first one.\n */\n@SinceKotlin(\"1.1\")\npublic
actual fun <T : Comparable<T>> minOf(a: T, b: T): T {\n    return if (a <= b) a else b\n}\n\n/**\n * Returns the
smaller of two values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual
```

inline fun minOf(a: Byte, b: Byte): Byte {\n   return minOf(a.toInt(), b.toInt()).unsafeCast<Byte>()\n}\n\n/**\n * Returns the smaller of two values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Short, b: Short): Short {\n   return minOf(a.toInt(), b.toInt()).unsafeCast<Short>()\n}\n\n/**\n * Returns the smaller of two values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Int, b: Int): Int {\n   return JsMath.min(a, b)\n}\n\n/**\n * Returns the smaller of two values.\n */\n@SinceKotlin(\"1.1\")\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline fun minOf(a: Long, b: Long): Long {\n   return if (a <= b) a else b\n}\n\n/**\n * Returns the smaller of two values.\n * \n * If either value is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Float, b: Float): Float {\n   return JsMath.min(a, b)\n}\n\n/**\n * Returns the smaller of two values.\n * \n * If either value is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Double, b: Double): Double {\n   return JsMath.min(a, b)\n}\n\n/**\n * Returns the smaller of three values.\n * \n * If there are multiple equal minimal values, returns the first of them.\n */\n@SinceKotlin(\"1.1\")\npublic actual fun <T : Comparable<T>> minOf(a: T, b: T, c: T): T {\n   return minOf(a, minOf(b, c))\n}\n\n/**\n * Returns the smaller of three values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Byte, b: Byte, c: Byte): Byte {\n   return JsMath.min(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Byte>()\n}\n\n/**\n * Returns the smaller of three values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Short, b: Short, c: Short): Short {\n   return JsMath.min(a.toInt(), b.toInt(), c.toInt()).unsafeCast<Short>()\n}\n\n/**\n * Returns the smaller of three values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Int, b: Int, c: Int): Int {\n   return JsMath.min(a, b, c)\n}\n\n/**\n * Returns the smaller of three values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Long, b: Long, c: Long): Long {\n   return minOf(a, minOf(b, c))\n}\n\n/**\n * Returns the smaller of three values.\n * \n * If any value is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Float, b: Float, c: Float): Float {\n   return JsMath.min(a, b, c)\n}\n\n/**\n * Returns the smaller of three values.\n * \n * If any value is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic actual inline fun minOf(a: Double, b: Double, c: Double): Double {\n   return JsMath.min(a, b, c)\n}\n\n/**\n * Returns the smaller of the given values.\n * \n * If there are multiple equal minimal values, returns the first of them.\n */\n@SinceKotlin(\"1.4\")\npublic actual fun <T : Comparable<T>> minOf(a: T, vararg other: T): T {\n   var min = a\n   for (e in other) min = minOf(min, e)\n   return min\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin(\"1.4\")\npublic actual fun minOf(a: Byte, vararg other: Byte): Byte {\n   var min = a\n   for (e in other) min = minOf(min, e)\n   return min\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin(\"1.4\")\npublic actual fun minOf(a: Short, vararg other: Short): Short {\n   var min = a\n   for (e in other) min = minOf(min, e)\n   return min\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin(\"1.4\")\npublic actual fun minOf(a: Int, vararg other: Int): Int {\n   var min = a\n   for (e in other) min = minOf(min, e)\n   return min\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin(\"1.4\")\npublic actual fun minOf(a: Long, vararg other: Long): Long {\n   var min = a\n   for (e in other) min = minOf(min, e)\n   return min\n}\n\n/**\n * Returns the smaller of the given values.\n * \n * If any value is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic actual fun minOf(a: Float, vararg other: Float): Float {\n   var min = a\n   for (e in other) min = minOf(min, e)\n   return min\n}\n\n/**\n * Returns the smaller of the given values.\n * \n * If any value is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic actual fun minOf(a: Double, vararg other: Double): Double {\n   var min = a\n   for (e in other) min = minOf(min, e)\n   return min\n}\n\n","/*\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin\n\nimport kotlin.experimental.*\nimport kotlin.jvm.*\n\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@JvmInline\npu

blic
value class ULong @kotlin.internal.IntrinsicConstEvaluation @PublishedApi internal constructor(@PublishedApi internal val data: Long) : Comparable<ULong> {\n\n companion object {\n /**\n * A constant holding the minimum value an instance of ULong can have.\n */\n public const val MIN_VALUE: ULong = ULong(0)\n\n /**\n * A constant holding the maximum value an instance of ULong can have.\n */\n public const val MAX_VALUE: ULong = ULong(-1)\n\n /**\n * The number of bytes used to represent an instance of ULong in a binary form.\n */\n public const val SIZE_BYTES: Int = 8\n\n /**\n * The number of bits used to represent an instance of ULong in a binary form.\n */\n public const val SIZE_BITS: Int = 64\n }\n\n /**\n * Compares this value with the specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun compareTo(other: UByte): Int = this.compareTo(other.toULong())\n\n /**\n * Compares this value with the specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun compareTo(other: UShort): Int = this.compareTo(other.toULong())\n\n /**\n * Compares this value with the specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n public inline
operator fun compareTo(other: UInt): Int = this.compareTo(other.toULong())\n\n /**\n * Compares this value with the specified value for order.\n * Returns zero if this value is equal to the specified other value, a negative number if it's less than other,\n * or a positive number if it's greater than other.\n */\n @kotlin.internal.InlineOnly\n @Suppress(\"OVERRIDE_BY_INLINE\")\n public override inline operator fun compareTo(other: ULong): Int = ulongCompare(this.data, other.data)\n\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UByte): ULong = this.plus(other.toULong())\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UShort): ULong = this.plus(other.toULong())\n /** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: UInt): ULong = this.plus(other.toULong())\n
/** Adds the other value to this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun plus(other: ULong): ULong = ULong(this.data.plus(other.data))\n\n /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UByte): ULong = this.minus(other.toULong())\n /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UShort): ULong = this.minus(other.toULong())\n /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: UInt): ULong = this.minus(other.toULong())\n /** Subtracts the other value from this value. */\n @kotlin.internal.InlineOnly\n public inline operator fun minus(other: ULong): ULong = ULong(this.data.minus(other.data))\n\n /** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n
public inline operator fun times(other: UByte): ULong = this.times(other.toULong())\n /** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: UShort): ULong = this.times(other.toULong())\n /** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: UInt): ULong = this.times(other.toULong())\n /** Multiplies this value by the other value. */\n @kotlin.internal.InlineOnly\n public inline operator fun times(other: ULong): ULong = ULong(this.data.times(other.data))\n\n /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: UByte): ULong = this.div(other.toULong())\n /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n
public inline operator fun div(other: UShort): ULong = this.div(other.toULong())\n /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public

inline operator fun div(other: UInt): ULong = this.div(other.toULong())\n /** Divides this value by the other value, truncating the result to an integer that is closer to zero. */\n @kotlin.internal.InlineOnly\n public inline operator fun div(other: ULong): ULong = ulongDivide(this, other)\n\n /**\n * Calculates the remainder of truncating division of this value by the other value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UByte): ULong = this.rem(other.toULong())\n /**\n * Calculates the remainder of truncating division of this value by the other value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UShort): ULong = this.rem(other.toULong())\n /**\n * Calculates the remainder of truncating division of this value by the other value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: UInt): ULong = this.rem(other.toULong())\n /**\n * Calculates the remainder of truncating division of this value by the other value.\n * \n * The result is always less than the divisor.\n */\n @kotlin.internal.InlineOnly\n public inline operator fun rem(other: ULong): ULong = ulongRemainder(this, other)\n\n /**\n * Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UByte): ULong = this.floorDiv(other.toULong())\n /**\n * Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UShort): ULong = this.floorDiv(other.toULong())\n /**\n * Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: UInt): ULong = this.floorDiv(other.toULong())\n /**\n * Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.\n * \n * For unsigned types, the results of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun floorDiv(other: ULong): ULong = div(other)\n\n /**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UByte): UByte = this.mod(other.toULong()).toUByte()\n /**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UShort): UShort = this.mod(other.toULong()).toUShort()\n /**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: UInt): UInt = this.mod(other.toULong()).toUInt()\n /**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is always less than the divisor.\n * \n * For unsigned types, the remainders of flooring division and truncating division are the same.\n */\n @kotlin.internal.InlineOnly\n public inline fun mod(other: ULong): ULong = rem(other)\n\n /**\n * Returns this value incremented by one.\n *\n * @sample samples.misc.Builtins.inc\n */\n @kotlin.internal.InlineOnly\n public inline operator fun inc(): ULong = ULong(data.inc())\n\n /**\n * Returns this value decremented by one.\n *\n * @sample samples.misc.Builtins.dec\n */\n @kotlin.internal.InlineOnly\n public inline operator fun dec(): ULong = ULong(data.dec())\n\n /** Creates a range from this value to the specified [other] value. */\n @kotlin.internal.InlineOnly\n public inline operator fun rangeTo(other: ULong): ULongRange = ULongRange(this, other)\n\n /**\n * Creates a range from this value up to but excluding the specified [other] value.\n *\n * If the [other] value is less than or equal to `this` value, then the returned range is empty.\n */\n @SinceKotlin(\"1.7\")\n @ExperimentalStdlibApi\n

@kotlin.internal.InlineOnly\n public inline operator fun rangeUntil(other: ULong): ULongRange = this until other\n\n /**\n * Shifts this value left by the [bitCount] number of bits.\n *\n * Note that only the six lowest-order bits of the [bitCount] are used as the shift distance.\n * The shift distance actually used is therefore always in the range `0..63`.\n */\n @kotlin.internal.InlineOnly\n public inline infix fun shl(bitCount: Int): ULong = ULong(data shl bitCount)\n\n /**\n * Shifts this value right by the [bitCount] number of bits, filling the leftmost bits with zeros.\n *\n * Note that only the six lowest-order bits of the [bitCount] are used as the shift distance.\n * The shift distance actually used is therefore always in the range `0..63`.\n */\n @kotlin.internal.InlineOnly\n public inline infix fun shr(bitCount: Int): ULong = ULong(data ushr bitCount)\n\n /** Performs a bitwise AND operation between the two values. */\n @kotlin.internal.InlineOnly\n public inline infix fun and(other: ULong): ULong = ULong(this.data and other.data)\n /** Performs a bitwise OR operation between the two values. */\n @kotlin.internal.InlineOnly\n public inline infix fun or(other: ULong): ULong = ULong(this.data or other.data)\n /** Performs a bitwise XOR operation between the two values. */\n @kotlin.internal.InlineOnly\n public inline infix fun xor(other: ULong): ULong = ULong(this.data xor other.data)\n /** Inverts the bits in this value. */\n @kotlin.internal.InlineOnly\n public inline fun inv(): ULong = ULong(data.inv())\n\n /**\n * Converts this [ULong] value to [Byte].\n *\n * If this value is less than or equals to [Byte.MAX_VALUE], the resulting `Byte` value represents\n * the same numerical value as this `ULong`.\n *\n * The resulting `Byte` value is represented by the least significant 8 bits of this `ULong` value.\n * Note that the resulting `Byte` value may be negative.\n */\n @kotlin.internal.InlineOnly\n public inline fun toByte(): Byte = data.toByte()\n /**\n * Converts this [ULong] value to [Short].\n *\n * If this value is less than or equals to [Short.MAX_VALUE], the resulting `Short` value represents\n * the same numerical value as this `ULong`.\n *\n * The resulting `Short` value is
represented by the least significant 16 bits of this `ULong` value.\n * Note that the resulting `Short` value may be negative.\n */\n @kotlin.internal.InlineOnly\n public inline fun toShort(): Short = data.toShort()\n /**\n * Converts this [ULong] value to [Int].\n *\n * If this value is less than or equals to [Int.MAX_VALUE], the resulting `Int` value represents\n * the same numerical value as this `ULong`.\n *\n * The resulting `Int` value is represented by the least significant 32 bits of this `ULong` value.\n * Note that the resulting `Int` value may be negative.\n */\n @kotlin.internal.InlineOnly\n public inline fun toInt(): Int = data.toInt()\n /**\n * Converts this [ULong] value to [Long].\n *\n * If this value is less than or equals to [Long.MAX_VALUE], the resulting `Long` value represents\n * the same numerical value as this `ULong`. Otherwise the result is negative.\n *\n * The resulting `Long`
value has the same binary representation as this `ULong` value.\n */\n @kotlin.internal.InlineOnly\n public inline fun toLong(): Long = data\n\n /**\n * Converts this [ULong] value to [UByte].\n *\n * If this value is less than or equals to [UByte.MAX_VALUE], the resulting `UByte` value represents\n * the same numerical value as this `ULong`.\n *\n * The resulting `UByte` value is represented by the least significant 8 bits of this `ULong` value.\n */\n @kotlin.internal.InlineOnly\n public inline fun toUByte(): UByte = data.toUByte()\n /**\n * Converts this [ULong] value to [UShort].\n *\n * If this value is less than or equals to [UShort.MAX_VALUE], the resulting `UShort` value represents\n * the same numerical value as this `ULong`.\n *\n * The resulting `UShort` value is represented by the least significant 16 bits of this `ULong` value.\n */\n @kotlin.internal.InlineOnly\n public inline fun toUShort():
UShort = data.toUShort()\n /**\n * Converts this [ULong] value to [UInt].\n *\n * If this value is less than or equals to [UInt.MAX_VALUE], the resulting `UInt` value represents\n * the same numerical value as this `ULong`.\n *\n * The resulting `UInt` value is represented by the least significant 32 bits of this `ULong` value.\n */\n @kotlin.internal.InlineOnly\n public inline fun toUInt(): UInt = data.toUInt()\n /** Returns this value. */\n @kotlin.internal.InlineOnly\n public inline fun toULong(): ULong = this\n\n /**\n * Converts this [ULong] value to [Float].\n *\n * The resulting value is the closest `Float` to this `ULong` value.\n * In case when this `ULong` value is exactly between two `Float`s,\n * the one with zero at least significant bit of mantissa is selected.\n */\n @kotlin.internal.InlineOnly\n public inline fun toFloat(): Float =

this.toDouble().toFloat()\n   /**\n   * Converts this
[ULong] value to [Double].\n   *\n   * The resulting value is the closest `Double` to this `ULong` value.\n   * In case when this `ULong` value is exactly between two `Double`s,\n   * the one with zero at least significant bit of mantissa is selected.\n   */\n   @kotlin.internal.InlineOnly\n   public inline fun toDouble(): Double = ulongToDouble(data)\n\n   public override fun toString(): String = ulongToString(data)\n\n}\n\n/**\n * Converts this [Byte] value to [ULong].\n *\n * If this value is positive, the resulting `ULong` value represents the same numerical value as this `Byte`.\n *\n * The least significant 8 bits of the resulting `ULong` value are the same as the bits of this `Byte` value,\n * whereas the most significant 56 bits are filled with the sign bit of this value.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.toULong(): ULong = ULong(this.toLong())\n/**\n * Converts this
[Short] value to [ULong].\n *\n * If this value is positive, the resulting `ULong` value represents the same numerical value as this `Short`.\n *\n * The least significant 16 bits of the resulting `ULong` value are the same as the bits of this `Short` value,\n * whereas the most significant 48 bits are filled with the sign bit of this value.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.toULong(): ULong = ULong(this.toLong())\n/**\n * Converts this [Int] value to [ULong].\n *\n * If this value is positive, the resulting `ULong` value represents the same numerical value as this `Int`.\n *\n * The least significant 32 bits of the resulting `ULong` value are the same as the bits of this `Int` value,\n * whereas the most significant 32 bits are filled with the sign bit of this value.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline
fun Int.toULong(): ULong = ULong(this.toLong())\n/**\n * Converts this [Long] value to [ULong].\n *\n * If this value is positive, the resulting `ULong` value represents the same numerical value as this `Long`.\n *\n * The resulting `ULong` value has the same binary representation as this `Long` value.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Long.toULong(): ULong = ULong(this)\n\n/**\n * Converts this [Float] value to [ULong].\n *\n * The fractional part, if any, is rounded down towards zero.\n * Returns zero if this `Float` value is negative or `NaN`, [ULong.MAX_VALUE] if it's bigger than `ULong.MAX_VALUE`.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Float.toULong(): ULong = doubleToULong(this.toDouble())\n/**\n * Converts this [Double] value to [ULong].\n *\n * The fractional part, if any, is rounded down towards
zero.\n * Returns zero if this `Double` value is negative or `NaN`, [ULong.MAX_VALUE] if it's bigger than `ULong.MAX_VALUE`.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun Double.toULong(): ULong = doubleToULong(this)\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"CollectionsKt\")\n\npackage kotlin.collections\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.random.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns 1st *element* from the list.\n * \n * Throws an [IndexOutOfBoundsException] if the size of this list is less
than 1.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> List<T>.component1(): T {\n   return get(0)\n}\n\n/**\n * Returns 2nd *element* from the list.\n * \n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 2.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> List<T>.component2(): T {\n   return get(1)\n}\n\n/**\n * Returns 3rd *element* from the list.\n * \n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 3.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> List<T>.component3(): T {\n   return get(2)\n}\n\n/**\n * Returns 4th *element* from the list.\n * \n * Throws an [IndexOutOfBoundsException] if the size of this list is less than 4.\n

```
*/\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> List<T>.component4(): T {\n    return get(3)\n}\n\n/**\n * Returns 5th *element* from the list.\n * \n * Throws an [IndexOutOfBoundsException] if the size of this list is less
 than 5.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> List<T>.component5(): T {\n    return get(4)\n}\n\n/**\n * Returns `true` if [element] is found in the collection.\n */\npublic operator fun <@kotlin.internal.OnlyInputTypes T> Iterable<T>.contains(element: T): Boolean {\n    if (this is Collection)\n return contains(element)\n    return indexOf(element) >= 0\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this collection.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic fun <T> Iterable<T>.elementAt(index: Int): T {\n if (this is List)\n        return get(index)\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"Collection doesn't contain element at index $index.\") }\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this list.\n * \n * @sample
 samples.collections.Collections.Elements.elementAt\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> List<T>.elementAt(index: Int): T {\n    return get(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this collection.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n */\npublic fun <T> Iterable<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\n    if (this is List)\n        return this.getOrElse(index, defaultValue)\n    if (index < 0)\n        return defaultValue(index)\n    val iterator = iterator()\n    var count = 0\n    while (iterator.hasNext()) {\n        val element = iterator.next()\n        if (index == count++)\n            return element\n    }\n    return defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this list.\n
 * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n
 */\n@kotlin.internal.InlineOnly\npublic inline fun <T> List<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\n    return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this collection.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\npublic fun <T> Iterable<T>.elementAtOrNull(index: Int): T? {\n    if (this is List)\n        return this.getOrNull(index)\n    if (index < 0)\n        return null\n    val iterator = iterator()\n    var count = 0\n    while (iterator.hasNext()) {\n        val element = iterator.next()\n        if (index == count++)\n            return element\n    }\n    return null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this list.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n
 */\n@kotlin.internal.InlineOnly\npublic inline fun <T> List<T>.elementAtOrNull(index: Int): T? {\n    return this.getOrNull(index)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n
 */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.find(predicate: (T) -> Boolean): T? {\n    return firstOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n
 */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.findLast(predicate: (T) -> Boolean): T? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n
 */\n@kotlin.internal.InlineOnly\npublic
 inline fun <T> List<T>.findLast(predicate: (T) -> Boolean): T? {\n    return lastOrNull(predicate)\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\npublic fun <T> Iterable<T>.first(): T {\n    when (this) {\n        is List -> return this.first()\n        else -> {\n            val iterator = iterator()\n            if (!iterator.hasNext())\n                throw NoSuchElementException(\"Collection is empty.\")\n            return iterator.next()\n        }\n    }\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the list is empty.\n */\npublic fun <T> List<T>.first(): T {\n    if (isEmpty())\n
```

throw NoSuchElementException(\"List is empty.\")\n return this[0]\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */\npublic inline fun <T> Iterable<T>.first(predicate: (T) -> Boolean):
T {\n for (element in this) if (predicate(element)) return element\n throw NoSuchElementException(\"Collection contains no element matching the predicate.\")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to elements of this collection in iteration order,\n * or throws [NoSuchElementException] if no non-null value was produced.\n * \n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any> Iterable<T>.firstNotNullOf(transform: (T) -> R?): R {\n return firstNotNullOfOrNull(transform) ?: throw NoSuchElementException(\"No element of the collection was transformed to a non-null value.\")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to elements of this collection in iteration order,\n * or `null` if no non-null value was produced.\n * \n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any> Iterable<T>.firstNotNullOfOrNull(transform: (T) -> R?): R? {\n for (element in this) {\n val result = transform(element)\n if (result != null) {\n return result\n }\n }\n return null\n}\n\n/**\n * Returns the first element, or `null` if the collection is empty.\n */\npublic fun <T> Iterable<T>.firstOrNull(): T? {\n when (this) {\n is List -> {\n if (isEmpty())\n return null\n else\n return this[0]\n }\n else -> {\n val iterator = iterator()\n if (!iterator.hasNext())\n return null\n return iterator.next()\n }\n }\n}\n\n/**\n * Returns the first element, or `null` if the list is empty.\n */\npublic fun <T> List<T>.firstOrNull(): T? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns
the first element matching the given [predicate], or `null` if element was not found.\n */\npublic inline fun <T> Iterable<T>.firstOrNull(predicate: (T) -> Boolean): T? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this list.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> List<T>.getOrElse(index: Int, defaultValue: (Int) -> T): T {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this list.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n */\npublic fun <T> List<T>.getOrNull(index: Int): T? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns first index of [element], or -1 if the collection does
not contain element.\n */\npublic fun <@kotlin.internal.OnlyInputTypes T> Iterable<T>.indexOf(element: T): Int {\n if (this is List) return this.indexOf(element)\n var index = 0\n for (item in this) {\n checkIndexOverflow(index)\n if (element == item)\n return index\n index++\n }\n return -1\n}\n\n/**\n * Returns first index of [element], or -1 if the list does not contain element.\n */\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\") // false warning, extension takes precedence in some cases\npublic fun <@kotlin.internal.OnlyInputTypes T> List<T>.indexOf(element: T): Int {\n return indexOf(element)\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the collection does not contain such element.\n */\npublic inline fun <T> Iterable<T>.indexOfFirst(predicate: (T) -> Boolean): Int {\n var index = 0\n for (item in this) {\n checkIndexOverflow(index)\n if (predicate(item))\n return index\n index++\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the list does not contain such element.\n */\npublic inline fun <T> List<T>.indexOfFirst(predicate: (T) -> Boolean): Int {\n var index = 0\n for (item in this) {\n if (predicate(item))\n return index\n index++\n }\n return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the collection does not contain such element.\n */\npublic inline fun <T> Iterable<T>.indexOfLast(predicate: (T) -> Boolean): Int {\n var lastIndex = -1\n var index = 0\n for (item in this) {\n checkIndexOverflow(index)\n if (predicate(item))\n lastIndex = index\n index++\n

}\n   return lastIndex\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the list does not contain such element.\n */\npublic inline fun <T> List<T>.indexOfLast(predicate:
 (T) -> Boolean): Int {\n   val iterator = this.listIterator(size)\n   while (iterator.hasPrevious()) {\n   if (predicate(iterator.previous())) {\n   return iterator.nextIndex()\n   }\n   }\n   return -1\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the collection is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun <T> Iterable<T>.last(): T {\n   when (this) {\n   is List -> return this.last()\n   else -> {\n   val iterator = iterator()\n   if (!iterator.hasNext())\n throw NoSuchElementException(\"Collection is empty.\")\n   var last = iterator.next()\n   while (iterator.hasNext())\n   last = iterator.next()\n   return last\n   }\n   }\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the list is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n
 */\npublic fun <T> List<T>.last(): T {\n   if (isEmpty())\n   throw NoSuchElementException(\"List is empty.\")\n   return this[lastIndex]\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun <T> Iterable<T>.last(predicate: (T) -> Boolean): T {\n   var last: T? = null\n   var found = false\n   for (element in this) {\n   if (predicate(element)) {\n   last = element\n   found = true\n   }\n   }\n   if (!found) throw NoSuchElementException(\"Collection contains no element matching the predicate.\")\n @Suppress(\"UNCHECKED_CAST\")\n   return last as T\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n
 */\npublic inline fun <T> List<T>.last(predicate: (T) -> Boolean): T {\n   val iterator = this.listIterator(size)\n   while (iterator.hasPrevious()) {\n   val element = iterator.previous()\n   if (predicate(element)) return element\n   }\n   throw NoSuchElementException(\"List contains no element matching the predicate.\")\n}\n\n/**\n * Returns last index of [element], or -1 if the collection does not contain element.\n */\npublic fun <@kotlin.internal.OnlyInputTypes T> Iterable<T>.lastIndexOf(element: T): Int {\n   if (this is List) return this.lastIndexOf(element)\n   var lastIndex = -1\n   var index = 0\n   for (item in this) {\n checkIndexOverflow(index)\n   if (element == item)\n   lastIndex = index\n   index++\n   }\n   return lastIndex\n}\n\n/**\n * Returns last index of [element], or -1 if the list does not contain element.\n */\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\") // false warning, extension takes precedence in some cases\npublic
 fun <@kotlin.internal.OnlyInputTypes T> List<T>.lastIndexOf(element: T): Int {\n   return lastIndexOf(element)\n}\n\n/**\n * Returns the last element, or `null` if the collection is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun <T> Iterable<T>.lastOrNull(): T? {\n   when (this) {\n   is List -> return if (isEmpty()) null else this[size - 1]\n   else -> {\n   val iterator = iterator()\n if (!iterator.hasNext())\n   return null\n   var last = iterator.next()\n   while (iterator.hasNext())\n   last = iterator.next()\n   return last\n   }\n   }\n}\n\n/**\n * Returns the last element, or `null` if the list is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun <T> List<T>.lastOrNull(): T? {\n   return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null`
 if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun <T> Iterable<T>.lastOrNull(predicate: (T) -> Boolean): T? {\n   var last: T? = null\n   for (element in this) {\n   if (predicate(element)) {\n   last = element\n   }\n   }\n   return last\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun <T> List<T>.lastOrNull(predicate: (T) -> Boolean): T? {\n   val iterator = this.listIterator(size)\n   while (iterator.hasPrevious()) {\n   val element = iterator.previous()\n   if (predicate(element)) return element\n   }\n   return null\n}\n\n/**\n * Returns a random element from this collection.\n * \n * @throws NoSuchElementException if this collection is empty.\n

```
*/\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Collection<T>.random():
 T {\n    return random(Random)\n}\n\n/**\n * Returns a random element from this collection using the specified
source of randomness.\n * \n * @throws NoSuchElementException if this collection is empty.\n
*/\n@SinceKotlin(\"1.3\")\npublic fun <T> Collection<T>.random(random: Random): T {\n    if (isEmpty())\n
throw NoSuchElementException(\"Collection is empty.\")\n    return elementAt(random.nextInt(size))\n}\n\n/**\n *
Returns a random element from this collection, or `null` if this collection is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun <T> Collection<T>.randomOrNull(): T? {\n    return randomOrNull(Random)\n}\n\n/**\n * Returns a
random element from this collection using the specified source of randomness, or `null` if this collection is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T>
Collection<T>.randomOrNull(random: Random):
 T? {\n    if (isEmpty())\n        return null\n    return elementAt(random.nextInt(size))\n}\n\n/**\n * Returns the
single element, or throws an exception if the collection is empty or has more than one element.\n */\npublic fun <T>
Iterable<T>.single(): T {\n    when (this) {\n        is List -> return this.single()\n        else -> {\n            val iterator =
iterator()\n            if (!iterator.hasNext())\n                throw NoSuchElementException(\"Collection is empty.\")\n
    val single = iterator.next()\n            if (iterator.hasNext())\n                throw
IllegalArgumentException(\"Collection has more than one element.\")\n            return single\n        }\n
}\n}\n\n/**\n * Returns the single element, or throws an exception if the list is empty or has more than one
element.\n */\npublic fun <T> List<T>.single(): T {\n    return when (size) {\n        0 -> throw
NoSuchElementException(\"List is empty.\")\n        1 -> this[0]\n        else -> throw
 IllegalArgumentException(\"List has more than one element.\")\n    }\n}\n\n/**\n * Returns the single element
matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\npublic
inline fun <T> Iterable<T>.single(predicate: (T) -> Boolean): T {\n    var single: T? = null\n    var found = false\n
for (element in this) {\n        if (predicate(element)) {\n            if (found) throw
IllegalArgumentException(\"Collection contains more than one matching element.\")\n            single = element\n
    found = true\n        }\n    }\n    if (!found) throw NoSuchElementException(\"Collection contains no element
matching the predicate.\")\n    @Suppress(\"UNCHECKED_CAST\")\n    return single as T\n}\n\n/**\n * Returns
single element, or `null` if the collection is empty or has more than one element.\n */\npublic fun <T>
Iterable<T>.singleOrNull(): T? {\n    when (this) {\n        is List -> return if (size == 1) this[0] else null\n
    else -> {\n            val iterator = iterator()\n            if (!iterator.hasNext())\n                return null\n            val single
= iterator.next()\n            if (iterator.hasNext())\n                return null\n            return single\n        }\n
}\n}\n\n/**\n * Returns single element, or `null` if the list is empty or has more than one element.\n */\npublic fun
<T> List<T>.singleOrNull(): T? {\n    return if (size == 1) this[0] else null\n}\n\n/**\n * Returns the single element
matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\npublic
inline fun <T> Iterable<T>.singleOrNull(predicate: (T) -> Boolean): T? {\n    var single: T? = null\n    var found =
false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single =
element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns a list
containing
 all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n *
@sample samples.collections.Collections.Transformations.drop\n */\npublic fun <T> Iterable<T>.drop(n: Int):
List<T> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n == 0) return toList()\n
val list: ArrayList<T>\n    if (this is Collection<*>) {\n        val resultSize = size - n\n        if (resultSize <= 0)\n
  return emptyList()\n        if (resultSize == 1)\n            return listOf(last())\n        list = ArrayList<T>(resultSize)\n
  if (this is List<T>) {\n            if (this is RandomAccess) {\n                for (index in n until size)\n
list.add(this[index])\n            } else {\n                for (item in listIterator(n))\n                    list.add(item)\n            }\n
        return list\n        }\n    }\n    else {\n        list = ArrayList<T>()\n    }\n    var count =
 0\n    for (item in this) {\n        if (count >= n) list.add(item) else ++count\n    }\n    return
list.optimizeReadOnlyList()\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n *
```

@throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun <T> List<T>.dropLast(n: Int): List<T> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun <T> List<T>.dropLastWhile(predicate: (T) -> Boolean): List<T> {\n if (!isEmpty()) {\n val iterator = listIterator(size)\n while (iterator.hasPrevious()) {\n if (!predicate(iterator.previous())) {\n return take(iterator.nextIndex() + 1)\n }\n }\n }\n return emptyList()\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic inline fun <T> Iterable<T>.dropWhile(predicate: (T) -> Boolean): List<T> {\n var yielding = false\n val list = ArrayList<T>()\n for (item in this)\n if (yielding)\n list.add(item)\n else if (!predicate(item)) {\n list.add(item)\n yielding = true\n }\n return list\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun <T> Iterable<T>.filter(predicate: (T) -> Boolean): List<T> {\n return filterTo(ArrayList<T>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun <T> Iterable<T>.filterIndexed(predicate: (index: Int, T) -> Boolean): List<T> {\n return filterIndexedTo(ArrayList<T>(), predicate)\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <T, C : MutableCollection<in T>> Iterable<T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C {\n forEachIndexed { index, element ->\n if (predicate(index, element)) destination.add(element)\n }\n return destination\n}\n\n/**\n * Returns a list containing all elements
that are instances of specified type parameter R.\n * \n * @sample samples.collections.Collections.Filtering.filterIsInstance\n */\npublic inline fun <reified R> Iterable<*>.filterIsInstance(): List<@kotlin.internal.NoInfer R> {\n return filterIsInstanceTo(ArrayList<R>())\n}\n\n/**\n * Appends all elements that are instances of specified type parameter R to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterIsInstanceTo\n */\npublic inline fun <reified R, C : MutableCollection<in R>> Iterable<*>.filterIsInstanceTo(destination: C): C {\n for (element in this) if (element is R) destination.add(element)\n return destination\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic inline fun <T> Iterable<T>.filterNot(predicate: (T) -> Boolean): List<T> {\n return filterNotTo(ArrayList<T>(), predicate)\n}\n\n/**\n * Returns a list
containing all elements that are not `null`.\n * \n * @sample samples.collections.Collections.Filtering.filterNotNull\n */\npublic fun <T : Any> Iterable<T?>.filterNotNull(): List<T> {\n return filterNotNullTo(ArrayList<T>())\n}\n\n/**\n * Appends all elements that are not `null` to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterNotNullTo\n */\npublic fun <C : MutableCollection<in T>, T : Any> Iterable<T?>.filterNotNullTo(destination: C): C {\n for (element in this) if (element != null) destination.add(element)\n return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <T, C : MutableCollection<in T>> Iterable<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {\n for (element in this) if (!predicate(element)) destination.add(element)\n return destination\n}\n\n/**\n

```
 * Appends all elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <T, C : MutableCollection<in T>>
Iterable<T>.filterTo(destination: C, predicate: (T) -> Boolean): C {\n    for (element in this) if (predicate(element))
destination.add(element)\n    return destination\n}\n\n/**\n * Returns a list containing elements at indices in the
specified [indices] range.\n */\npublic fun <T> List<T>.slice(indices: IntRange): List<T> {\n    if
(indices.isEmpty()) return listOf()\n    return this.subList(indices.start, indices.endInclusive + 1).toList()\n}\n\n/**\n
* Returns a list containing elements at specified [indices].\n */\npublic fun <T> List<T>.slice(indices:
Iterable<Int>): List<T> {\n    val size = indices.collectionSizeOrDefault(10)\n    if (size == 0) return emptyList()\n
val list = ArrayList<T>(size)\n    for (index in indices) {\n        list.add(get(index))\n    }\n    return
 list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is
negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun <T>
Iterable<T>.take(n: Int): List<T> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    if (n
== 0) return emptyList()\n    if (this is Collection<T>) {\n        if (n >= size) return toList()\n        if (n == 1) return
listOf(first())\n    }\n    var count = 0\n    val list = ArrayList<T>(n)\n    for (item in this) {\n        list.add(item)\n
if (++count == n)\n            break\n    }\n    return list.optimizeReadOnlyList()\n}\n\n/**\n * Returns a list containing
last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic fun <T> List<T>.takeLast(n: Int): List<T> {\n
require(n >= 0) { \"Requested element count $n is less than zero.\"
 }\n    if (n == 0) return emptyList()\n    val size = size\n    if (n >= size) return toList()\n    if (n == 1) return
listOf(last())\n    val list = ArrayList<T>(n)\n    if (this is RandomAccess) {\n        for (index in size - n until size)\n
        list.add(this[index])\n    } else {\n        for (item in listIterator(size - n))\n            list.add(item)\n    }\n    return
list\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic inline fun <T> List<T>.takeLastWhile(predicate:
(T) -> Boolean): List<T> {\n    if (isEmpty())\n        return emptyList()\n    val iterator = listIterator(size)\n    while
(iterator.hasPrevious()) {\n        if (!predicate(iterator.previous())) {\n            iterator.next()\n            val
expectedSize = size - iterator.nextIndex()\n            if (expectedSize == 0) return emptyList()\n            return
ArrayList<T>(expectedSize).apply
 {\n                while (iterator.hasNext())\n                    add(iterator.next())\n            }\n        }\n    }\n    return
toList()\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample
samples.collections.Collections.Transformations.take\n */\npublic inline fun <T> Iterable<T>.takeWhile(predicate:
(T) -> Boolean): List<T> {\n    val list = ArrayList<T>()\n    for (item in this) {\n        if (!predicate(item))\n
break\n        list.add(item)\n    }\n    return list\n}\n\n/**\n * Reverses elements in the list in-place.\n */\npublic
expect fun <T> MutableList<T>.reverse(): Unit\n\n/**\n * Returns a list with elements in reversed order.\n
*/\npublic fun <T> Iterable<T>.reversed(): List<T> {\n    if (this is Collection && size <= 1) return toList()\n    val
list = toMutableList()\n    list.reverse()\n    return list\n}\n\n/**\n * Randomly shuffles elements in this list in-place
using the specified [random]
 instance as the source of randomness.\n * \n * See:
https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm\n
*/\n@SinceKotlin(\"1.3\")\npublic fun <T> MutableList<T>.shuffle(random: Random): Unit {\n    for (i in lastIndex
downTo 1) {\n        val j = random.nextInt(i + 1)\n        this[j] = this.set(i, this[j])\n    }\n}\n\n/**\n * Sorts elements
in the list in-place according to natural sort order of the value returned by specified [selector] function.\n * \n * The
sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic
inline fun <T, R : Comparable<R>> MutableList<T>.sortBy(crossinline selector: (T) -> R?): Unit {\n    if (size > 1)
sortWith(compareBy(selector))\n}\n\n/**\n * Sorts elements in the list in-place descending according to natural sort
order of the value returned by specified [selector] function.\n * \n * The sort is _stable_. It means that equal
elements preserve their order relative
 to each other after sorting.\n */\npublic inline fun <T, R : Comparable<R>>
MutableList<T>.sortByDescending(crossinline selector: (T) -> R?): Unit {\n    if (size > 1)
```

sortWith(compareByDescending(selector))\n}\n\n/**\n * Sorts elements in the list in-place descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> MutableList<T>.sortDescending(): Unit {\n sortWith(reverseOrder())\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> Iterable<T>.sorted(): List<T> {\n if (this is Collection) {\n if (size <= 1) return this.toList()\n @Suppress(\"UNCHECKED_CAST\")\n return (toTypedArray<Comparable<T>>() as Array<T>).apply {
 sort() }.asList()\n }\n return toMutableList().apply { sort() }\n}\n\n/**\n * Returns a list of all elements sorted according to natural sort order of the value returned by specified [selector] function.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n * \n * @sample samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <T, R : Comparable<R>> Iterable<T>.sortedBy(crossinline selector: (T) -> R?): List<T> {\n return sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a list of all elements sorted descending according to natural sort order of the value returned by specified [selector] function.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic inline fun <T, R : Comparable<R>> Iterable<T>.sortedByDescending(crossinline selector: (T) -> R?): List<T> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n
 * Returns a list of all elements sorted descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T : Comparable<T>> Iterable<T>.sortedDescending(): List<T> {\n return sortedWith(reverseOrder())\n}\n\n/**\n * Returns a list of all elements sorted according to the specified [comparator].\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\npublic fun <T> Iterable<T>.sortedWith(comparator: Comparator<in T>): List<T> {\n if (this is Collection) {\n if (size <= 1) return this.toList()\n @Suppress(\"UNCHECKED_CAST\")\n return (toTypedArray<Any?>() as Array<T>).apply { sortWith(comparator) }.asList()\n }\n return toMutableList().apply { sortWith(comparator) }\n}\n\n/**\n * Returns an array of Boolean containing all of the elements of this collection.\n */\npublic
 fun Collection<Boolean>.toBooleanArray(): BooleanArray {\n val result = BooleanArray(size)\n var index = 0\n for (element in this)\n result[index++] = element\n return result\n}\n\n/**\n * Returns an array of Byte containing all of the elements of this collection.\n */\npublic fun Collection<Byte>.toByteArray(): ByteArray {\n val result = ByteArray(size)\n var index = 0\n for (element in this)\n result[index++] = element\n return result\n}\n\n/**\n * Returns an array of Char containing all of the elements of this collection.\n */\npublic fun Collection<Char>.toCharArray(): CharArray {\n val result = CharArray(size)\n var index = 0\n for (element in this)\n result[index++] = element\n return result\n}\n\n/**\n * Returns an array of Double containing all of the elements of this collection.\n */\npublic fun Collection<Double>.toDoubleArray(): DoubleArray {\n val result = DoubleArray(size)\n var index = 0\n for (element
 in this)\n result[index++] = element\n return result\n}\n\n/**\n * Returns an array of Float containing all of the elements of this collection.\n */\npublic fun Collection<Float>.toFloatArray(): FloatArray {\n val result = FloatArray(size)\n var index = 0\n for (element in this)\n result[index++] = element\n return result\n}\n\n/**\n * Returns an array of Int containing all of the elements of this collection.\n */\npublic fun Collection<Int>.toIntArray(): IntArray {\n val result = IntArray(size)\n var index = 0\n for (element in this)\n result[index++] = element\n return result\n}\n\n/**\n * Returns an array of Long containing all of the elements of this collection.\n */\npublic fun Collection<Long>.toLongArray(): LongArray {\n val result = LongArray(size)\n var index = 0\n for (element in this)\n result[index++] = element\n return result\n}\n\n/**\n * Returns an array of Short containing all of the elements of this
 collection.\n */\npublic fun Collection<Short>.toShortArray(): ShortArray {\n val result = ShortArray(size)\n var index = 0\n for (element in this)\n result[index++] = element\n return result\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to elements of the given collection.\n

* \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original collection.\n * \n * @sample samples.collections.Collections.Transformations.associate\n */\npublic inline fun <T, K, V> Iterable<T>.associate(transform: (T) -> Pair<K, V>): Map<K, V> {\n val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)\n return associateTo(LinkedHashMap<K, V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing the elements from the given collection indexed by the key\n * returned from [keySelector] function applied to
each element.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original collection.\n * \n * @sample samples.collections.Collections.Transformations.associateBy\n */\npublic inline fun <T, K> Iterable<T>.associateBy(keySelector: (T) -> K): Map<K, T> {\n val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)\n return associateByTo(LinkedHashMap<K, T>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given collection.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original collection.\n * \n * @sample samples.collections.Collections.Transformations.associateByWithValueTransform\n
*/\npublic inline fun <T, K, V> Iterable<T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {\n val capacity = mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16)\n return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given collection\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Collections.Transformations.associateByTo\n */\npublic inline fun <T, K, M : MutableMap<in K, in T>> Iterable<T>.associateByTo(destination: M, keySelector: (T) -> K): M {\n for (element in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination]
mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given collection.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample samples.collections.Collections.Transformations.associateByToWithValueTransform\n */\npublic inline fun <T, K, V, M : MutableMap<in K, in V>> Iterable<T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by [transform] function applied to each element of the given collection.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample samples.collections.Collections.Transformations.associateTo\n
*/\npublic inline fun <T, K, V, M : MutableMap<in K, in V>> Iterable<T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n }\n return destination\n}\n\n/**\n * Returns a [Map] where keys are elements from the given collection and values are\n * produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original collection.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin(\"1.3\")\npublic inline fun <K, V> Iterable<K>.associateWith(valueSelector: (K) -> V): Map<K, V> {\n val result = LinkedHashMap<K, V>(mapCapacity(collectionSizeOrDefault(10)).coerceAtLeast(16))\n return associateWithTo(result, valueSelector)\n}\n\n/**\n * Populates and returns
the [destination] mutable map with key-value pairs for each element of the given collection,\n * where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements

are equal, the last one overwrites the former value in the map.\n * \n * @sample samples.collections.Collections.Transformations.associateWithTo\n */\n@SinceKotlin(\"1.3\")\npublic inline fun <K, V, M : MutableMap<in K, in V>> Iterable<K>.associateWithTo(destination: M, valueSelector: (K) -> V): M {\n for (element in this) {\n destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n/**\n * Appends all elements to the given [destination] collection.\n */\npublic fun <T, C : MutableCollection<in T>> Iterable<T>.toCollection(destination: C): C {\n for (item in this) {\n destination.add(item)\n }\n return destination\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n */\npublic fun <T> Iterable<T>.toHashSet(): HashSet<T> {\n return toCollection(HashSet<T>(mapCapacity(collectionSizeOrDefault(12))))\n}\n\n/**\n * Returns a [List] containing all elements.\n */\npublic fun <T> Iterable<T>.toList(): List<T> {\n if (this is Collection) {\n return when (size) {\n 0 -> emptyList()\n 1 -> listOf(if (this is List) get(0) else iterator().next())\n else -> this.toMutableList()\n }\n }\n return this.toMutableList().optimizeReadOnlyList()\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this collection.\n */\npublic fun <T> Iterable<T>.toMutableList(): MutableList<T> {\n if (this is Collection<T>)\n return this.toMutableList()\n return toCollection(ArrayList<T>())\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this collection.\n */\npublic fun <T> Collection<T>.toMutableList(): MutableList<T> {\n return ArrayList(this)\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original collection.\n */\npublic fun <T> Iterable<T>.toSet(): Set<T> {\n if (this is Collection) {\n return when (size) {\n 0 -> emptySet()\n 1 -> setOf(if (this is List) this[0] else iterator().next())\n else -> toCollection(LinkedHashSet<T>(mapCapacity(size)))\n }\n }\n return toCollection(LinkedHashSet<T>()).optimizeReadOnlySet()\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original collection.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <T, R> Iterable<T>.flatMap(transform: (T) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original collection.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapSequence\")\npublic inline fun <T, R> Iterable<T>.flatMap(transform: (T) -> Sequence<R>): List<R> {\n return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original collection.\n * \n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterable\")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Iterable<T>.flatMapIndexed(transform: (index: Int, T) -> Iterable<R>): List<R> {\n return flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original collection.\n * \n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedSequence\")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Iterable<T>.flatMapIndexed(transform: (index: Int, T) -> Sequence<R>): List<R> {\n return flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original collection, to the given [destination].\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterableTo\")\n@kotlin.internal.InlineOnly\npublic

c inline
fun <T, R, C : MutableCollection<in R>> Iterable<T>.flatMapIndexedTo(destination: C, transform: (index: Int, T) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(checkIndexOverflow(index++), element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original collection, to the given [destination].\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedSequenceTo\")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.flatMapIndexedTo(destination: C, transform: (index: Int, T) -> Sequence<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(checkIndexOverflow(index++), element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original collection, to the given [destination].\n */\npublic inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.flatMapTo(destination: C, transform: (T) -> Iterable<R>): C {\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original collection, to the given [destination].\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapSequenceTo\")\npublic inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C {\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Groups elements of the original collection by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original collection.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <T, K> Iterable<T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {\n return groupByTo(LinkedHashMap<K, MutableList<T>>(), keySelector)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original collection\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration
order of the keys produced from the original collection.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <T, K, V> Iterable<T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>> {\n return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups elements of the original collection by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <T, K, M : MutableMap<in K, MutableList<T>>> Iterable<T>.groupByTo(destination: M, keySelector: (T) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<T>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original collection\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <T, K, V, M : MutableMap<in K, MutableList<V>>> Iterable<T>.groupByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =

destination.getOrPut(key) { ArrayList<V>() }\n   list.add(valueTransform(element))\n   }\n   return destination\n}\n\n/**\n * Creates a [Grouping] source from a collection to be used later with one of group-and-fold operations\n * using the specified [keySelector] function to
extract a key from each element.\n * \n * @sample samples.collections.Grouping.groupingByEachCount\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <T, K> Iterable<T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> {\n   return object : Grouping<T, K> {\n   override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()\n   override fun keyOf(element: T): K = keySelector(element)\n   }\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original collection.\n * \n * @sample samples.collections.Collections.Transformations.map\n */\npublic inline fun <T, R> Iterable<T>.map(transform: (T) -> R): List<R> {\n   return mapTo(ArrayList<R>(collectionSizeOrDefault(10)), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original collection.\n * @param [transform] function that takes the index of an
element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <T, R> Iterable<T>.mapIndexed(transform: (index: Int, T) -> R): List<R> {\n   return mapIndexedTo(ArrayList<R>(collectionSizeOrDefault(10)), transform)\n}\n\n/**\n * Returns a list containing only the non-null results of applying the given [transform] function\n * to each element and its index in the original collection.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <T, R : Any> Iterable<T>.mapIndexedNotNull(transform: (index: Int, T) -> R?): List<R> {\n   return mapIndexedNotNullTo(ArrayList<R>(), transform)\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original collection\n * and appends only the non-null results to the given [destination].\n * @param [transform] function that takes
the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <T, R : Any, C : MutableCollection<in R>> Iterable<T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {\n   forEachIndexed { index, element -> transform(index, element)?.let { destination.add(it) } }\n   return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original collection\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\npublic inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.mapIndexedTo(destination: C, transform: (index: Int, T) -> R): C {\n   var index = 0\n   for (item in this)\n   destination.add(transform(checkIndexOverflow(index++), item))\n   return destination\n}\n\n/**\n
* Returns a list containing only the non-null results of applying the given [transform] function\n * to each element in the original collection.\n * \n * @sample samples.collections.Collections.Transformations.mapNotNull\n */\npublic inline fun <T, R : Any> Iterable<T>.mapNotNull(transform: (T) -> R?): List<R> {\n   return mapNotNullTo(ArrayList<R>(), transform)\n}\n\n/**\n * Applies the given [transform] function to each element in the original collection\n * and appends only the non-null results to the given [destination].\n */\npublic inline fun <T, R : Any, C : MutableCollection<in R>> Iterable<T>.mapNotNullTo(destination: C, transform: (T) -> R?): C {\n forEach { element -> transform(element)?.let { destination.add(it) } }\n   return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original collection\n * and appends the results to the given [destination].\n */\npublic inline fun <T, R, C : MutableCollection<in R>> Iterable<T>.mapTo(destination:
C, transform: (T) -> R): C {\n   for (item in this)\n   destination.add(transform(item))\n   return destination\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original collection\n * into an [IndexedValue] containing the index of that element and the element itself.\n */\npublic fun <T> Iterable<T>.withIndex(): Iterable<IndexedValue<T>> {\n   return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a list containing only distinct elements from the given collection.\n * \n * Among equal elements of the given collection, only the first one will be present in the resulting list.\n * The elements in the resulting list are in the same order as they were in the source collection.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun <T> Iterable<T>.distinct():

```
List<T> {\n    return this.toMutableSet().toList()\n}\n\n/**\n * Returns a list containing only elements from the given collection\n * having
 distinct keys returned by the given [selector] function.\n * \n * Among elements of the given collection with equal keys, only the first one will be present in the resulting list.\n * The elements in the resulting list are in the same order as they were in the source collection.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic inline fun <T, K> Iterable<T>.distinctBy(selector: (T) -> K): List<T> {\n    val set = HashSet<K>()\n    val list = ArrayList<T>()\n    for (e in this) {\n        val key = selector(e)\n        if (set.add(key))\n            list.add(e)\n    }\n    return list\n}\n\n/**\n * Returns a set containing all elements that are contained by both this collection and the specified collection.\n * \n * The returned set preserves the element iteration order of the original collection.\n * \n * To get a set containing all elements that are contained at least in one of these collections use [union].\n */\npublic infix fun
 <T> Iterable<T>.intersect(other: Iterable<T>): Set<T> {\n    val set = this.toMutableSet()\n    set.retainAll(other)\n    return set\n}\n\n/**\n * Returns a set containing all elements that are contained by this collection and not contained by the specified collection.\n * \n * The returned set preserves the element iteration order of the original collection.\n */\npublic infix fun <T> Iterable<T>.subtract(other: Iterable<T>): Set<T> {\n    val set = this.toMutableSet()\n    set.removeAll(other)\n    return set\n}\n\n/**\n * Returns a new [MutableSet] containing all distinct elements from the given collection.\n * \n * The returned set preserves the element iteration order of the original collection.\n */\npublic fun <T> Iterable<T>.toMutableSet(): MutableSet<T> {\n    return when (this) {\n
 is Collection<T> -> LinkedHashSet(this)\n        else -> toCollection(LinkedHashSet<T>())\n    }\n}\n\n/**\n * Returns a set containing all distinct elements from both collections.\n * \n *
 The returned set preserves the element iteration order of the original collection.\n * Those elements of the [other] collection that are unique are iterated in the end\n * in the order of the [other] collection.\n * \n * To get a set containing all elements that are contained in both collections use [intersect].\n */\npublic infix fun <T> Iterable<T>.union(other: Iterable<T>): Set<T> {\n    val set = this.toMutableSet()\n    set.addAll(other)\n    return set\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the collection contains no elements, the function returns `true`\n * because there are no elements in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in [\"Vacuous truth\"](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun <T> Iterable<T>.all(predicate: (T) -> Boolean): Boolean {\n    if (this is Collection
 && isEmpty()) return true\n    for (element in this) if (!predicate(element)) return false\n    return true\n}\n\n/**\n * Returns `true` if collection has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\npublic fun <T> Iterable<T>.any(): Boolean {\n    if (this is Collection) return !isEmpty()\n    return iterator().hasNext()\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun <T> Iterable<T>.any(predicate: (T) -> Boolean): Boolean {\n    if (this is Collection && isEmpty()) return false\n    for (element in this) if (predicate(element)) return true\n    return false\n}\n\n/**\n * Returns the number of elements in this collection.\n */\npublic fun <T> Iterable<T>.count(): Int {\n    if (this is Collection) return size\n    var count = 0\n    for (element in this) checkCountOverflow(++count)\n    return count\n}\n\n/**\n
 * Returns the number of elements in this collection.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Collection<T>.count(): Int {\n    return size\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n */\npublic inline fun <T> Iterable<T>.count(predicate: (T) -> Boolean): Int {\n    if (this is Collection && isEmpty()) return 0\n    var count = 0\n    for (element in this) if (predicate(element)) checkCountOverflow(++count)\n    return count\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the collection is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n */\npublic inline fun <T, R>
```

Iterable<T>.fold(initial: R, operation: (acc: R, T) -> R): R {\n var accumulator = initial\n for (element in this) accumulator = operation(accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original collection.\n * \n * Returns the specified [initial] value if the collection is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n */\npublic inline fun <T, R> Iterable<T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {\n var index = 0\n var accumulator = initial\n for (element in this) accumulator = operation(checkIndexOverflow(index++), accumulator, element)\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the list is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <T, R> List<T>.foldRight(initial: R, operation: (T, acc: R) -> R): R {\n var accumulator = initial\n if (!isEmpty()) {\n val iterator = listIterator(size)\n while (iterator.hasPrevious()) {\n accumulator = operation(iterator.previous(), accumulator)\n }\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original list and current accumulator value.\n * \n * Returns the specified [initial] value if the list is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <T, R> List<T>.foldRightIndexed(initial: R, operation: (index: Int, T, acc: R) -> R): R {\n var accumulator = initial\n if (!isEmpty()) {\n val iterator = listIterator(size)\n while (iterator.hasPrevious()) {\n val index = iterator.previousIndex()\n accumulator = operation(index, iterator.previous(), accumulator)\n }\n }\n return accumulator\n}\n\n/**\n * Performs the given [action] on each element.\n */\n@kotlin.internal.HidesMembers\npublic inline fun <T> Iterable<T>.forEach(action: (T) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\npublic inline fun <T> Iterable<T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {\n var index = 0\n for (item in this) action(checkIndexOverflow(index++), item)\n}\n\n/**\n * Returns the largest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun Iterable<Double>.max(): Double {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var max = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n * Returns the largest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun Iterable<Float>.max(): Float {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var max = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n max = maxOf(max, e)\n }\n return max\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun <T : Comparable<T>> Iterable<T>.max(): T {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var max = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the collection is empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <T, R : Comparable<R>> Iterable<T>.maxBy(selector:

(T) -> R): T {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var maxElem = iterator.next()\n if (!iterator.hasNext()) return maxElem\n var maxValue = selector(maxElem)\n do {\n val e = iterator.next()\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n } while (iterator.hasNext())\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <T, R : Comparable<R>> Iterable<T>.maxByOrNull(selector: (T) -> R): T? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var maxElem = iterator.next()\n if (!iterator.hasNext()) return maxElem\n var maxValue = selector(maxElem)\n do {\n val e = iterator.next()\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n } while (iterator.hasNext())\n return maxElem\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the collection.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.maxOf(selector: (T) -> Double): Double {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var maxValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the collection.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.maxOf(selector: (T) -> Float): Float {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var maxValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the collection.\n * \n * @throws NoSuchElementException if the
collection is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Iterable<T>.maxOf(selector: (T) -> R): R {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var maxValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the collection or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <T> Iterable<T>.maxOfOrNull(selector: (T) -> Double): Double? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var maxValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the collection or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.maxOfOrNull(selector: (T)

-> Float): Float? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var maxValue = selector(iterator.next())\n   while (iterator.hasNext())
 {\n   val v = selector(iterator.next())\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the collection or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Iterable<T>.maxOfOrNull(selector: (T) -> R): R? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var maxValue = selector(iterator.next())\n   while (iterator.hasNext()) {\n   val v = selector(iterator.next())\n   if (maxValue < v) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the collection.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Iterable<T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n   val iterator = iterator()\n   if (!iterator.hasNext()) throw NoSuchElementException()\n   var maxValue = selector(iterator.next())\n   while (iterator.hasNext()) {\n   val v = selector(iterator.next())\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the collection or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
 inline fun <T, R> Iterable<T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var maxValue = selector(iterator.next())\n   while (iterator.hasNext()) {\n   val v = selector(iterator.next())\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic fun Iterable<Double>.maxOrNull(): Double? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var max = iterator.next()\n   while (iterator.hasNext()) {\n   val e = iterator.next()\n   max = maxOf(max, e)\n   }\n   return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is
 `NaN` returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic fun Iterable<Float>.maxOrNull(): Float? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var max = iterator.next()\n   while (iterator.hasNext()) {\n   val e = iterator.next()\n   max = maxOf(max, e)\n   }\n   return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T : Comparable<T>> Iterable<T>.maxOrNull(): T? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var max = iterator.next()\n   while (iterator.hasNext()) {\n   val e = iterator.next()\n   if (max < e) max = e\n   }\n   return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic
 fun <T> Iterable<T>.maxWith(comparator: Comparator<in T>): T {\n   val iterator = iterator()\n   if (!iterator.hasNext()) throw NoSuchElementException()\n   var max = iterator.next()\n   while (iterator.hasNext()) {\n   val e = iterator.next()\n   if (comparator.compare(max, e) < 0) max = e\n   }\n   return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T> Iterable<T>.maxWithOrNull(comparator: Comparator<in T>): T? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var max = iterator.next()\n   while

(iterator.hasNext()) {\n val e = iterator.next()\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic
fun Iterable<Double>.min(): Double {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun Iterable<Float>.min(): Float {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException
if the collection is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun <T : Comparable<T>> Iterable<T>.min(): T {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the collection is empty.\n * \n * @sample samples.collections.Collections.Aggregates.minBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <T, R : Comparable<R>> Iterable<T>.minBy(selector: (T) -> R): T {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var minElem = iterator.next()\n if (!iterator.hasNext())
return minElem\n var minValue = selector(minElem)\n do {\n val e = iterator.next()\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n } while (iterator.hasNext())\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <T, R : Comparable<R>> Iterable<T>.minByOrNull(selector: (T) -> R): T? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var minElem = iterator.next()\n if (!iterator.hasNext()) return minElem\n var minValue = selector(minElem)\n do {\n val e = iterator.next()\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n } while (iterator.hasNext())\n return
minElem\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the collection.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOf(selector: (T) -> Double): Double {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the collection.\n * \n * If any of values produced
by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOf(selector: (T) ->

Float): Float {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the collection.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Iterable<T>.minOf(selector: (T) -> R): R {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the collection or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOfOrNull(selector: (T) -> Double): Double? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the collection or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minOfOrNull(selector: (T) -> Float): Float? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the collection or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Iterable<T>.minOfOrNull(selector: (T) -> R): R? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the collection.\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Iterable<T>.minOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the collection or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Iterable<T>.minOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n val iterator = iterator()\n

if (!iterator.hasNext()) return null\n   var minValue = selector(iterator.next())\n   while (iterator.hasNext()) {\n val v = selector(iterator.next())\n   if (comparator.compare(minValue, v) > 0) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic fun Iterable<Double>.minOrNull(): Double? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var min = iterator.next()\n   while (iterator.hasNext()) {\n   val e = iterator.next()\n   min = minOf(min, e)\n   }\n   return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic fun Iterable<Float>.minOrNull(): Float? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return
null\n   var min = iterator.next()\n   while (iterator.hasNext()) {\n   val e = iterator.next()\n   min = minOf(min, e)\n   }\n   return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T : Comparable<T>> Iterable<T>.minOrNull(): T? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var min = iterator.next()\n   while (iterator.hasNext()) {\n val e = iterator.next()\n   if (min > e) min = e\n   }\n   return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the collection is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun <T> Iterable<T>.minWith(comparator: Comparator<in T>): T {\n   val iterator = iterator()\n   if (!iterator.hasNext()) throw NoSuchElementException()\n   var
min = iterator.next()\n   while (iterator.hasNext()) {\n   val e = iterator.next()\n   if (comparator.compare(min, e) > 0) min = e\n   }\n   return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T> Iterable<T>.minWithOrNull(comparator: Comparator<in T>): T? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var min = iterator.next()\n   while (iterator.hasNext()) {\n   val e = iterator.next()\n   if (comparator.compare(min, e) > 0) min = e\n   }\n   return min\n}\n\n/**\n * Returns `true` if the collection has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\npublic fun <T> Iterable<T>.none(): Boolean {\n   if (this is Collection) return isEmpty()\n   return
!iterator().hasNext()\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun <T> Iterable<T>.none(predicate: (T) -> Boolean): Boolean {\n   if (this is Collection && isEmpty()) return true\n   for (element in this) if (predicate(element)) return false\n   return true\n}\n\n/**\n * Performs the given [action] on each element and returns the collection itself afterwards.\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <T, C : Iterable<T>> C.onEach(action: (T) -> Unit): C {\n   return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the collection itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <T, C : Iterable<T>> C.onEachIndexed(action: (index: Int, T) -> Unit): C {\n   return apply { forEachIndexed(action) }\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this collection is empty. If the collection can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun <S, T : S> Iterable<T>.reduce(operation: (acc: S, T) -> S): S {\n   val iterator = this.iterator()\n   if (!iterator.hasNext()) throw UnsupportedOperationException(\"Empty collection can't be reduced.\")\n   var accumulator: S = iterator.next()\n   while (iterator.hasNext()) {\n   accumulator = operation(accumulator, iterator.next())\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting
with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original collection.\n * \n * Throws an exception if this collection is empty. If the collection can

be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun <S, T : S> Iterable<T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {\n val iterator = this.iterator()\n if (!iterator.hasNext()) throw UnsupportedOperationException(\"Empty collection can't be reduced.\")\n var index = 1\n var accumulator: S = iterator.next()\n while (iterator.hasNext()) {\n accumulator = operation(checkIndexOverflow(index++),
accumulator, iterator.next())\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original collection.\n * \n * Returns `null` if the collection is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <S, T : S> Iterable<T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {\n val iterator = this.iterator()\n if (!iterator.hasNext()) return null\n var index = 1\n var accumulator: S = iterator.next()\n while (iterator.hasNext()) {\n accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the collection is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S> Iterable<T>.reduceOrNull(operation: (acc: S, T) -> S): S? {\n val iterator = this.iterator()\n if (!iterator.hasNext()) return null\n var accumulator: S = iterator.next()\n while (iterator.hasNext()) {\n accumulator = operation(accumulator, iterator.next())\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n
* Throws an exception if this list is empty. If the list can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun <S, T : S> List<T>.reduceRight(operation: (T, acc: S) -> S): S {\n val iterator = listIterator(size)\n if (!iterator.hasPrevious())\n throw UnsupportedOperationException(\"Empty list can't be reduced.\")\n var accumulator: S = iterator.previous()\n while (iterator.hasPrevious()) {\n accumulator = operation(iterator.previous(), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original list and current accumulator
value.\n * \n * Throws an exception if this list is empty. If the list can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun <S, T : S> List<T>.reduceRightIndexed(operation: (index: Int, T, acc: S) -> S): S {\n val iterator = listIterator(size)\n if (!iterator.hasPrevious())\n throw UnsupportedOperationException(\"Empty list can't be reduced.\")\n var accumulator: S = iterator.previous()\n while (iterator.hasPrevious()) {\n val index = iterator.previousIndex()\n accumulator = operation(index, iterator.previous(), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element
and applying [operation] from right to left\n * to each element with its index in the original list and current accumulator value.\n * \n * Returns `null` if the list is empty.\n * \n * @param [operation] function that takes the

index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <S, T : S> List<T>.reduceRightIndexedOrNull(operation: (index: Int, T, acc: S) -> S): S? {\n val iterator = listIterator(size)\n if (!iterator.hasPrevious())\n return null\n var accumulator: S = iterator.previous()\n while (iterator.hasPrevious()) {\n val index = iterator.previousIndex()\n accumulator = operation(index, iterator.previous(), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if the list is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S> List<T>.reduceRightOrNull(operation: (T, acc: S) -> S): S? {\n val iterator = listIterator(size)\n if (!iterator.hasPrevious())\n return null\n var accumulator: S = iterator.previous()\n while (iterator.hasPrevious()) {\n accumulator = operation(iterator.previous(), accumulator)\n }\n return accumulator\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed
to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <T, R> Iterable<T>.runningFold(initial: R, operation: (acc: R, T) -> R): List<R> {\n val estimatedSize = collectionSizeOrDefault(9)\n if (estimatedSize == 0) return listOf(initial)\n val result = ArrayList<R>(estimatedSize + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original collection and current accumulator
value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <T, R> Iterable<T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {\n val estimatedSize = collectionSizeOrDefault(9)\n if (estimatedSize == 0) return listOf(initial)\n val result = ArrayList<R>(estimatedSize + 1).apply { add(initial) }\n var index = 0\n var accumulator = initial\n for (element in this) {\n accumulator = operation(index++, accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list
containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this collection.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and the element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S> Iterable<T>.runningReduce(operation: (acc: S, T) -> S): List<S> {\n val iterator = this.iterator()\n if (!iterator.hasNext()) return emptyList()\n var accumulator: S = iterator.next()\n val result = ArrayList<S>(collectionSizeOrDefault(10)).apply { add(accumulator) }\n while (iterator.hasNext()) {\n accumulator
= operation(accumulator, iterator.next())\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original collection and current accumulator value that starts with the first element of this

collection.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <S, T : S> Iterable<T>.runningReduceIndexed(operation: (index: Int, acc: S, T) -> S): List<S> {\n val iterator = this.iterator()\n if (!iterator.hasNext()) return emptyList()\n var accumulator: S = iterator.next()\n val result = ArrayList<S>(collectionSizeOrDefault(10)).apply { add(accumulator) }\n var index = 1\n while (iterator.hasNext()) {\n accumulator = operation(index++, accumulator, iterator.next())\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <T, R> Iterable<T>.scan(initial: R, operation: (acc: R, T) -> R): List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original collection and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <T, R> Iterable<T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): List<R> {\n return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the collection.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun <T> Iterable<T>.sumBy(selector: (T) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the collection.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun <T> Iterable<T>.sumByDouble(selector: (T) -> Double): Double {\n var sum: Double = 0.0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the collection.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfDouble\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.sumOf(selector: (T) -> Double): Double {\n var sum: Double = 0.toDouble()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the collection.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfInt\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.sumOf(selector: (T) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the collection.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfLong\")\n@kotlin.internal.InlineOnly\npublic

inline fun <T> Iterable<T>.sumOf(selector: (T) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the collection.\n */\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.sumOf(selector: (T) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the collection.\n */\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.sumOf(selector: (T) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns an original collection containing all the non-`null` elements, throwing an [IllegalArgumentException] if there are any `null` elements.\n */\npublic fun <T : Any> Iterable<T?>.requireNoNulls(): Iterable<T> {\n for (element in this) {\n if (element == null) {\n throw IllegalArgumentException(\"null element found in $this.\")\n }\n }\n @Suppress(\"UNCHECKED_CAST\")\n return this as Iterable<T>\n}\n\n/**\n * Returns an original collection containing all the non-`null` elements, throwing an [IllegalArgumentException] if there are any `null` elements.\n */\npublic fun <T : Any> List<T?>.requireNoNulls(): List<T> {\n for (element in this) {\n if (element == null) {\n throw IllegalArgumentException(\"null element found in $this.\")\n }\n }\n @Suppress(\"UNCHECKED_CAST\")\n return this as List<T>\n}\n\n/**\n * Splits this collection into a list of lists each not exceeding the given [size].\n * \n * The last list in the resulting list may have fewer elements than the given [size].\n * \n * @param size the number of elements to take in each list, must be positive and can be greater than the number of elements in this collection.\n * \n * @sample samples.collections.Collections.Transformations.chunked\n */\n@SinceKotlin(\"1.2\")\npublic fun <T> Iterable<T>.chunked(size: Int): List<List<T>> {\n return windowed(size, size, partialWindows = true)\n}\n\n/**\n * Splits this collection into several lists each not exceeding the given [size]\n * and applies the given [transform] function to an each.\n * \n * @return list of results of the [transform] applied to an each list.\n * \n * Note that the list passed to the [transform] function is ephemeral and is valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a snapshot of it.\n * The last list may have fewer elements than the given [size].\n * \n * @param size the number of elements to take in each list, must be positive and can be greater than the number of elements in this collection.\n * \n * @sample samples.text.Strings.chunkedTransform\n */\n@SinceKotlin(\"1.2\")\npublic fun <T, R> Iterable<T>.chunked(size: Int, transform: (List<T>) -> R): List<R> {\n return windowed(size, size, partialWindows = true, transform = transform)\n}\n\n/**\n * Returns a list containing all elements of the original collection without the first occurrence of the given [element].\n */\npublic operator fun <T> Iterable<T>.minus(element: T): List<T> {\n val result = ArrayList<T>(collectionSizeOrDefault(10))\n var removed = false\n return this.filterTo(result) { if (!removed && it == element) { removed = true; false } else true }\n}\n\n/**\n * Returns a list containing all elements of the original collection except the elements contained in the given [elements] array.\n */\npublic operator fun <T> Iterable<T>.minus(elements: Array<out T>): List<T> {\n if (elements.isEmpty()) return this.toList()\n return this.filterNot { it in elements }\n}\n\n/**\n * Returns a list containing all elements of the original collection except the elements contained in the given [elements] collection.\n */\npublic operator fun <T> Iterable<T>.minus(elements: Iterable<T>): List<T> {\n val other = elements.convertToListIfNotCollection()\n if (other.isEmpty())\n return this.toList()\n return this.filterNot { it in other }\n}\n\n/**\n * Returns a list containing all elements of the original collection except

the elements contained in the given [elements] sequence.\n */\npublic operator fun <T> Iterable<T>.minus(elements: Sequence<T>): List<T> {\n val other = elements.toList()\n if (other.isEmpty())\n return this.toList()\n return this.filterNot { it in other }\n}\n\n/**\n * Returns a list containing all elements of the original collection without the first occurrence of the given [element].\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.minusElement(element: T): List<T> {\n return minus(element)\n}\n\n/**\n * Splits the original collection into pair of lists,\n * where *first* list contains elements for which [predicate] yielded `true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n * \n * @sample samples.collections.Iterables.Operations.partition\n */\npublic inline fun <T> Iterable<T>.partition(predicate: (T) -> Boolean): Pair<List<T>, List<T>> {\n val first = ArrayList<T>()\n val second = ArrayList<T>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Returns a list containing all elements of the original collection and then the given [element].\n */\npublic operator fun <T> Iterable<T>.plus(element: T): List<T> {\n if (this is Collection) return this.plus(element)\n val result = ArrayList<T>()\n result.addAll(this)\n result.add(element)\n return result\n}\n\n/**\n * Returns a list containing all elements of the original collection and then the given [element].\n */\npublic operator fun <T> Collection<T>.plus(element: T): List<T> {\n val result = ArrayList<T>(size + 1)\n result.addAll(this)\n result.add(element)\n return result\n}\n\n/**\n * Returns a list containing all elements of the original collection and then all elements of the given [elements] array.\n */\npublic operator fun <T> Iterable<T>.plus(elements: Array<out T>): List<T> {\n if (this is Collection) return this.plus(elements)\n val result = ArrayList<T>()\n result.addAll(this)\n result.addAll(elements)\n return result\n}\n\n/**\n * Returns a list containing all elements of the original collection and then all elements of the given [elements] array.\n */\npublic operator fun <T> Collection<T>.plus(elements: Array<out T>): List<T> {\n val result = ArrayList<T>(this.size + elements.size)\n result.addAll(this)\n result.addAll(elements)\n return result\n}\n\n/**\n * Returns a list containing all elements of the original collection and then all elements of the given [elements] collection.\n */\npublic operator fun <T> Iterable<T>.plus(elements: Iterable<T>): List<T> {\n if (this is Collection) return this.plus(elements)\n val result = ArrayList<T>()\n result.addAll(this)\n result.addAll(elements)\n return result\n}\n\n/**\n * Returns a list containing all elements of the original collection and then all elements of the given [elements] collection.\n */\npublic operator fun <T> Collection<T>.plus(elements: Iterable<T>): List<T> {\n if (elements is Collection) {\n val result = ArrayList<T>(this.size + elements.size)\n result.addAll(this)\n result.addAll(elements)\n return result\n } else {\n val result = ArrayList<T>(this)\n result.addAll(elements)\n return result\n }\n}\n\n/**\n * Returns a list containing all elements of the original collection and then all elements of the given [elements] sequence.\n */\npublic operator fun <T> Iterable<T>.plus(elements: Sequence<T>): List<T> {\n val result = ArrayList<T>()\n result.addAll(this)\n result.addAll(elements)\n return result\n}\n\n/**\n * Returns a list containing all elements of the original collection and then all elements of the given [elements] sequence.\n */\npublic operator fun <T> Collection<T>.plus(elements: Sequence<T>): List<T> {\n val result = ArrayList<T>(this.size + 10)\n result.addAll(this)\n result.addAll(elements)\n return result\n}\n\n/**\n * Returns a list containing all elements of the original collection and then the given [element].\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.plusElement(element: T): List<T> {\n return plus(element)\n}\n\n/**\n * Returns a list containing all elements of the original collection and then the given [element].\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Collection<T>.plusElement(element: T): List<T> {\n return plus(element)\n}\n\n/**\n * Returns a list of snapshots of the window of the given [size]\n * sliding along this collection with the given [step], where each\n * snapshot is a list.\n * \n * Several last lists may have fewer elements than the given [size].\n * \n * Both [size] and [step] must be positive and can be greater than the number of elements in this collection.\n * @param size the number of elements to take in each window\n * @param step the number of elements to move the window forward by on an each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the end if any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample

samples.collections.Sequences.Transformations.takeWindows\n */\n@SinceKotlin(\"1.2\")\npublic fun <T> Iterable<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false): List<List<T>> {\n checkWindowSizeStep(size, step)\n if (this is RandomAccess && this is List) {\n val thisSize = this.size\n val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1\n val result = ArrayList<List<T>>(resultCapacity)\n var index = 0\n while (index in 0 until thisSize) {\n val windowSize = size.coerceAtMost(thisSize - index)\n if (windowSize < size && !partialWindows) break\n result.add(List(windowSize) { this[it + index] })\n index += step\n }\n return result\n }\n val result = ArrayList<List<T>>()\n windowedIterator(iterator(), size, step, partialWindows, reuseBuffer = false).forEach {\n result.add(it)\n }\n return result\n}\n\n/**\n * Returns a list of results of applying the given [transform] function to\n * an each list representing a view over the window of the given [size]\n * sliding along this collection with the given [step].\n * \n * Note that the list passed to the [transform] function is ephemeral and is valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a snapshot of it.\n * Several last lists may have fewer elements than the given [size].\n * \n * Both [size] and [step] must be positive and can be greater than the number of elements in this collection.\n * @param size the number of elements to take in each window\n * @param step the number of elements to move the window forward by on an each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the end if any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample samples.collections.Sequences.Transformations.averageWindows\n */\n@SinceKotlin(\"1.2\")\npublic fun <T, R> Iterable<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) -> R): List<R> {\n checkWindowSizeStep(size, step)\n if (this is RandomAccess && this is List) {\n val thisSize = this.size\n val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1\n val result = ArrayList<R>(resultCapacity)\n val window = MovingSubList(this)\n var index = 0\n while (index in 0 until thisSize) {\n val windowSize = size.coerceAtMost(thisSize - index)\n if (!partialWindows && windowSize < size) break\n window.move(index, index + windowSize)\n result.add(transform(window))\n index += step\n }\n return result\n }\n val result = ArrayList<R>()\n windowedIterator(iterator(), size, step, partialWindows, reuseBuffer = true).forEach {\n result.add(transform(it))\n }\n return result\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <T, R> Iterable<T>.zip(other: Array<out R>): List<Pair<T, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values built from the elements of `this` collection and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <T, R, V> Iterable<T>.zip(other: Array<out R>, transform: (a: T, b: R) -> V): List<V> {\n val arraySize = other.size\n val list = ArrayList<V>(minOf(collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in this) {\n if (i >= arraySize) break\n list.add(transform(element, other[i++]))\n }\n return list\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] collection with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\npublic infix fun <T, R> Iterable<T>.zip(other: Iterable<R>): List<Pair<T, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values built from the elements of `this` collection and the [other] collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\npublic inline fun <T, R, V> Iterable<T>.zip(other: Iterable<R>, transform: (a: T, b: R) -> V): List<V> {\n val first = iterator()\n val second = other.iterator()\n val list = ArrayList<V>(minOf(collectionSizeOrDefault(10), other.collectionSizeOrDefault(10)))\n while (first.hasNext() && second.hasNext()) {\n

list.add(transform(first.next(), second.next()))\n   }\n   return list\n}\n\n/**\n * Returns a list of pairs of each two adjacent elements in this collection.\n * \n * The returned list is empty if this collection contains less than two elements.\n * \n * @sample samples.collections.Collections.Transformations.zipWithNext\n */\n@SinceKotlin(\"1.2\")\npublic fun <T> Iterable<T>.zipWithNext(): List<Pair<T, T>> {\n   return zipWithNext { a, b -> a to b }\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to an each pair of two adjacent elements in this collection.\n * \n * The returned list is empty if this collection contains less than two elements.\n * \n * @sample samples.collections.Collections.Transformations.zipWithNextToFindDeltas\n */\n@SinceKotlin(\"1.2\")\npublic inline fun <T, R> Iterable<T>.zipWithNext(transform: (a: T, b: T) -> R): List<R> {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return emptyList()\n   val result = mutableListOf<R>()\n   var current = iterator.next()\n   while (iterator.hasNext()) {\n   val next = iterator.next()\n   result.add(transform(current, next))\n   current = next\n   }\n   return result\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n */\npublic fun <T, A : Appendable> Iterable<T>.joinTo(buffer: A, separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((T) -> CharSequence)? = null): A {\n   buffer.append(prefix)\n   var count = 0\n   for (element in this) {\n   if (++count > 1) buffer.append(separator)\n   if (limit < 0 || count <= limit) {\n   buffer.appendElement(element, transform)\n   } else break\n   }\n   if (limit >= 0 && count > limit) buffer.append(truncated)\n   buffer.append(postfix)\n   return buffer\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n * \n * @sample samples.collections.Collections.Transformations.joinToString\n */\npublic fun <T> Iterable<T>.joinToString(separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((T) -> CharSequence)? = null): String {\n   return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Returns this collection as an [Iterable].\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Iterable<T>.asIterable(): Iterable<T> {\n   return this\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original collection returning its elements when being iterated.\n * \n * @sample samples.collections.Sequences.Building.sequenceFromCollection\n */\npublic fun <T> Iterable<T>.asSequence(): Sequence<T> {\n   return Sequence { this.iterator() }\n}\n\n/**\n * Returns an average value of elements in the collection.\n */\n@kotlin.jvm.JvmName(\"averageOfByte\")\npublic fun Iterable<Byte>.average(): Double {\n   var sum: Double = 0.0\n   var count: Int = 0\n   for (element in this) {\n   sum += element\n   checkCountOverflow(++count)\n   }\n   return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the collection.\n */\n@kotlin.jvm.JvmName(\"averageOfShort\")\npublic fun Iterable<Short>.average(): Double {\n   var sum: Double = 0.0\n   var count: Int = 0\n   for (element in this) {\n   sum += element\n   checkCountOverflow(++count)\n   }\n   return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the collection.\n */\n@kotlin.jvm.JvmName(\"averageOfInt\")\npublic fun Iterable<Int>.average(): Double {\n   var sum: Double = 0.0\n   var count: Int = 0\n   for (element in this) {\n   sum += element\n   checkCountOverflow(++count)\n   }\n   return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the collection.\n */\n@kotlin.jvm.JvmName(\"averageOfLong\")\npublic fun Iterable<Long>.average(): Double {\n   var sum: Double = 0.0\n   var count: Int = 0\n   for (element in this) {\n   sum += element\n   checkCountOverflow(++count)\n   }\n   return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns

```
an average value of elements in the collection.\n */\n@kotlin.jvm.JvmName(\"averageOfFloat\")\npublic fun Iterable<Float>.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the collection.\n */\n@kotlin.jvm.JvmName(\"averageOfDouble\")\npublic fun Iterable<Double>.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName(\"sumOfByte\")\npublic fun Iterable<Byte>.sum(): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName(\"sumOfShort\")\npublic fun Iterable<Short>.sum(): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName(\"sumOfInt\")\npublic fun Iterable<Int>.sum(): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName(\"sumOfLong\")\npublic fun Iterable<Long>.sum(): Long {\n    var sum: Long = 0L\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName(\"sumOfFloat\")\npublic fun Iterable<Float>.sum(): Float {\n    var sum: Float = 0.0f\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName(\"sumOfDouble\")\npublic fun Iterable<Double>.sum(): Double {\n    var sum: Double = 0.0\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\nimport kotlin.comparisons.naturalOrder\nimport kotlin.random.Random\nimport kotlin.js.arrayBufferIsView\n\n/**\n * Returns the array if it's not `null`, or an empty array otherwise.\n * @sample samples.collections.Arrays.Usage.arrayOrEmpty\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun <T> Array<out T>?.orEmpty(): Array<out T> = this ?: emptyArray<T>()\n\n/**\n * Returns a *typed* array containing all of the elements of this collection.\n *\n * Allocates an array of runtime type `T` having its size equal to the size of this collection\n * and populates the array with the elements of this collection.\n * @sample samples.collections.Collections.Collections.collectionToTypedArray\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun <T> Collection<T>.toTypedArray(): Array<T> = copyToArray(this)\n\n@JsName(\"copyToArray\")\n@PublishedApi\ninternal fun <T> copyToArray(collection: Collection<T>): Array<T> {\n    return if (collection.asDynamic().toArray !== undefined)\n        collection.asDynamic().toArray().unsafeCast<Array<T>>()\n    else\n        copyToArrayImpl(collection).unsafeCast<Array<T>>()\n}\n\n@JsName(\"copyToArrayImpl\")\ninternal actual fun copyToArrayImpl(collection: Collection<*>): Array<Any?> {\n    val array = emptyArray<Any?>()\n    val iterator = collection.iterator()\n    while (iterator.hasNext())\n        array.asDynamic().push(iterator.next())\n    return array\n}\n\n@JsName(\"copyToExistingArrayImpl\")\ninternal actual fun <T> copyToArrayImpl(collection: Collection<*>, array: Array<T>): Array<T> {\n    if (array.size < collection.size)\n        return copyToArrayImpl(collection).unsafeCast<Array<T>>()\n\n    val iterator = collection.iterator()\n    var index = 0\n    while (iterator.hasNext()) {\n        array[index++] = iterator.next().unsafeCast<T>()\n    }\n    if (index < array.size) {\n        array[index] = null.unsafeCast<T>()\n    }\n    return array\n}\n\n\n/**\n * Returns an immutable list containing only the specified object [element].\n */\npublic
 fun <T> listOf(element: T): List<T> = arrayListOf(element)\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\ninternal actual inline fun <E> buildListInternal(builderAction: MutableList<E>.() -> Unit): List<E> {\n    return
```

```
ArrayList<E>().apply(builderAction).build()\n}\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\ninternal actual inline fun <E> buildListInternal(capacity: Int, builderAction: MutableList<E>.() -> Unit): List<E> {\n    checkBuilderCapacity(capacity)\n    return ArrayList<E>(capacity).apply(builderAction).build()\n}\n\n\n/**\n * Returns an immutable set containing only the specified object [element].\n */\npublic fun <T> setOf(element: T): Set<T> = hashSetOf(element)\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\ninternal actual inline fun <E> buildSetInternal(builderAction: MutableSet<E>.() -> Unit): Set<E> {\n    return LinkedHashSet<E>().apply(builderAction).build()\n}\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\ninternal
 actual inline fun <E> buildSetInternal(capacity: Int, builderAction: MutableSet<E>.() -> Unit): Set<E> {\n    return LinkedHashSet<E>(capacity).apply(builderAction).build()\n}\n\n\n/**\n * Returns an immutable map, mapping only the specified key to the\n * specified value.\n */\npublic fun <K, V> mapOf(pair: Pair<K, V>): Map<K, V> = hashMapOf(pair)\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\ninternal actual inline fun <K, V> buildMapInternal(builderAction: MutableMap<K, V>.() -> Unit): Map<K, V> {\n    return LinkedHashMap<K,
V>().apply(builderAction).build()\n}\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\ninternal actual inline fun <K, V> buildMapInternal(capacity: Int, builderAction: MutableMap<K, V>.() -> Unit): Map<K, V> {\n    return LinkedHashMap<K, V>(capacity).apply(builderAction).build()\n}\n\n\n/**\n * Fills the list with the provided [value].\n *\n * Each element in the list gets
 replaced with the [value].\n */\n@SinceKotlin(\"1.2\")\npublic actual fun <T> MutableList<T>.fill(value: T): Unit {\n    for (index in 0..lastIndex) {\n        this[index] = value\n    }\n}\n\n/**\n * Randomly shuffles elements in this list.\n *\n * See: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm\n */\n@SinceKotlin(\"1.2\")\npublic actual fun <T> MutableList<T>.shuffle(): Unit = shuffle(Random)\n\n/**\n * Returns a new list with the elements of this list randomly shuffled.\n */\n@SinceKotlin(\"1.2\")\npublic actual fun <T> Iterable<T>.shuffled(): List<T> = toMutableList().apply { shuffle() }\n\n/**\n * Sorts elements in the list in-place according to their natural sort order.\n *\n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n *\n * @sample samples.collections.Collections.Sorting.sortMutableList\n */\npublic actual fun <T : Comparable<T>> MutableList<T>.sort(): Unit {\n    collectionsSort(this,
 naturalOrder())\n}\n\n/**\n * Sorts elements in the list in-place according to the order specified with [comparator].\n *\n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n *\n * @sample samples.collections.Collections.Sorting.sortMutableListWith\n */\npublic actual fun <T> MutableList<T>.sortWith(comparator: Comparator<in T>): Unit {\n    collectionsSort(this, comparator)\n}\n\nprivate fun <T> collectionsSort(list: MutableList<T>, comparator: Comparator<in T>) {\n    if (list.size <= 1) return\n\n    val array = copyToArray(list)\n    sortArrayWith(array, comparator)\n\n    for (i in 0 until array.size) {\n        list[i] = array[i]\n    }\n}\n\ninternal actual fun <T> arrayOfNulls(reference: Array<T>, size: Int): Array<T> {\n    return arrayOfNulls<Any>(size).unsafeCast<Array<T>>()\n}\n\n@SinceKotlin(\"1.3\")\n@PublishedApi\n@JsName(\"arrayCopy\")\ninternal fun <T> arrayCopy(source: Array<out T>, destination:
 Array<in T>, destinationOffset: Int, startIndex: Int, endIndex: Int) {\n    AbstractList.checkRangeIndexes(startIndex, endIndex, source.size)\n    val rangeSize = endIndex - startIndex\n    AbstractList.checkRangeIndexes(destinationOffset, destinationOffset + rangeSize, destination.size)\n\n    if (arrayBufferIsView(destination) && arrayBufferIsView(source)) {\n        val subrange = source.asDynamic().subarray(startIndex, endIndex)\n        destination.asDynamic().set(subrange, destinationOffset)\n    } else {\n        if (source !== destination || destinationOffset <= startIndex) {\n            for (index in 0 until rangeSize) {\n                destination[destinationOffset + index] = source[startIndex + index]\n            }\n        } else {\n            for (index in rangeSize - 1 downTo 0) {\n                destination[destinationOffset + index] = source[startIndex + index]\n            }\n        }\n    }\n}\n\n// no singleton map implementation in js, return map as is\n@Suppress(\"NOTHING_TO_INLINE\")\ninternal
```

actual inline fun <K, V> Map<K, V>.toSingletonMapOrSelf(): Map<K, V> = this\n\n@Suppress(\"NOTHING_TO_INLINE\")\ninternal actual inline fun <K, V> Map<out K, V>.toSingletonMap(): Map<K, V> = this.toMutableMap()\n\n\n@Suppress(\"NOTHING_TO_INLINE\")\ninternal actual inline fun <T> Array<out T>.copyToArrayOfAny(isVarargs: Boolean): Array<out Any?> =\n   if (isVarargs)\n   // no need to copy vararg array in JS\n   this\n   else\n this.copyOf()\n\n\n\n@PublishedApi\ninternal actual fun checkIndexOverflow(index: Int): Int {\n   if (index < 0) {\n   throwIndexOverflow()\n   }\n   return index\n}\n\n@PublishedApi\ninternal actual fun checkCountOverflow(count: Int): Int {\n   if (count < 0) {\n   throwCountOverflow()\n   }\n   return count\n}\n\n\n/**\n * JS map and set implementations do not make use of capacities or load factors.\n */\n@PublishedApi\ninternal actual fun mapCapacity(expectedSize: Int) = expectedSize\n\n/**\n * Checks a collection builder function capacity argument.\n * In JS no validation is made in Map/Set constructor yet.\n */\n@SinceKotlin(\"1.3\")\n@PublishedApi\ninternal fun checkBuilderCapacity(capacity: Int) {\n require(capacity >= 0) { \"capacity must be non-negative.\" }\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"CollectionsKt\")\n\npackage kotlin.collections\n\n\n/**\n * Returns the given iterator itself. This allows to use an instance of iterator in a `for` loop.\n * @sample samples.collections.Iterators.iterator\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Iterator<T>.iterator(): Iterator<T> = this\n\n/**\n * Returns an [Iterator] that wraps each element produced by the original iterator\n * into an [IndexedValue]
containing the index of that element and the element itself.\n *\n * @sample samples.collections.Iterators.withIndexIterator\n */\npublic fun <T> Iterator<T>.withIndex(): Iterator<IndexedValue<T>> = IndexingIterator(this)\n\n/**\n * Performs the given [operation] on each element of this [Iterator].\n * @sample samples.collections.Iterators.forEachIterator\n */\npublic inline fun <T> Iterator<T>.forEach(operation: (T) -> Unit): Unit {\n   for (element in this) operation(element)\n}\n\n/**\n * Iterator transforming original `iterator` into iterator of [IndexedValue], counting index from zero.\n */\ninternal class IndexingIterator<out T>(private val iterator: Iterator<T>) : Iterator<IndexedValue<T>> {\n   private var index = 0\n   final override fun hasNext(): Boolean = iterator.hasNext()\n   final override fun next(): IndexedValue<T> = IndexedValue(checkIndexOverflow(index++), iterator.next())\n}\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n
* Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"ComparisonsKt\")\n\npackage kotlin.comparisons\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.random.*\n\n/**\n * Returns the greater of two values.\n * \n * If values are equal, returns the first one.\n */\n@SinceKotlin(\"1.1\")\npublic expect fun <T : Comparable<T>> maxOf(a: T, b: T): T\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Byte, b: Byte): Byte\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Short, b: Short): Short\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic
expect inline fun maxOf(a: Int, b: Int): Int\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Long, b: Long): Long\n\n/**\n * Returns the greater of two values.\n * \n * If either value is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Float, b: Float): Float\n\n/**\n * Returns the greater of two values.\n * \n * If either value is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Double, b: Double): Double\n\n/**\n * Returns the greater of three values.\n * \n * If there are multiple equal maximal values, returns the first of them.\n */\n@SinceKotlin(\"1.1\")\npublic expect fun <T : Comparable<T>> maxOf(a: T, b: T, c: T):

```
T\n\n/**\n * Returns the greater of three values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic
 expect inline fun maxOf(a: Byte, b: Byte, c: Byte): Byte\n\n/**\n * Returns the greater of three values.\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Short, b: Short, c:
Short): Short\n\n/**\n * Returns the greater of three values.\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a: Int, b: Int, c: Int):
Int\n\n/**\n * Returns the greater of three values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic
expect inline fun maxOf(a: Long, b: Long, c: Long): Long\n\n/**\n * Returns the greater of three values.\n * \n * If
any value is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline
fun maxOf(a: Float, b: Float, c: Float): Float\n\n/**\n * Returns the greater of three values.\n * \n * If any value is
`NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun maxOf(a:
Double, b: Double, c:
 Double): Double\n\n/**\n * Returns the greater of three values according to the order specified by the given
[comparator].\n * \n * If there are multiple equal maximal values, returns the first of them.\n
*/\n@SinceKotlin(\"1.1\")\npublic fun <T> maxOf(a: T, b: T, c: T, comparator: Comparator<in T>): T {\n    return
maxOf(a, maxOf(b, c, comparator), comparator)\n}\n\n/**\n * Returns the greater of two values according to the
order specified by the given [comparator].\n * \n * If values are equal, returns the first one.\n
*/\n@SinceKotlin(\"1.1\")\npublic fun <T> maxOf(a: T, b: T, comparator: Comparator<in T>): T {\n    return if
(comparator.compare(a, b) >= 0) a else b\n}\n\n/**\n * Returns the greater of the given values.\n * \n * If there are
multiple equal maximal values, returns the first of them.\n */\n@SinceKotlin(\"1.4\")\npublic expect fun <T :
Comparable<T>> maxOf(a: T, vararg other: T): T\n\n/**\n * Returns the greater of the given values.\n
*/\n@SinceKotlin(\"1.4\")\npublic
 expect fun maxOf(a: Byte, vararg other: Byte): Byte\n\n/**\n * Returns the greater of the given values.\n
*/\n@SinceKotlin(\"1.4\")\npublic expect fun maxOf(a: Short, vararg other: Short): Short\n\n/**\n * Returns the
greater of the given values.\n */\n@SinceKotlin(\"1.4\")\npublic expect fun maxOf(a: Int, vararg other: Int):
Int\n\n/**\n * Returns the greater of the given values.\n */\n@SinceKotlin(\"1.4\")\npublic expect fun maxOf(a:
Long, vararg other: Long): Long\n\n/**\n * Returns the greater of the given values.\n * \n * If any value is `NaN`,
returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic expect fun maxOf(a: Float, vararg other: Float): Float\n\n/**\n *
Returns the greater of the given values.\n * \n * If any value is `NaN`, returns `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\npublic expect fun maxOf(a: Double, vararg other: Double): Double\n\n/**\n * Returns
the greater of the given values according to the order specified by the given [comparator].\n * \n * If there are
multiple equal
 maximal values, returns the first of them.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T> maxOf(a: T, vararg other:
T, comparator: Comparator<in T>): T {\n    var max = a\n    for (e in other) if (comparator.compare(max, e) < 0)
max = e\n    return max\n}\n\n/**\n * Returns the smaller of two values.\n * \n * If values are equal, returns the first
one.\n */\n@SinceKotlin(\"1.1\")\npublic expect fun <T : Comparable<T>> minOf(a: T, b: T): T\n\n/**\n * Returns
the smaller of two values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun
minOf(a: Byte, b: Byte): Byte\n\n/**\n * Returns the smaller of two values.\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun minOf(a: Short, b: Short):
Short\n\n/**\n * Returns the smaller of two values.\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun minOf(a: Int, b: Int): Int\n\n/**\n
* Returns the smaller of two values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic
 expect inline fun minOf(a: Long, b: Long): Long\n\n/**\n * Returns the smaller of two values.\n * \n * If either
value is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun
minOf(a: Float, b: Float): Float\n\n/**\n * Returns the smaller of two values.\n * \n * If either value is `NaN`,
returns `NaN`.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun minOf(a:
Double, b: Double): Double\n\n/**\n * Returns the smaller of three values.\n * \n * If there are multiple equal
minimal values, returns the first of them.\n */\n@SinceKotlin(\"1.1\")\npublic expect fun <T : Comparable<T>>
minOf(a: T, b: T, c: T): T\n\n/**\n * Returns the smaller of three values.\n
```

*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun minOf(a: Byte, b: Byte, c: Byte): Byte\n\n/**\n * Returns the smaller of three values.\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic
expect inline fun minOf(a: Short, b: Short, c: Short): Short\n\n/**\n * Returns the smaller of three values.\n
*/\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun minOf(a: Int, b: Int, c: Int): Int\n\n/**\n * Returns the smaller of three values.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun minOf(a: Long, b: Long, c: Long): Long\n\n/**\n * Returns the smaller of three values.\n * \n * If any value is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun minOf(a: Float, b: Float, c: Float): Float\n\n/**\n * Returns the smaller of three values.\n * \n * If any value is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic expect inline fun minOf(a: Double, b: Double, c: Double): Double\n\n/**\n * Returns the smaller of three values according to the order specified by the given [comparator].\n * \n * If there are multiple equal
minimal values, returns the first of them.\n */\n@SinceKotlin(\"1.1\")\npublic fun <T> minOf(a: T, b: T, c: T, comparator: Comparator<in T>): T {\n   return minOf(a, minOf(b, c, comparator), comparator)\n}\n\n/**\n * Returns the smaller of two values according to the order specified by the given [comparator].\n * \n * If values are equal, returns the first one.\n */\n@SinceKotlin(\"1.1\")\npublic fun <T> minOf(a: T, b: T, comparator: Comparator<in T>): T {\n   return if (comparator.compare(a, b) <= 0) a else b\n}\n\n/**\n * Returns the smaller of the given values.\n * \n * If there are multiple equal minimal values, returns the first of them.\n */\n@SinceKotlin(\"1.4\")\npublic expect fun <T : Comparable<T>> minOf(a: T, vararg other: T): T\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin(\"1.4\")\npublic expect fun minOf(a: Byte, vararg other: Byte): Byte\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin(\"1.4\")\npublic expect fun minOf(a:
Short, vararg other: Short): Short\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin(\"1.4\")\npublic expect fun minOf(a: Int, vararg other: Int): Int\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin(\"1.4\")\npublic expect fun minOf(a: Long, vararg other: Long): Long\n\n/**\n * Returns the smaller of the given values.\n * \n * If any value is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic expect fun minOf(a: Float, vararg other: Float): Float\n\n/**\n * Returns the smaller of the given values.\n * \n * If any value is `NaN`, returns `NaN`.\n */\n@SinceKotlin(\"1.4\")\npublic expect fun minOf(a: Double, vararg other: Double): Double\n\n/**\n * Returns the smaller of the given values according to the order specified by the given [comparator].\n * \n * If there are multiple equal minimal values, returns the first of them.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T> minOf(a: T, vararg other: T, comparator: Comparator<in T>): T {\n   var
min = a\n   for (e in other) if (comparator.compare(min, e) > 0) min = e\n   return min\n}\n\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"MapsKt\")\n\npackage kotlin.collections\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.random.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to entries of this map in iteration order,\n * or throws [NoSuchElementException] if no non-null value was produced.\n * \n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic
inline fun <K, V, R : Any> Map<out K, V>.firstNotNullOf(transform: (Map.Entry<K, V>) -> R?): R {\n   return firstNotNullOfOrNull(transform) ?: throw NoSuchElementException(\"No element of the map was transformed to a non-null value.\")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to entries of this map in iteration order,\n * or `null` if no non-null value was produced.\n * \n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n

*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Any> Map<out K, V>.firstNotNullOfOrNull(transform: (Map.Entry<K, V>) -> R?): R? {\n for (element in this) {\n val result = transform(element)\n if (result != null) {\n return result\n }\n }\n return null\n}\n\n/**\n * Returns a [List] containing all key-value pairs.\n */\npublic fun <K, V> Map<out K, V>.toList(): List<Pair<K, V>> {\n if (size == 0)\n return emptyList()\n val iterator = entries.iterator()\n if (!iterator.hasNext())\n return emptyList()\n val first = iterator.next()\n if (!iterator.hasNext())\n return listOf(first.toPair())\n val result = ArrayList<Pair<K, V>>(size)\n result.add(first.toPair())\n do {\n result.add(iterator.next().toPair())\n } while (iterator.hasNext())\n return result\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each entry of original map.\n * \n * @sample samples.collections.Maps.Transformations.flatMap\n */\npublic inline fun <K, V, R> Map<out K, V>.flatMap(transform: (Map.Entry<K, V>) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each entry of original map.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapSequence\")\npublic inline fun <K, V, R> Map<out K, V>.flatMap(transform: (Map.Entry<K, V>) -> Sequence<R>): List<R> {\n return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each entry of original map, to the given [destination].\n */\npublic inline fun <K, V, R, C : MutableCollection<in R>> Map<out K, V>.flatMapTo(destination: C, transform: (Map.Entry<K, V>) -> Iterable<R>): C {\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each entry of original map, to the given [destination].\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapSequenceTo\")\npublic inline fun <K, V, R, C : MutableCollection<in R>> Map<out K, V>.flatMapTo(destination: C, transform: (Map.Entry<K, V>) -> Sequence<R>): C {\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each entry in the original map.\n * \n * @sample samples.collections.Maps.Transformations.mapToList\n */\npublic inline fun <K, V, R> Map<out K, V>.map(transform: (Map.Entry<K, V>) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing only the non-null results of applying the given [transform] function\n * to each entry in the original map.\n * \n * @sample samples.collections.Maps.Transformations.mapNotNull\n */\npublic inline fun <K, V, R : Any> Map<out K, V>.mapNotNull(transform:\n (Map.Entry<K, V>) -> R?): List<R> {\n return mapNotNullTo(ArrayList<R>(), transform)\n}\n\n/**\n * Applies the given [transform] function to each entry in the original map\n * and appends only the non-null results to the given [destination].\n */\npublic inline fun <K, V, R : Any, C : MutableCollection<in R>> Map<out K, V>.mapNotNullTo(destination: C, transform: (Map.Entry<K, V>) -> R?): C {\n forEach { element -> transform(element)?.let { destination.add(it) } }\n return destination\n}\n\n/**\n * Applies the given [transform] function to each entry of the original map\n * and appends the results to the given [destination].\n */\npublic inline fun <K, V, R, C : MutableCollection<in R>> Map<out K, V>.mapTo(destination: C, transform: (Map.Entry<K, V>) -> R): C {\n for (item in this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Returns `true` if all entries match the given [predicate].\n * \n * Note that if the map contains no entries,\n the function returns `true`\n * because there are no entries in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in [\"Vacuous truth\"](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun <K, V> Map<out K, V>.all(predicate: (Map.Entry<K, V>) -> Boolean): Boolean {\n if (isEmpty()) return true\n for (element in this)

if (!predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if map has at least one entry.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\npublic fun <K, V> Map<out K, V>.any(): Boolean {\n   return !isEmpty()\n}\n\n/**\n * Returns `true` if at least one entry matches the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun <K, V> Map<out K, V>.any(predicate: (Map.Entry<K, V>) -> Boolean): Boolean {\n   if (isEmpty())
    return false\n   for (element in this) if (predicate(element)) return true\n   return false\n}\n\n/**\n * Returns the number of entries in this map.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.count(): Int {\n   return size\n}\n\n/**\n * Returns the number of entries matching the given [predicate].\n */\npublic inline fun <K, V> Map<out K, V>.count(predicate: (Map.Entry<K, V>) -> Boolean): Int {\n   if (isEmpty()) return 0\n   var count = 0\n   for (element in this) if (predicate(element)) ++count\n   return count\n}\n\n/**\n * Performs the given [action] on each entry.\n */\n@kotlin.internal.HidesMembers\npublic inline fun <K, V> Map<out K, V>.forEach(action: (Map.Entry<K, V>) -> Unit): Unit {\n   for (element in this) action(element)\n}\n\n/**\n * Returns the first entry yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the map is empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow\")\n@kotlin.internal.InlineOnly\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <K, V, R : Comparable<R>> Map<out K, V>.maxBy(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V> {\n   return entries.maxBy(selector)\n}\n\n/**\n * Returns the first entry yielding the largest value of the given function or `null` if there are no entries.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K, V>.maxByOrNull(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V>? {\n   return entries.maxByOrNull(selector)\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each entry in the map.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the map is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
    inline fun <K, V> Map<out K, V>.maxOf(selector: (Map.Entry<K, V>) -> Double): Double {\n   return entries.maxOf(selector)\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each entry in the map.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the map is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.maxOf(selector: (Map.Entry<K, V>) -> Float): Float {\n   return entries.maxOf(selector)\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each entry in the map.\n * \n * @throws
    NoSuchElementException if the map is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K, V>.maxOf(selector: (Map.Entry<K, V>) -> R): R {\n   return entries.maxOf(selector)\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each entry in the map or `null` if there are no entries.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.maxOfOrNull(selector: (Map.Entry<K, V>) -> Double): Double? {\n   return entries.maxOfOrNull(selector)\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied
    to each entry in the map or `null` if there are no entries.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.maxOfOrNull(selector: (Map.Entry<K, V>) -> Float): Float? {\n   return entries.maxOfOrNull(selector)\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each entry in the map or `null` if there are no entries.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K, V>.maxOfOrNull(selector: (Map.Entry<K, V>) -> R): R? {\n   return entries.maxOfOrNull(selector)\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each entry in the map.\n * \n * @throws NoSuchElementException if the map is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R> Map<out K, V>.maxOfWith(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R {\n   return entries.maxOfWith(comparator, selector)\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each entry in the map or `null` if there are no entries.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R> Map<out K, V>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R? {\n   return entries.maxOfWithOrNull(comparator, selector)\n}\n\n/**\n * Returns the first entry having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the map is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow\")\n@kotlin.internal.InlineOnly\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <K, V> Map<out K, V>.maxWith(comparator: Comparator<in Map.Entry<K, V>>): Map.Entry<K, V> {\n   return entries.maxWith(comparator)\n}\n\n/**\n * Returns the first entry having the largest value according to the provided [comparator] or `null` if there are no entries.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.maxWithOrNull(comparator: Comparator<in Map.Entry<K, V>>): Map.Entry<K, V>? {\n   return entries.maxWithOrNull(comparator)\n}\n\n/**\n * Returns the first entry yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the map is empty.\n * \n * @sample samples.collections.Collections.Aggregates.minBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow\")\n@kotlin.internal.InlineOnly\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <K, V, R : Comparable<R>> Map<out K, V>.minBy(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V> {\n   return entries.minBy(selector)\n}\n\n/**\n * Returns the first entry yielding the smallest value of the given function or `null` if there are no entries.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K, V>.minByOrNull(selector: (Map.Entry<K, V>) -> R): Map.Entry<K, V>? {\n   return entries.minByOrNull(selector)\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each entry in the map.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the map is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.minOf(selector: (Map.Entry<K, V>) -> Double): Double {\n   return entries.minOf(selector)\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each entry in the map.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the map is empty.\n

*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.minOf(selector: (Map.Entry<K, V>) -> Float): Float {\n   return entries.minOf(selector)\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each entry in the map.\n * \n * @throws NoSuchElementException if the map is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K, V>.minOf(selector: (Map.Entry<K, V>) -> R): R {\n   return entries.minOf(selector)\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each entry in the map or `null` if there are no entries.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.minOfOrNull(selector: (Map.Entry<K, V>) -> Double): Double? {\n   return entries.minOfOrNull(selector)\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each entry in the map or `null` if there are no entries.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.minOfOrNull(selector: (Map.Entry<K, V>) -> Float): Float? {\n   return entries.minOfOrNull(selector)\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each entry in the map or `null` if there are no entries.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R : Comparable<R>> Map<out K, V>.minOfOrNull(selector: (Map.Entry<K, V>) -> R): R? {\n   return entries.minOfOrNull(selector)\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each entry in the map.\n * \n * @throws NoSuchElementException if the map is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R> Map<out K, V>.minOfWith(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R {\n   return entries.minOfWith(comparator, selector)\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each entry in the map or `null` if there are no entries.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, R> Map<out K, V>.minOfWithOrNull(comparator: Comparator<in R>, selector: (Map.Entry<K, V>) -> R): R? {\n   return entries.minOfWithOrNull(comparator, selector)\n}\n\n/**\n * Returns the first entry having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the map is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow\")\n@kotlin.internal.InlineOnly\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <K, V> Map<out K, V>.minWith(comparator: Comparator<in Map.Entry<K, V>>): Map.Entry<K, V> {\n   return entries.minWith(comparator)\n}\n\n/**\n * Returns the first entry having the smallest value according to the provided [comparator] or `null` if there are no entries.\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.minWithOrNull(comparator: Comparator<in Map.Entry<K, V>>): Map.Entry<K, V>? {\n   return entries.minWithOrNull(comparator)\n}\n\n/**\n * Returns `true` if the map has no entries.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\npublic fun <K, V> Map<out K, V>.none(): Boolean {\n   return isEmpty()\n}\n\n/**\n * Returns `true` if no entries match the given [predicate].\n * \n * @sample

samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun <K, V> Map<out K, V>.none(predicate: (Map.Entry<K, V>) -> Boolean): Boolean {\n if (isEmpty()) return true\n for (element in this) if (predicate(element)) return false\n return true\n}\n\n/**\n * Performs the given [action] on each entry and returns the map itself afterwards.\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <K, V, M : Map<out K, V>> M.onEach(action: (Map.Entry<K, V>) -> Unit): M {\n return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each entry, providing sequential index with the entry,\n * and returns the map itself afterwards.\n * @param [action] function that takes the index of an entry and the entry itself\n * and performs
the action on the entry.\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <K, V, M : Map<out K, V>> M.onEachIndexed(action: (index: Int, Map.Entry<K, V>) -> Unit): M {\n return apply { entries.forEachIndexed(action) }\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original map returning its entries when being iterated.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> Map<out K, V>.asIterable(): Iterable<Map.Entry<K, V>> {\n return entries\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original map returning its entries when being iterated.\n */\npublic fun <K, V> Map<out K, V>.asSequence(): Sequence<Map.Entry<K, V>> {\n return entries.asSequence()\n}\n\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n//
See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\n// 10 mappings totally\ninternal fun Char.titlecaseImpl(): String {\n val uppercase = uppercase()\n if (uppercase.length > 1) {\n return if (this == '\\u0149') uppercase else uppercase[0] + uppercase.substring(1).lowercase()\n }\n return titlecaseChar().toString()\n}\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/**\n * Converts this character to lower case using Unicode mapping rules of the invariant locale.\n */\n@Deprecated(\"Use lowercaseChar() instead.\", ReplaceWith(\"lowercaseChar()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\n@kotlin.internal.InlineOnly\npublic actual inline fun Char.toLowerCase(): Char = lowercaseChar()\n\n/**\n * Converts this character to lower
case using Unicode mapping rules of the invariant locale.\n *\n * This function performs one-to-one character mapping.\n * To support one-to-many character mapping use the [lowercase] function.\n * If this character has no mapping equivalent, the character itself is returned.\n *\n * @sample samples.text.Chars.lowercase\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic actual inline fun Char.lowercaseChar(): Char = lowercase()[0]\n\n/**\n * Converts this character to lower case using Unicode mapping rules of the invariant locale.\n *\n * This function supports one-to-many character mapping, thus the length of the returned string can be greater than one.\n * For example, `'\\u0130'.lowercase()` returns `\"\\u0069\\u0307\"`,\n * where `'\\u0130'` is the LATIN CAPITAL LETTER I WITH DOT ABOVE character (`'\u0130'`).\n * If this character has no lower case mapping, the result of `toString()` of this char is returned.\n *\n * @sample samples.text.Chars.lowercase\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic actual inline fun Char.lowercase(): String = toString().asDynamic().toLowerCase().unsafeCast<String>()\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the invariant locale.\n */\n@Deprecated(\"Use uppercaseChar() instead.\", ReplaceWith(\"uppercaseChar()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\n@kotlin.internal.InlineOnly\npublic actual inline fun Char.toUpperCase(): Char = uppercaseChar()\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the invariant locale.\n *\n * This function performs one-to-one character mapping.\n * To support one-to-many character mapping use the [uppercase] function.\n * If this character has no mapping equivalent, the character itself is returned.\n *\n * @sample samples.text.Chars.uppercase\n

```
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
 actual fun Char.uppercaseChar(): Char {\n    val uppercase = uppercase()\n    return if (uppercase.length > 1) this
else uppercase[0]\n}\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the invariant
locale.\n *\n * This function supports one-to-many character mapping, thus the length of the returned string can be
greater than one.\n * For example, `\\uFB00'.uppercase()` returns `\"\\u0046\\u0046\"`,\n * where `\\uFB00'` is the
LATIN SMALL LIGATURE FF character (`\ufb00`).\n * If this character has no upper case mapping, the result of
`toString()` of this char is returned.\n *\n * @sample samples.text.Chars.uppercase\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c actual inline fun Char.uppercase(): String = toString().asDynamic().toUpperCase().unsafeCast<String>()\n\n/**\n
* Converts this character to title case using Unicode mapping rules of the invariant
 locale.\n *\n * This function performs one-to-one character mapping.\n * To support one-to-many character
mapping use the [titlecase] function.\n * If this character has no mapping equivalent, the result of calling
[uppercaseChar] is returned.\n *\n * @sample samples.text.Chars.titlecase\n */\n@SinceKotlin(\"1.5\")\npublic
actual fun Char.titlecaseChar(): Char = titlecaseCharImpl()\n\n/**\n * Returns `true` if this character is a Unicode
high-surrogate code unit (also known as leading-surrogate code unit).\n */\npublic actual fun
Char.isHighSurrogate(): Boolean = this in
Char.MIN_HIGH_SURROGATE..Char.MAX_HIGH_SURROGATE\n\n/**\n * Returns `true` if this character is a
Unicode low-surrogate code unit (also known as trailing-surrogate code unit).\n */\npublic actual fun
Char.isLowSurrogate(): Boolean = this in
Char.MIN_LOW_SURROGATE..Char.MAX_LOW_SURROGATE\n\n/**\n * Returns the Unicode general
category of this character.\n */\n@SinceKotlin(\"1.5\")\npublic actual val Char.category:
 CharCategory\n    get() = CharCategory.valueOf(getCategoryValue())\n\n/**\n * Returns `true` if this character
(Unicode code point) is defined in Unicode.\n *\n * A character is considered to be defined in Unicode if its
[category] is not [CharCategory.UNASSIGNED].\n */\n@SinceKotlin(\"1.5\")\npublic actual fun Char.isDefined():
Boolean {\n    if (this < '\\u0080') {\n        return true\n    }\n    return getCategoryValue() !=
CharCategory.UNASSIGNED.value\n}\n\n/**\n * Returns `true` if this character is a letter.\n *\n * A character is
considered to be a letter if its [category] is [CharCategory.UPPERCASE_LETTER],\n *
[CharCategory.LOWERCASE_LETTER], [CharCategory.TITLECASE_LETTER],
[CharCategory.MODIFIER_LETTER], or [CharCategory.OTHER_LETTER].\n *\n * @sample
samples.text.Chars.isLetter\n */\n@SinceKotlin(\"1.5\")\npublic actual fun Char.isLetter(): Boolean {\n    if (this in
'a'..'z' || this in 'A'..'Z') {\n        return true\n    }\n    if (this < '\\u0080') {\n        return false\n
    }\n    return isLetterImpl()\n}\n\n/**\n * Returns `true` if this character is a letter or digit.\n *\n * @see isLetter\n
* @see isDigit\n *\n * @sample samples.text.Chars.isLetterOrDigit\n */\n@SinceKotlin(\"1.5\")\npublic actual fun
Char.isLetterOrDigit(): Boolean {\n    if (this in 'a'..'z' || this in 'A'..'Z' || this in '0'..'9') {\n        return true\n    }\n    if
(this < '\\u0080') {\n        return false\n    }\n\n    return isDigitImpl() || isLetterImpl()\n}\n\n/**\n * Returns `true` if
this character is a digit.\n *\n * A character is considered to be a digit if its [category] is
[CharCategory.DECIMAL_DIGIT_NUMBER].\n *\n * @sample samples.text.Chars.isDigit\n
*/\n@SinceKotlin(\"1.5\")\npublic actual fun Char.isDigit(): Boolean {\n    if (this in '0'..'9') {\n        return true\n
}\n    if (this < '\\u0080') {\n        return false\n    }\n    return isDigitImpl()\n}\n\n/**\n * Returns `true` if this
character is upper case.\n *\n * A character is considered to
 be an upper case character if its [category] is [CharCategory.UPPERCASE_LETTER],\n * or it has contributory
property `Other_Uppercase` as defined by the Unicode Standard.\n *\n * @sample
samples.text.Chars.isUpperCase\n */\n@SinceKotlin(\"1.5\")\npublic actual fun Char.isUpperCase(): Boolean {\n
if (this in 'A'..'Z') {\n        return true\n    }\n    if (this < '\\u0080') {\n        return false\n    }\n    return
isUpperCaseImpl()\n}\n\n/**\n * Returns `true` if this character is lower case.\n *\n * A character is considered to
be a lower case character if its [category] is [CharCategory.LOWERCASE_LETTER],\n * or it has contributory
property `Other_Lowercase` as defined by the Unicode Standard.\n *\n * @sample
samples.text.Chars.isLowerCase\n */\n@SinceKotlin(\"1.5\")\npublic actual fun Char.isLowerCase(): Boolean {\n
```

if (this in 'a'..'z') {\n return true\n }\n if (this < '\\u0080') {\n return false\n }\n return isLowerCaseImpl()\n}\n\n/**\n * Returns
`true` if this character is a title case letter.\n *\n * A character is considered to be a title case letter if its [category] is [CharCategory.TITLECASE_LETTER].\n *\n * @sample samples.text.Chars.isTitleCase\n */\n@SinceKotlin(\"1.5\")\npublic actual fun Char.isTitleCase(): Boolean {\n if (this < '\\u0080') {\n return false\n }\n return getCategoryValue() == CharCategory.TITLECASE_LETTER.value\n}\n\n/**\n * Returns `true` if this character is an ISO control character.\n *\n * A character is considered to be an ISO control character if its [category] is [CharCategory.CONTROL],\n * meaning the Char is in the range `'\\u0000'..'\\u001F'` or in the range `'\\u007F'..'\\u009F'`.\n *\n * @sample samples.text.Chars.isISOControl\n */\n@SinceKotlin(\"1.5\")\npublic actual fun Char.isISOControl(): Boolean {\n return this <= '\\u001F' || this in '\\u007F'..'\\u009F'\n}\n\n/**\n * Determines whether a character is whitespace according to the Unicode standard.\n * Returns `true`
if the character is whitespace.\n *\n * @sample samples.text.Chars.isWhitespace\n */\npublic actual fun Char.isWhitespace(): Boolean = isWhitespaceImpl()","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\nimport kotlin.js.RegExp\n\n/**\n * Converts the characters in the specified array to a string.\n */\n@SinceKotlin(\"1.2\")\n@Deprecated(\"Use CharArray.concatToString() instead\", ReplaceWith(\"chars.concatToString()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.5\")\npublic actual fun String(chars: CharArray): String {\n var result = \"\"\n for (char in chars) {\n result += char\n }\n return result\n}\n\n/**\n * Converts the characters from a portion of the specified array to a string.\n *\n * @throws IndexOutOfBoundsException if either [offset] or [length]
are less than zero\n * or `offset + length` is out of [chars] array bounds.\n */\n@SinceKotlin(\"1.2\")\n@Deprecated(\"Use CharArray.concatToString(startIndex, endIndex) instead\", ReplaceWith(\"chars.concatToString(offset, offset + length)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.5\")\npublic actual fun String(chars: CharArray, offset: Int, length: Int): String {\n if (offset < 0 || length < 0 || chars.size - offset < length)\n throw IndexOutOfBoundsException(\"size: ${chars.size}; offset: $offset; length: $length\")\n var result = \"\"\n for (index in offset until offset + length) {\n result += chars[index]\n }\n return result\n}\n\n/**\n * Concatenates characters in this [CharArray] into a String.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun CharArray.concatToString(): String {\n var result = \"\"\n for (char in this) {\n result += char\n }\n return result\n}\n\n/**\n
* Concatenates characters in this [CharArray] or its subrange into a String.\n *\n * @param startIndex the beginning (inclusive) of the subrange of characters, 0 by default.\n * @param endIndex the end (exclusive) of the subrange of characters, size of this array by default.\n *\n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun CharArray.concatToString(startIndex: Int = 0, endIndex: Int = this.size): String {\n AbstractList.checkBoundsIndexes(startIndex, endIndex, this.size)\n var result = \"\"\n for (index in startIndex until endIndex) {\n result += this[index]\n }\n return result\n}\n\n/**\n * Returns a [CharArray] containing characters of this string.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun String.toCharArray(): CharArray {\n return CharArray(length) { get(it) }\n}\n\n/**\n * Returns a [CharArray] containing characters of this string or its substring.\n *\n * @param startIndex the beginning (inclusive) of the substring, 0 by default.\n * @param endIndex the end (exclusive) of the substring, length of this string by default.\n *\n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the length of this string.\n * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress(\"ACTUAL_FUNCTI

ON_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun String.toCharArray(startIndex: Int = 0, endIndex: Int = this.length): CharArray {\n AbstractList.checkBoundsIndexes(startIndex, endIndex, length)\n return CharArray(endIndex - startIndex)
{ get(startIndex + it) }\n}\n\n/**\n * Decodes a string from the bytes in UTF-8 encoding in this array.\n *\n * Malformed byte sequences are replaced by the replacement char `\\uFFFD`.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun ByteArray.decodeToString(): String {\n return decodeUtf8(this, 0, size, false)\n}\n\n/**\n * Decodes a string from the bytes in UTF-8 encoding in this array or its subrange.\n *\n * @param startIndex the beginning (inclusive) of the subrange to decode, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to decode, size of this array by default.\n * @param throwOnInvalidSequence specifies whether to throw an exception on malformed byte sequence or replace it by the replacement char `\\uFFFD`.\n *\n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].\n
* @throws CharacterCodingException if the byte array contains malformed UTF-8 byte sequence and [throwOnInvalidSequence] is true.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun ByteArray.decodeToString(\n startIndex: Int = 0,\n endIndex: Int = this.size,\n throwOnInvalidSequence: Boolean = false\n): String {\n AbstractList.checkBoundsIndexes(startIndex, endIndex, this.size)\n return decodeUtf8(this, startIndex, endIndex, throwOnInvalidSequence)\n}\n\n/**\n * Encodes this string to an array of bytes in UTF-8 encoding.\n *\n * Any malformed char sequence is replaced by the replacement byte sequence.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun String.encodeToByteArray(): ByteArray {\n return encodeUtf8(this, 0, length, false)\n}\n\n/**\n * Encodes this string or its substring to an array of bytes in UTF-8 encoding.\n
*\n * @param startIndex the beginning (inclusive) of the substring to encode, 0 by default.\n * @param endIndex the end (exclusive) of the substring to encode, length of this string by default.\n * @param throwOnInvalidSequence specifies whether to throw an exception on malformed char sequence or replace.\n *\n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or [endIndex] is greater than the length of this string.\n * @throws IllegalArgumentException if [startIndex] is greater than [endIndex].\n * @throws CharacterCodingException if this string contains malformed char sequence and [throwOnInvalidSequence] is true.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun String.encodeToByteArray(\n startIndex: Int = 0,\n endIndex: Int = this.length,\n throwOnInvalidSequence: Boolean = false\n): ByteArray {\n AbstractList.checkBoundsIndexes(startIndex,
endIndex, length)\n return encodeUtf8(this, startIndex, endIndex, throwOnInvalidSequence)\n}\n\n/**\n * Returns a copy of this string converted to upper case using the rules of the default locale.\n */\n@Deprecated(\"Use uppercase() instead.\", ReplaceWith(\"uppercase()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\n@kotlin.internal.InlineOnly\npublic actual inline fun String.toUpperCase(): String = asDynamic().toUpperCase()\n\n/**\n * Returns a copy of this string converted to upper case using Unicode mapping rules of the invariant locale.\n *\n * This function supports one-to-many and many-to-one character mapping,\n * thus the length of the returned string can be different from the length of the original string.\n *\n * @sample samples.text.Strings.uppercase\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic actual inline fun String.uppercase(): String = asDynamic().toUpperCase()\n\n/**\n * Returns a copy of this
string converted to lower case using the rules of the default locale.\n */\n@Deprecated(\"Use lowercase() instead.\", ReplaceWith(\"lowercase()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\n@kotlin.internal.InlineOnly\npublic actual inline fun String.toLowerCase(): String = asDynamic().toLowerCase()\n\n/**\n * Returns a copy of this string converted to lower case using Unicode

mapping rules of the invariant locale.\n *\n * This function supports one-to-many and many-to-one character mapping,\n * thus the length of the returned string can be different from the length of the original string.\n *\n * @sample samples.text.Strings.lowercase\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic actual inline fun String.lowercase(): String = asDynamic().toLowerCase()\n\n@kotlin.internal.InlineOnly\ninternal actual inline fun String.nativeIndexOf(str: String, fromIndex: Int): Int = asDynamic().indexOf(str, fromIndex)\n\n@kotlin.internal.InlineOnly\ninternal
actual inline fun String.nativeLastIndexOf(str: String, fromIndex: Int): Int = asDynamic().lastIndexOf(str, fromIndex)\n\n@kotlin.internal.InlineOnly\n@kotlin.js.JsPolyfill(\"\"\"\nif (typeof String.prototype.startsWith === \"undefined\") {\n Object.defineProperty(String.prototype, \"startsWith\", {\n value: function (searchString, position) {\n position = position || 0;\n return this.lastIndexOf(searchString, position) === position;\n }\n });\n}\n\"\"\")\ninternal inline fun String.nativeStartsWith(s: String, position: Int): Boolean = asDynamic().startsWith(s, position)\n\n@kotlin.internal.InlineOnly\n@kotlin.js.JsPolyfill(\"\"\"\nif (typeof String.prototype.endsWith === \"undefined\") {\n Object.defineProperty(String.prototype, \"endsWith\", {\n value: function (searchString, position) {\n var subjectString = this.toString();\n if (position === undefined || position > subjectString.length)
{\n position = subjectString.length;\n }\n position -= searchString.length;\n var lastIndex = subjectString.indexOf(searchString, position);\n return lastIndex !== -1 && lastIndex === position;\n }\n });\n}\n\"\"\")\ninternal inline fun String.nativeEndsWith(s: String): Boolean = asDynamic().endsWith(s)\n\n@kotlin.internal.InlineOnly\npublic actual inline fun String.substring(startIndex: Int): String = asDynamic().substring(startIndex)\n\n@kotlin.internal.InlineOnly\npublic actual inline fun String.substring(startIndex: Int, endIndex: Int): String = asDynamic().substring(startIndex, endIndex)\n\n@Deprecated(\"Use String.plus() instead\", ReplaceWith(\"this + str\"))\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n@kotlin.internal.InlineOnly\npublic inline fun String.concat(str: String): String = asDynamic().concat(str)\n\n@Deprecated(\"Use Regex.findAll() instead or invoke matches() on String dynamically:
this.asDynamic().match(regex)\")\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n@kotlin.internal.InlineOnly\npublic inline fun String.match(regex: String): Array<String>? = asDynamic().match(regex)\n\n//native public fun String.trim(): String\n//TODO: String.replace to implement effective trimLeading and trimTrailing\n\n@kotlin.internal.InlineOnly\ninternal inline fun String.nativeReplace(pattern: RegExp, replacement: String): String = asDynamic().replace(pattern, replacement)\n\n/**\n * Compares two strings lexicographically, optionally ignoring case differences.\n *\n * If [ignoreCase] is true, the result of `Char.uppercaseChar().lowercaseChar()` on each character is compared.\n */\n@SinceKotlin(\"1.2\")\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun String.compareTo(other: String, ignoreCase: Boolean = false): Int {\n if (ignoreCase) {\n val n1 = this.length\n val n2 = other.length\n val min = minOf(n1, n2)\n if (min == 0)
return n1 - n2\n for (index in 0 until min) {\n var thisChar = this[index]\n var otherChar = other[index]\n\n if (thisChar != otherChar) {\n thisChar = thisChar.uppercaseChar()\n otherChar = otherChar.uppercaseChar()\n\n if (thisChar != otherChar) {\n thisChar = thisChar.lowercaseChar()\n otherChar = otherChar.lowercaseChar()\n\n if (thisChar != otherChar) {\n return thisChar.compareTo(otherChar)\n }\n }\n }\n }\n return n1 - n2\n } else {\n return compareTo(other)\n }\n}\n\n/**\n * Returns `true` if the contents of this char sequence are equal to the contents of the specified [other],\n * i.e. both char sequences contain the same number of the same characters in the same order.\n *\n * @sample samples.text.Strings.contentEquals\n */\n@SinceKotlin(\"1.5\")\npublic
actual infix fun CharSequence?.contentEquals(other: CharSequence?): Boolean = contentEqualsImpl(other)\n\n/**\n * Returns `true` if the contents of this char sequence are equal to the contents of the specified [other], optionally ignoring case difference.\n *\n * @param ignoreCase `true` to ignore character case

when comparing contents.\n *\n * @sample samples.text.Strings.contentEquals\n */\n@SinceKotlin(\"1.5\")\npublic actual fun CharSequence?.contentEquals(other: CharSequence?, ignoreCase: Boolean): Boolean {\n return if (ignoreCase)\n this.contentEqualsIgnoreCaseImpl(other)\n else\n this.contentEqualsImpl(other)\n}\n\n\nprivate val STRING_CASE_INSENSITIVE_ORDER = Comparator<String> { a, b -> a.compareTo(b, ignoreCase = true) }\n\n@SinceKotlin(\"1.2\")\npublic actual val String.Companion.CASE_INSENSITIVE_ORDER: Comparator<String>\n get() = STRING_CASE_INSENSITIVE_ORDER\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"CharsKt\")\n\npackage kotlin.text\n\n/**\n * Returns the numeric value of the decimal digit that this Char represents.\n * Throws an exception if this Char is not a valid decimal digit.\n *\n * A Char is considered to represent a decimal digit if [isDigit] is true for the Char.\n * In this case, the Unicode decimal digit value of the character is returned.\n *\n * @sample samples.text.Chars.digitToInt\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Char.digitToInt(): Int {\n return digitOf(this, 10).also {\n if (it < 0) throw IllegalArgumentException(\"Char $this is not a decimal digit\")\n }\n}\n\n/**\n * Returns the numeric value of the digit that this Char represents in the specified [radix].\n * Throws an exception if the [radix]
is not in the range `2..36` or if this Char is not a valid digit in the specified [radix].\n *\n * A Char is considered to represent a digit in the specified [radix] if at least one of the following is true:\n * - [isDigit] is `true` for the Char and the Unicode decimal digit value of the character is less than the specified [radix]. In this case the decimal digit value is returned.\n * - The Char is one of the uppercase Latin letters 'A' through 'Z' and its [code] is less than `radix + 'A'.code - 10`. In this case, `this.code - 'A'.code + 10` is returned.\n * - The Char is one of the lowercase Latin letters 'a' through 'z' and its [code] is less than `radix + 'a'.code - 10`. In this case, `this.code - 'a'.code + 10` is returned.\n * - The Char is one of the fullwidth Latin capital letters '\\uFF21' through '\\uFF3A' and its [code] is less than `radix + 0xFF21 - 10`. In this case, `this.code - 0xFF21 + 10` is returned.\n * - The Char is one of the fullwidth Latin small letters '\\uFF41'
through '\\uFF5A' and its [code] is less than `radix + 0xFF41 - 10`. In this case, `this.code - 0xFF41 + 10` is returned.\n *\n * @sample samples.text.Chars.digitToInt\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Char.digitToInt(radix: Int): Int {\n return digitToIntOrNull(radix) ?: throw IllegalArgumentException(\"Char $this is not a digit in the given radix=$radix\")\n}\n\n/**\n *\n * Returns the numeric value of the decimal digit that this Char represents, or `null` if this Char is not a valid decimal digit.\n *\n * A Char is considered to represent a decimal digit if [isDigit] is true for the Char.\n * In this case, the Unicode decimal digit value of the character is returned.\n *\n * @sample samples.text.Chars.digitToIntOrNull\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Char.digitToIntOrNull(): Int? {\n return digitOf(this, 10).takeIf { it >= 0 }\n}\n\n/**\n * Returns the numeric value
of the digit that this Char represents in the specified [radix], or `null` if this Char is not a valid digit in the specified [radix].\n * Throws an exception if the [radix] is not in the range `2..36`.\n *\n * A Char is considered to represent a digit in the specified [radix] if at least one of the following is true:\n * - [isDigit] is `true` for the Char and the Unicode decimal digit value of the character is less than the specified [radix]. In this case the decimal digit value is returned.\n * - The Char is one of the uppercase Latin letters 'A' through 'Z' and its [code] is less than `radix + 'A'.code - 10`. In this case, `this.code - 'A'.code + 10` is returned.\n * - The Char is one of the lowercase Latin letters 'a' through 'z' and its [code] is less than `radix + 'a'.code - 10`. In this case, `this.code - 'a'.code + 10` is returned.\n * - The Char is one of the fullwidth Latin capital letters '\\uFF21' through '\\uFF3A' and its [code] is less than `radix + 0xFF21 - 10`.

In this case, `this.code - 0xFF21 + 10` is returned.\n * - The Char is one of the fullwidth Latin small letters '\\uFF41' through '\\uFF5A' and its [code] is less than `radix + 0xFF41 - 10`. In this case, `this.code - 0xFF41 + 10` is returned.\n *\n * @sample samples.text.Chars.digitToIntOrNull\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Char.digitToIntOrNull(radix: Int): Int? {\n   checkRadix(radix)\n   return digitOf(this, radix).takeIf { it >= 0 }\n}\n\n/**\n * Returns the Char that represents this decimal digit.\n * Throws an exception if this value is not in the range `0..9`.\n *\n * If this value is in `0..9`, the decimal digit Char with code `'0'.code + this` is returned.\n *\n * @sample samples.text.Chars.digitToChar\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Int.digitToChar(): Char {\n   if (this in 0..9) {\n   return '0' + this\n   }\n   throw IllegalArgumentException(\"Int $this is not a decimal digit\")\n}\n\n/**\n * Returns the Char that represents this numeric digit value in the specified [radix].\n * Throws an exception if the [radix] is not in the range `2..36` or if this value is not in the range `0 until radix`.\n *\n * If this value is less than `10`, the decimal digit Char with code `'0'.code + this` is returned.\n * Otherwise, the uppercase Latin letter with code `'A'.code + this - 10` is returned.\n *\n * @sample samples.text.Chars.digitToChar\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Int.digitToChar(radix: Int): Char {\n   if (radix !in 2..36) {\n   throw IllegalArgumentException(\"Invalid radix: $radix. Valid radix values are in range 2..36\")\n   }\n   if (this < 0 || this >= radix) {\n   throw IllegalArgumentException(\"Digit $this does not represent a valid digit in radix $radix\")\n   }\n   return if (this < 10) {\n   '0' + this\n   } else {\n   'A' + this - 10\n   }\n}\n\n/**\n * Converts this character to lower case using Unicode mapping rules of the invariant locale.\n */\n@Deprecated(\"Use lowercaseChar() instead.\", ReplaceWith(\"lowercaseChar()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic expect fun Char.toLowerCase(): Char\n\n/**\n * Converts this character to lower case using Unicode mapping rules of the invariant locale.\n *\n * This function performs one-to-one character mapping.\n * To support one-to-many character mapping use the [lowercase] function.\n * If this character has no mapping equivalent, the character itself is returned.\n *\n * @sample samples.text.Chars.lowercase\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Char.lowercaseChar(): Char\n\n/**\n * Converts this character to lower case using Unicode mapping rules of the invariant locale.\n *\n * This function supports one-to-many character mapping, thus the length of the returned string can be greater than one.\n * For example, `'\\u0130'.lowercase()` returns `\"\\u0069\\u0307\"`,\n * where `'\\u0130'` is the LATIN CAPITAL LETTER I WITH DOT ABOVE character (`\u0130`).\n * If this character has no lower case mapping, the result of `toString()` of this char is returned.\n *\n * @sample samples.text.Chars.lowercase\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Char.lowercase(): String\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the invariant locale.\n */\n@Deprecated(\"Use uppercaseChar() instead.\", ReplaceWith(\"uppercaseChar()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic expect fun Char.toUpperCase(): Char\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the invariant locale.\n *\n * This function performs one-to-one character mapping.\n * To support one-to-many character mapping use the [uppercase] function.\n * If this character has no mapping equivalent, the character itself is returned.\n *\n * @sample samples.text.Chars.uppercase\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Char.uppercaseChar(): Char\n\n/**\n * Converts this character to upper case using Unicode mapping rules of the invariant locale.\n *\n * This function supports one-to-many character mapping, thus the length of the returned string can be greater than one.\n * For example, `'\\uFB00'.uppercase()` returns `\"\\u0046\\u0046\"`,\n * where `'\\uFB00'` is the LATIN SMALL LIGATURE FF character (`\ufb00`).\n * If this character has no upper case mapping, the result of `toString()` of this char is returned.\n *\n * @sample samples.text.Chars.uppercase\n

*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Char.uppercase(): String\n\n/**\n * Converts this character to title case using Unicode mapping rules of the invariant locale.\n *\n * This function performs one-to-one character mapping.\n *
To support one-to-many character mapping use the [titlecase] function.\n * If this character has no mapping equivalent, the result of calling [uppercaseChar] is returned.\n *\n * @sample samples.text.Chars.titlecase\n */\n@SinceKotlin(\"1.5\")\npublic expect fun Char.titlecaseChar(): Char\n\n/**\n * Converts this character to title case using Unicode mapping rules of the invariant locale.\n *\n * This function supports one-to-many character mapping, thus the length of the returned string can be greater than one.\n * For example, `'\\uFB00'.titlecase()` returns `\"\\u0046\\u0066\"`,\n * where `'\\uFB00'` is the LATIN SMALL LIGATURE FF character (`\ufb00`).\n * If this character has no title case mapping, the result of [uppercase] is returned instead.\n *\n * @sample samples.text.Chars.titlecase\n */\n@SinceKotlin(\"1.5\")\npublic fun Char.titlecase(): String = titlecaseImpl()\n\n/**\n * Concatenates this Char and a String.\n *\n * @sample samples.text.Chars.plus\n */\n@kotlin.internal.InlineOnly\npublic
inline operator fun Char.plus(other: String): String = this.toString() + other\n\n/**\n * Returns `true` if this character is equal to the [other] character, optionally ignoring character case.\n *\n * Two characters are considered equal ignoring case if `Char.uppercaseChar().lowercaseChar()` on each character produces the same result.\n *\n * @param ignoreCase `true` to ignore character case when comparing characters. By default `false`.\n * @sample samples.text.Chars.equals\n */\npublic fun Char.equals(other: Char, ignoreCase: Boolean = false): Boolean {\n   if (this == other) return true\n   if (!ignoreCase) return false\n\n   val thisUpper = this.uppercaseChar()\n   val otherUpper = other.uppercaseChar()\n\n   return thisUpper == otherUpper || thisUpper.lowercaseChar() == otherUpper.lowercaseChar()\n}\n\n/**\n * Returns `true` if this character is a Unicode surrogate code unit.\n */\npublic fun Char.isSurrogate(): Boolean = this in Char.MIN_SURROGATE..Char.MAX_SURROGATE\n\n/**\n * Returns the Unicode general category of this character.\n */\n@SinceKotlin(\"1.5\")\npublic expect val Char.category: CharCategory\n\n/**\n * Returns `true` if this character (Unicode code point) is defined in Unicode.\n *\n * A character is considered to be defined in Unicode if its [category] is not [CharCategory.UNASSIGNED].\n */\n@SinceKotlin(\"1.5\")\npublic expect fun Char.isDefined(): Boolean\n\n/**\n * Returns `true` if this character is a letter.\n *\n * A character is considered to be a letter if its [category] is [CharCategory.UPPERCASE_LETTER],\n * [CharCategory.LOWERCASE_LETTER], [CharCategory.TITLECASE_LETTER], [CharCategory.MODIFIER_LETTER], or [CharCategory.OTHER_LETTER].\n *\n * @sample samples.text.Chars.isLetter\n */\n@SinceKotlin(\"1.5\")\npublic expect fun Char.isLetter(): Boolean\n\n/**\n * Returns `true` if this character is a letter or digit.\n *\n * @see isLetter\n * @see isDigit\n *\n * @sample samples.text.Chars.isLetterOrDigit\n */\n@SinceKotlin(\"1.5\")\npublic
expect fun Char.isLetterOrDigit(): Boolean\n\n/**\n * Returns `true` if this character is a digit.\n *\n * A character is considered to be a digit if its [category] is [CharCategory.DECIMAL_DIGIT_NUMBER].\n *\n * @sample samples.text.Chars.isDigit\n */\n@SinceKotlin(\"1.5\")\npublic expect fun Char.isDigit(): Boolean\n\n/**\n * Returns `true` if this character is upper case.\n *\n * A character is considered to be an upper case character if its [category] is [CharCategory.UPPERCASE_LETTER],\n * or it has contributory property `Other_Uppercase` as defined by the Unicode Standard.\n *\n * @sample samples.text.Chars.isUpperCase\n */\n@SinceKotlin(\"1.5\")\npublic expect fun Char.isUpperCase(): Boolean\n\n/**\n * Returns `true` if this character is lower case.\n *\n * A character is considered to be a lower case character if its [category] is [CharCategory.LOWERCASE_LETTER],\n * or it has contributory property `Other_Lowercase` as defined by the Unicode Standard.\n
*\n * @sample samples.text.Chars.isLowerCase\n */\n@SinceKotlin(\"1.5\")\npublic expect fun Char.isLowerCase(): Boolean\n\n/**\n * Returns `true` if this character is a title case letter.\n *\n * A character is considered to be a title case letter if its [category] is [CharCategory.TITLECASE_LETTER].\n *\n * @sample samples.text.Chars.isTitleCase\n */\n@SinceKotlin(\"1.5\")\npublic expect fun Char.isTitleCase(): Boolean\n\n/**\n * Returns `true` if this character is an ISO control character.\n *\n * A character is considered to be an ISO control

character if its [category] is [CharCategory.CONTROL],\n * meaning the Char is in the range `'\\u0000'..'\\u001F'` or in the range `'\\u007F'..'\\u009F'`.\n *\n * @sample samples.text.Chars.isISOControl\n */\n@SinceKotlin(\"1.5\")\npublic expect fun Char.isISOControl(): Boolean\n\n/**\n * Determines whether a character is whitespace according to the Unicode standard.\n * Returns `true` if the character is whitespace.\n *\n * @sample samples.text.Chars.isWhitespace\n */\npublic expect fun Char.isWhitespace(): Boolean\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\n\n/**\n * Creates a Char with the specified [code], or throws an exception if the [code] is out of `Char.MIN_VALUE.code..Char.MAX_VALUE.code`.\n *\n * If the program that calls this function is written in a way that only valid [code] is passed as the argument,\n * using the overload that takes a [UShort] argument is preferable (`Char(intValue.toUShort())`).\n * That overload doesn't check validity of the argument, and may improve program performance when the function is called routinely inside a loop.\n *\n * @sample samples.text.Chars.charFromCode\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Char(code:
Int): Char {\n   if (code < Char.MIN_VALUE.code || code > Char.MAX_VALUE.code) {\n   throw IllegalArgumentException(\"Invalid Char code: $code\")\n   }\n   return code.toChar()\n}\n\n/**\n * Creates a Char with the specified [code].\n *\n * @sample samples.text.Chars.charFromCode\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress(\"NO_ACTUAL_FOR_EXPECT\")\npublic expect fun Char(code: UShort): Char\n\n/**\n * Returns the code of this Char.\n *\n * Code of a Char is the value it was constructed with, and the UTF-16 code unit corresponding to this Char.\n *\n * @sample samples.text.Chars.code\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\n@Suppress(\"DEPRECATION\")\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline val Char.code: Int get() = this.toInt()\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"SequencesKt\")\n\npackage kotlin.sequences\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.random.*\n\n/**\n * Returns `true` if [element] is found in the sequence.\n *\n * The operation is _terminal_.\n */\npublic operator fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.contains(element: T): Boolean {\n   return indexOf(element) >= 0\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this sequence.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic fun <T> Sequence<T>.elementAt(index: Int): T {\n   return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"Sequence
doesn't contain element at index $index.\") }\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this sequence.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n */\npublic fun <T> Sequence<T>.elementAtOrElse(index: Int, defaultValue: (Int) -> T): T {\n   if (index < 0)\n   return defaultValue(index)\n   val iterator = iterator()\n   var count = 0\n   while (iterator.hasNext()) {\n   val element = iterator.next()\n   if (index == count++)\n   return element\n   }\n   return defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this sequence.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\npublic fun <T> Sequence<T>.elementAtOrNull(index: Int): T? {\n   if (index
< 0)\n   return null\n   val iterator = iterator()\n   var count = 0\n   while (iterator.hasNext()) {\n   val element = iterator.next()\n   if (index == count++)\n   return element\n   }\n   return null\n}\n\n/**\n *

Returns the first element matching the given [predicate], or `null` if no such element was found.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.find(predicate: (T) -> Boolean): T? {\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.findLast(predicate: (T) -> Boolean): T? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the first element.\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n */\npublic fun <T> Sequence<T>.first(): T {\n val iterator = iterator()\n if (!iterator.hasNext())\n throw NoSuchElementException(\"Sequence is empty.\")\n return iterator.next()\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T> Sequence<T>.first(predicate: (T) -> Boolean): T {\n for (element in this) if (predicate(element)) return element\n throw NoSuchElementException(\"Sequence contains no element matching the predicate.\")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to elements of this sequence in iteration order,\n * or throws [NoSuchElementException] if no non-null value was produced.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any> Sequence<T>.firstNotNullOf(transform: (T) -> R?): R {\n return firstNotNullOfOrNull(transform) ?: throw NoSuchElementException(\"No element of the sequence was transformed to a non-null value.\")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to elements of this sequence in iteration order,\n * or `null` if no non-null value was produced.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Any> Sequence<T>.firstNotNullOfOrNull(transform: (T) -> R?): R? {\n for (element in this) {\n val result = transform(element)\n if (result != null) {\n return result\n }\n }\n return null\n}\n\n/**\n * Returns the first element, or `null` if the sequence is empty.\n *\n * The operation is _terminal_.\n */\npublic fun <T> Sequence<T>.firstOrNull(): T? {\n val iterator = iterator()\n if (!iterator.hasNext())\n return null\n return iterator.next()\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T> Sequence<T>.firstOrNull(predicate: (T) -> Boolean): T? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns first index of [element], or -1 if the sequence does not contain element.\n *\n * The operation is _terminal_.\n */\npublic fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.indexOf(element: T): Int {\n var index = 0\n for (item in this) {\n checkIndexOverflow(index)\n if (element == item)\n return index\n index++\n }\n return -1\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the sequence does not contain such element.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T> Sequence<T>.indexOfFirst(predicate: (T) -> Boolean): Int {\n var index = 0\n for (item in this) {\n checkIndexOverflow(index)\n if (predicate(item))\n return index\n index++\n }\n return -1\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the sequence does not contain such element.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T> Sequence<T>.indexOfLast(predicate: (T) -> Boolean): Int {\n var lastIndex = -1\n var index = 0\n for (item in this) {\n checkIndexOverflow(index)\n if (predicate(item))\n lastIndex = index\n index++\n }\n return lastIndex\n}\n\n/**\n * Returns the last element.\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n

```
*/\npublic fun <T> Sequence<T>.last(): T {\n    val iterator = iterator()\n    if (!iterator.hasNext())\n        throw NoSuchElementException(\"Sequence is empty.\")\n    var last = iterator.next()\n    while (iterator.hasNext())\n        last = iterator.next()\n    return last\n}\n\n/**\n * Returns the last element matching the given [predicate].\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun <T> Sequence<T>.last(predicate: (T) -> Boolean): T {\n    var last: T? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            last = element\n            found = true\n        }\n    }\n    if (!found) throw NoSuchElementException(\"Sequence contains no element matching the predicate.\")\n    @Suppress(\"UNCHECKED_CAST\")\n    return last as T\n}\n\n/**\n * Returns last index of [element], or -1 if the sequence does not contain element.\n *\n * The operation is _terminal_.\n */\npublic fun <@kotlin.internal.OnlyInputTypes T> Sequence<T>.lastIndexOf(element: T): Int {\n    var lastIndex = -1\n    var index = 0\n    for (item in this) {\n        checkIndexOverflow(index)\n        if (element == item)\n            lastIndex = index\n        index++\n    }\n    return lastIndex\n}\n\n/**\n * Returns the last element, or `null` if the sequence is empty.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic fun <T> Sequence<T>.lastOrNull(): T? {\n    val iterator = iterator()\n    if (!iterator.hasNext())\n        return null\n    var last = iterator.next()\n    while (iterator.hasNext())\n        last = iterator.next()\n    return last\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Elements.last\n */\npublic inline fun <T> Sequence<T>.lastOrNull(predicate: (T) -> Boolean): T? {\n    var last: T? = null\n    for (element in this) {\n        if (predicate(element)) {\n            last = element\n        }\n    }\n    return last\n}\n\n/**\n * Returns the single element, or throws an exception if the sequence is empty or has more than one element.\n *\n * The operation is _terminal_.\n */\npublic fun <T> Sequence<T>.single(): T {\n    val iterator = iterator()\n    if (!iterator.hasNext())\n        throw NoSuchElementException(\"Sequence is empty.\")\n    val single = iterator.next()\n    if (iterator.hasNext())\n        throw IllegalArgumentException(\"Sequence has more than one element.\")\n    return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T> Sequence<T>.single(predicate: (T) -> Boolean): T {\n    var single: T? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) throw IllegalArgumentException(\"Sequence contains more than one matching element.\")\n            single = element\n            found = true\n        }\n    }\n    if (!found) throw NoSuchElementException(\"Sequence contains no element matching the predicate.\")\n    @Suppress(\"UNCHECKED_CAST\")\n    return single as T\n}\n\n/**\n * Returns single element, or `null` if the sequence is empty or has more than one element.\n *\n * The operation is _terminal_.\n */\npublic fun <T> Sequence<T>.singleOrNull(): T? {\n    val iterator = iterator()\n    if (!iterator.hasNext())\n        return null\n    val single = iterator.next()\n    if (iterator.hasNext())\n        return null\n    return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T> Sequence<T>.singleOrNull(predicate: (T) -> Boolean): T? {\n    var single: T? = null\n    var found = false\n    for (element in this) {\n        if (predicate(element)) {\n            if (found) return null\n            single = element\n            found = true\n        }\n    }\n    if (!found) return null\n    return single\n}\n\n/**\n * Returns a sequence containing all elements except first [n] elements.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun <T> Sequence<T>.drop(n: Int): Sequence<T> {\n    require(n >= 0) { \"Requested element count $n is less than zero.\" }\n    return when {\n        n == 0 -> this\n        this is DropTakeSequence -> this.drop(n)\n        else -> DropSequence(this, n)\n    }\n}\n\n/**\n * Returns a sequence containing all elements except first elements that satisfy the given [predicate].\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\npublic fun <T>
```

Sequence<T>.dropWhile(predicate: (T) -> Boolean): Sequence<T> {\n   return DropWhileSequence(this, predicate)\n}\n\n/**\n * Returns a sequence containing only elements matching the given [predicate].\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic fun <T> Sequence<T>.filter(predicate: (T) -> Boolean): Sequence<T> {\n   return FilteringSequence(this, true, predicate)\n}\n\n/**\n * Returns a sequence containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n *
 @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic fun <T> Sequence<T>.filterIndexed(predicate: (index: Int, T) -> Boolean): Sequence<T> {\n   // TODO: Rewrite with generalized MapFilterIndexingSequence\n   return TransformingSequence(FilteringSequence(IndexingSequence(this), true, { predicate(it.index, it.value) }), { it.value })\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <T, C : MutableCollection<in T>> Sequence<T>.filterIndexedTo(destination: C, predicate: (index: Int, T) -> Boolean): C {\n   forEachIndexed { index, element ->\n   if (predicate(index, element)) destination.add(element)\n   }\n   return destination\n}\n\n/**\n
 * Returns a sequence containing all elements that are instances of specified type parameter R.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Filtering.filterIsInstance\n */\npublic inline fun <reified R> Sequence<*>.filterIsInstance(): Sequence<@kotlin.internal.NoInfer R> {\n   @Suppress(\"UNCHECKED_CAST\")\n   return filter { it is R } as Sequence<R>\n}\n\n/**\n * Appends all elements that are instances of specified type parameter R to the given [destination].\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Filtering.filterIsInstanceTo\n */\npublic inline fun <reified R, C : MutableCollection<in R>> Sequence<*>.filterIsInstanceTo(destination: C): C {\n   for (element in this) if (element is R) destination.add(element)\n   return destination\n}\n\n/**\n * Returns a sequence containing all elements not matching the given [predicate].\n *\n * The operation is _intermediate_
 and _stateless_.\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\npublic fun <T> Sequence<T>.filterNot(predicate: (T) -> Boolean): Sequence<T> {\n   return FilteringSequence(this, false, predicate)\n}\n\n/**\n * Returns a sequence containing all elements that are not `null`.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Filtering.filterNotNull\n */\npublic fun <T : Any> Sequence<T?>.filterNotNull(): Sequence<T> {\n   @Suppress(\"UNCHECKED_CAST\")\n   return filterNot { it == null } as Sequence<T>\n}\n\n/**\n * Appends all elements that are not `null` to the given [destination].\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Filtering.filterNotNullTo\n */\npublic fun <C : MutableCollection<in T>, T : Any> Sequence<T?>.filterNotNullTo(destination: C): C {\n   for (element in this) if (element != null) destination.add(element)\n   return destination\n}\n\n/**\n
 * Appends all elements not matching the given [predicate] to the given [destination].\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <T, C : MutableCollection<in T>> Sequence<T>.filterNotTo(destination: C, predicate: (T) -> Boolean): C {\n   for (element in this) if (!predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <T, C : MutableCollection<in T>> Sequence<T>.filterTo(destination: C, predicate: (T) -> Boolean): C {\n   for (element in this) if (predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Returns a sequence containing first [n] elements.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @throws IllegalArgumentException

if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun <T> Sequence<T>.take(n: Int): Sequence<T> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return when {\n n == 0 -> emptySequence()\n this is DropTakeSequence -> this.take(n)\n else -> TakeSequence(this, n)\n }\n}\n\n/**\n * Returns a sequence containing first elements satisfying the given [predicate].\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\npublic fun <T> Sequence<T>.takeWhile(predicate: (T) -> Boolean): Sequence<T> {\n return TakeWhileSequence(this, predicate)\n}\n\n/**\n * Returns a sequence that yields elements of this sequence sorted according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n *\n * The operation is _intermediate_ and _stateful_.\n */\npublic fun <T : Comparable<T>> Sequence<T>.sorted(): Sequence<T> {\n return object : Sequence<T> {\n override fun iterator(): Iterator<T> {\n val sortedList = this@sorted.toMutableList()\n sortedList.sort()\n return sortedList.iterator()\n }\n }\n}\n\n/**\n * Returns a sequence that yields elements of this sequence sorted according to natural sort order of the value returned by specified [selector] function.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n *\n * The operation is _intermediate_ and _stateful_.\n * \n * @sample samples.collections.Collections.Sorting.sortedBy\n */\npublic inline fun <T, R : Comparable<R>> Sequence<T>.sortedBy(crossinline selector: (T) -> R?): Sequence<T> {\n return sortedWith(compareBy(selector))\n}\n\n/**\n * Returns a sequence that yields elements of this sequence sorted descending according
to natural sort order of the value returned by specified [selector] function.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n *\n * The operation is _intermediate_ and _stateful_.\n */\npublic inline fun <T, R : Comparable<R>> Sequence<T>.sortedByDescending(crossinline selector: (T) -> R?): Sequence<T> {\n return sortedWith(compareByDescending(selector))\n}\n\n/**\n * Returns a sequence that yields elements of this sequence sorted descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n *\n * The operation is _intermediate_ and _stateful_.\n */\npublic fun <T : Comparable<T>> Sequence<T>.sortedDescending(): Sequence<T> {\n return sortedWith(reverseOrder())\n}\n\n/**\n * Returns a sequence that yields elements of this sequence sorted according to the specified [comparator].\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n *\n * The operation is _intermediate_ and _stateful_.\n */\npublic fun <T> Sequence<T>.sortedWith(comparator: Comparator<in T>): Sequence<T> {\n return object : Sequence<T> {\n override fun iterator(): Iterator<T> {\n val sortedList = this@sortedWith.toMutableList()\n sortedList.sortWith(comparator)\n return sortedList.iterator()\n }\n }\n}\n\n/**\n * Returns a [Map] containing key-value pairs provided by [transform] function\n * applied to elements of the given sequence.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original sequence.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.associate\n */\npublic inline fun <T, K, V> Sequence<T>.associate(transform: (T) -> Pair<K,
V>): Map<K, V> {\n return associateTo(LinkedHashMap<K, V>(), transform)\n}\n\n/**\n * Returns a [Map] containing the elements from the given sequence indexed by the key\n * returned from [keySelector] function applied to each element.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original sequence.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.associateBy\n */\npublic inline fun <T, K> Sequence<T>.associateBy(keySelector: (T) -> K): Map<K, T> {\n return associateByTo(LinkedHashMap<K, T>(), keySelector)\n}\n\n/**\n * Returns a [Map] containing the values provided by [valueTransform] and indexed by [keySelector] functions applied to elements of the given sequence.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * The

returned map preserves the entry iteration order of the original sequence.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.associateByWithValueTransform\n */\npublic inline fun <T, K, V> Sequence<T>.associateBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, V> {\n return associateByTo(LinkedHashMap<K, V>(), keySelector, valueTransform)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function applied to each element of the given sequence\n * and value is the element itself.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.associateByTo\n */\npublic inline fun <T, K, M : MutableMap<in K, in T>> Sequence<T>.associateByTo(destination: M, keySelector: (T) -> K): M {\n for (element in this) {\n destination.put(keySelector(element), element)\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and value is provided by the [valueTransform] function applied to elements of the given sequence.\n * \n * If any two elements would have the same key returned by [keySelector] the last one gets added to the map.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.associateByToWithValueTransform\n */\npublic inline fun <T, K, V, M : MutableMap<in K, in V>> Sequence<T>.associateByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {\n for (element in this) {\n destination.put(keySelector(element), valueTransform(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs\n * provided by
[transform] function applied to each element of the given sequence.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.associateTo\n */\npublic inline fun <T, K, V, M : MutableMap<in K, in V>> Sequence<T>.associateTo(destination: M, transform: (T) -> Pair<K, V>): M {\n for (element in this) {\n destination += transform(element)\n }\n return destination\n}\n\n/**\n * Returns a [Map] where keys are elements from the given sequence and values are\n * produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original sequence.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin(\"1.3\")\npublic inline fun <K, V> Sequence<K>.associateWith(valueSelector: (K) -> V): Map<K, V> {\n val result = LinkedHashMap<K, V>()\n return associateWithTo(result, valueSelector)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element of the given sequence,\n * where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.associateWithTo\n */\n@SinceKotlin(\"1.3\")\npublic inline fun <K, V, M : MutableMap<in K, in V>> Sequence<K>.associateWithTo(destination: M, valueSelector: (K) -> V): M {\n for (element in this) {\n destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n/**\n * Appends all elements to the given [destination] collection.\n *\n * The operation is _terminal_.\n
*/\npublic fun <T, C : MutableCollection<in T>> Sequence<T>.toCollection(destination: C): C {\n for (item in this) {\n destination.add(item)\n }\n return destination\n}\n\n/**\n * Returns a new [HashSet] of all elements.\n *\n * The operation is _terminal_.\n */\npublic fun <T> Sequence<T>.toHashSet(): HashSet<T> {\n return toCollection(HashSet<T>())\n}\n\n/**\n * Returns a [List] containing all elements.\n *\n * The operation is _terminal_.\n */\npublic fun <T> Sequence<T>.toList(): List<T> {\n return this.toMutableList().optimizeReadOnlyList()\n}\n\n/**\n * Returns a new [MutableList] filled with all elements of this sequence.\n *\n * The operation is _terminal_.\n */\npublic fun <T> Sequence<T>.toMutableList(): MutableList<T> {\n return toCollection(ArrayList<T>())\n}\n\n/**\n * Returns a [Set] of all elements.\n * \n * The returned set preserves the element iteration order of the original sequence.\n *\n * The operation is _terminal_.\n */\npublic fun <T> Sequence<T>.toSet():

```
Set<T> {\n    return toCollection(LinkedHashSet<T>()).optimizeReadOnlySet()\n}\n\n/**\n * Returns a single sequence of all elements from results of [transform] function being invoked on each element of original sequence.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIterable\")\npublic fun <T, R> Sequence<T>.flatMap(transform: (T) -> Iterable<R>): Sequence<R> {\n    return FlatteningSequence(this, transform, Iterable<R>::iterator)\n}\n\n/**\n * Returns a single sequence of all elements from results of [transform] function being invoked on each element of original sequence.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic fun <T, R> Sequence<T>.flatMap(transform:
(T) -> Sequence<R>): Sequence<R> {\n    return FlatteningSequence(this, transform, Sequence<R>::iterator)\n}\n\n/**\n * Returns a single sequence of all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original sequence.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterable\")\npublic fun <T, R> Sequence<T>.flatMapIndexed(transform: (index: Int, T) -> Iterable<R>): Sequence<R> {\n    return flatMapIndexed(this, transform, Iterable<R>::iterator)\n}\n\n/**\n * Returns a single sequence of all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original sequence.\n *\n * The operation is _intermediate_
and _stateless_.\n * \n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedSequence\")\npublic fun <T, R> Sequence<T>.flatMapIndexed(transform: (index: Int, T) -> Sequence<R>): Sequence<R> {\n    return flatMapIndexed(this, transform, Sequence<R>::iterator)\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original sequence, to the given [destination].\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterableTo\")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.flatMapIndexedTo(destination: C, transform: (index:
Int, T) -> Iterable<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(checkIndexOverflow(index++), element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original sequence, to the given [destination].\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedSequenceTo\")\n@kotlin.internal.InlineOnly\npublic inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.flatMapIndexedTo(destination: C, transform: (index: Int, T) -> Sequence<R>): C {\n    var index = 0\n    for (element in this) {\n        val list = transform(checkIndexOverflow(index++), element)\n        destination.addAll(list)\n    }\n    return destination\n}\n\n/**\n * Appends all elements
yielded from results of [transform] function being invoked on each element of original sequence, to the given [destination].\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIterableTo\")\npublic inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.flatMapTo(destination: C, transform: (T) -> Iterable<R>): C {\n    for
```

(element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original sequence, to the given [destination].\n *\n * The operation is _terminal_.\n */\npublic inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.flatMapTo(destination: C, transform: (T) -> Sequence<R>): C {\n for (element in this) {\n val list = transform(element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Groups elements of the original sequence by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original sequence.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <T, K> Sequence<T>.groupBy(keySelector: (T) -> K): Map<K, List<T>> {\n return groupByTo(LinkedHashMap<K, MutableList<T>>(), keySelector)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original sequence\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned
map preserves the entry iteration order of the keys produced from the original sequence.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <T, K, V> Sequence<T>.groupBy(keySelector: (T) -> K, valueTransform: (T) -> V): Map<K, List<V>> {\n return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups elements of the original sequence by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <T, K, M : MutableMap<in K, MutableList<T>>> Sequence<T>.groupByTo(destination: M, keySelector: (T) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<T>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original sequence\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination] map.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <T, K, V, M : MutableMap<in K, MutableList<V>>> Sequence<T>.groupByTo(destination: M, keySelector: (T) -> K, valueTransform: (T) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Creates a [Grouping]
source from a sequence to be used later with one of group-and-fold operations\n * using the specified [keySelector] function to extract a key from each element.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Grouping.groupingByEachCount\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <T, K> Sequence<T>.groupingBy(crossinline keySelector: (T) -> K): Grouping<T, K> {\n return object : Grouping<T, K> {\n override fun sourceIterator(): Iterator<T> = this@groupingBy.iterator()\n override fun keyOf(element: T): K = keySelector(element)\n }\n}\n\n/**\n * Returns a sequence containing the results of applying the given [transform] function\n * to each element in the original sequence.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Transformations.map\n */\npublic fun <T, R> Sequence<T>.map(transform: (T) -> R): Sequence<R> {\n return TransformingSequence(this,
transform)\n}\n\n/**\n * Returns a sequence containing the results of applying the given [transform] function\n * to each element and its index in the original sequence.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n *\n * The operation is _intermediate_ and _stateless_.\n */\npublic fun <T, R> Sequence<T>.mapIndexed(transform: (index:

Int, T) -> R): Sequence<R> {\n   return TransformingIndexedSequence(this, transform)\n}\n\n/**\n * Returns a sequence containing only the non-null results of applying the given [transform] function\n * to each element and its index in the original sequence.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n *\n * The operation is _intermediate_ and _stateless_.\n */\npublic fun <T, R : Any> Sequence<T>.mapIndexedNotNull(transform: (index:
Int, T) -> R?): Sequence<R> {\n   return TransformingIndexedSequence(this,
transform).filterNotNull()\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original sequence\n * and appends only the non-null results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T, R : Any, C : MutableCollection<in R>> Sequence<T>.mapIndexedNotNullTo(destination: C, transform: (index: Int, T) -> R?): C {\n   forEachIndexed { index, element -> transform(index, element)?.let { destination.add(it) } }\n   return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original sequence\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element
itself\n * and returns the result of the transform applied to the element.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T, R, C : MutableCollection<in R>> Sequence<T>.mapIndexedTo(destination: C, transform: (index: Int, T) -> R): C {\n   var index = 0\n   for (item in this)\n
destination.add(transform(checkIndexOverflow(index++), item))\n   return destination\n}\n\n/**\n * Returns a sequence containing only the non-null results of applying the given [transform] function\n * to each element in the original sequence.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample
samples.collections.Collections.Transformations.mapNotNull\n */\npublic fun <T, R : Any>
Sequence<T>.mapNotNull(transform: (T) -> R?): Sequence<R> {\n   return TransformingSequence(this,
transform).filterNotNull()\n}\n\n/**\n * Applies the given [transform] function to each element in the original sequence\n * and appends only the non-null results to the given [destination].\n
*\n * The operation is _terminal_.\n */\npublic inline fun <T, R : Any, C : MutableCollection<in R>>
Sequence<T>.mapNotNullTo(destination: C, transform: (T) -> R?): C {\n   forEach { element ->
transform(element)?.let { destination.add(it) } }\n   return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original sequence\n * and appends the results to the given [destination].\n *\n * The operation is _terminal_.\n */\npublic inline fun <T, R, C : MutableCollection<in R>>
Sequence<T>.mapTo(destination: C, transform: (T) -> R): C {\n   for (item in this)\n
destination.add(transform(item))\n   return destination\n}\n\n/**\n * Returns a sequence that wraps each element of the original sequence\n * into an [IndexedValue] containing the index of that element and the element itself.\n *\n * The operation is _intermediate_ and _stateless_.\n */\npublic fun <T> Sequence<T>.withIndex():
Sequence<IndexedValue<T>> {\n   return IndexingSequence(this)\n}\n\n/**\n
* Returns a sequence containing only distinct elements from the given sequence.\n * \n * Among equal elements of the given sequence, only the first one will be present in the resulting sequence.\n * The elements in the resulting sequence are in the same order as they were in the source sequence.\n *\n * The operation is _intermediate_ and _stateful_.\n * \n * @sample samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun <T> Sequence<T>.distinct(): Sequence<T> {\n   return this.distinctBy { it }\n}\n\n/**\n * Returns a sequence containing only elements from the given sequence\n * having distinct keys returned by the given [selector] function.\n * \n * Among elements of the given sequence with equal keys, only the first one will be present in the resulting sequence.\n * The elements in the resulting sequence are in the same order as they were in the source sequence.\n *\n * The operation is _intermediate_ and _stateful_.\n * \n * @sample
samples.collections.Collections.Transformations.distinctAndDistinctBy\n */\npublic fun <T, K>
Sequence<T>.distinctBy(selector: (T) -> K): Sequence<T> {\n   return DistinctSequence(this, selector)\n}\n\n/**\n * Returns a new [MutableSet] containing all distinct elements from the given sequence.\n * \n * The returned set preserves the element iteration order of the original sequence.\n *\n * The operation is _terminal_.\n */\npublic fun

<T> Sequence<T>.toMutableSet(): MutableSet<T> {\n   val set = LinkedHashSet<T>()\n   for (item in this) set.add(item)\n   return set\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the sequence contains no elements, the function returns `true`\n * because there are no elements in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in [\"Vacuous truth\"](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun <T> Sequence<T>.all(predicate: (T) -> Boolean): Boolean {\n   for (element in this) if (!predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if sequence has at least one element.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\npublic fun <T> Sequence<T>.any(): Boolean {\n   return iterator().hasNext()\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun <T> Sequence<T>.any(predicate: (T) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return true\n   return false\n}\n\n/**\n * Returns the number of elements in this sequence.\n *\n * The operation is _terminal_.\n */\npublic fun <T> Sequence<T>.count(): Int {\n   var count = 0\n   for (element in this) checkCountOverflow(++count)\n   return count\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n *\n * The operation is _terminal_.\n */\npublic inline fun <T> Sequence<T>.count(predicate: (T) -> Boolean): Int {\n   var count = 0\n   for (element in this) if (predicate(element)) checkCountOverflow(++count)\n   return count\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the sequence is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T, R> Sequence<T>.fold(initial: R, operation: (acc: R, T) -> R): R {\n   var accumulator = initial\n   for (element in this) accumulator = operation(accumulator, element)\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original sequence.\n * \n * Returns the specified [initial] value if the sequence is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T, R> Sequence<T>.foldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): R {\n   var index = 0\n   var accumulator = initial\n   for (element in this) accumulator = operation(checkIndexOverflow(index++), accumulator, element)\n   return accumulator\n}\n\n/**\n * Performs the given [action] on each element.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T> Sequence<T>.forEach(action: (T) -> Unit): Unit {\n   for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n *\n * The operation is _terminal_.\n */\npublic inline fun <T> Sequence<T>.forEachIndexed(action: (index: Int, T) -> Unit): Unit {\n   var index = 0\n   for (item in this) action(checkIndexOverflow(index++), item)\n}\n\n/**\n * Returns the largest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun Sequence<Double>.max(): Double {\n   val iterator = iterator()\n   if (!iterator.hasNext()) throw NoSuchElementException()\n   var max = iterator.next()\n   while (iterator.hasNext()) {\n   val e = iterator.next()\n   max = maxOf(max, e)\n   }\n   return max\n}\n\n/**\n * Returns the largest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun Sequence<Float>.max(): Float {\n   val iterator = iterator()\n   if (!iterator.hasNext()) throw NoSuchElementException()\n   var max = iterator.next()\n   while (iterator.hasNext()) {\n   val e = iterator.next()\n   max = maxOf(max, e)\n   }\n   return max\n}\n\n/**\n * Returns the largest element.\n *\n *

The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun <T : Comparable<T>> Sequence<T>.max(): T {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var max = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <T, R : Comparable<R>> Sequence<T>.maxBy(selector: (T) -> R): T {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var maxElem = iterator.next()\n if (!iterator.hasNext()) return maxElem\n var maxValue = selector(maxElem)\n do {\n val e = iterator.next()\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n } while (iterator.hasNext())\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <T, R : Comparable<R>> Sequence<T>.maxByOrNull(selector: (T) -> R): T? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var maxElem = iterator.next()\n if (!iterator.hasNext()) return maxElem\n var maxValue = selector(maxElem)\n do {\n val e = iterator.next()\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n } while (iterator.hasNext())\n return maxElem\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the sequence.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.maxOf(selector: (T) -> Double): Double {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var maxValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the sequence.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic\n inline fun <T> Sequence<T>.maxOf(selector: (T) -> Float): Float {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var maxValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the sequence.\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Sequence<T>.maxOf(selector: (T) -> R): R {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var maxValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n * \n * If any of values

produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.maxOfOrNull(selector: (T) -> Double): Double? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var maxValue = selector(iterator.next())\n   while (iterator.hasNext()) {\n   val v = selector(iterator.next())\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.maxOfOrNull(selector: (T) -> Float): Float? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var maxValue = selector(iterator.next())\n   while (iterator.hasNext()) {\n   val v = selector(iterator.next())\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Sequence<T>.maxOfOrNull(selector: (T) -> R): R? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var maxValue = selector(iterator.next())\n   while (iterator.hasNext()) {\n   val v = selector(iterator.next())\n   if (maxValue < v) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the sequence.\n * \n * @throws NoSuchElementException if the sequence is empty.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Sequence<T>.maxOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n   val iterator = iterator()\n   if (!iterator.hasNext()) throw NoSuchElementException()\n   var maxValue = selector(iterator.next())\n   while (iterator.hasNext()) {\n   val v = selector(iterator.next())\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the sequence or `null` if there are no elements.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R> Sequence<T>.maxOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var maxValue = selector(iterator.next())\n   while (iterator.hasNext()) {\n   val v = selector(iterator.next())\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\npublic fun Sequence<Double>.maxOrNull(): Double? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var max = iterator.next()\n   while (iterator.hasNext()) {\n   val e = iterator.next()\n   max = maxOf(max, e)\n   }\n   return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\npublic fun Sequence<Float>.maxOrNull(): Float? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var max = iterator.next()\n   while (iterator.hasNext()) {\n   val e = iterator.next()\n   max = maxOf(max, e)\n   }\n   return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n *\n * The operation

is _terminal_.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T : Comparable<T>> Sequence<T>.maxOrNull(): T? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var max = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic
fun <T> Sequence<T>.maxWith(comparator: Comparator<in T>): T {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var max = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T> Sequence<T>.maxWithOrNull(comparator: Comparator<in T>): T? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var max = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n *\n * The operation
is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun Sequence<Double>.min(): Double {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n * \n * If any of elements is `NaN` returns `NaN`.\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun Sequence<Float>.min(): Float {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n
min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element.\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun <T : Comparable<T>> Sequence<T>.min(): T {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n * \n * @sample samples.collections.Collections.Aggregates.minBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <T, R :
Comparable<R>> Sequence<T>.minBy(selector: (T) -> R): T {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var minElem = iterator.next()\n if (!iterator.hasNext()) return minElem\n var minValue = selector(minElem)\n do {\n val e = iterator.next()\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n } while (iterator.hasNext())\n return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <T, R :
Comparable<R>> Sequence<T>.minByOrNull(selector: (T) -> R): T? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var minElem = iterator.next()\n if (!iterator.hasNext()) return minElem\n var

minValue = selector(minElem)\n   do {\n   val e = iterator.next()\n   val v = selector(e)\n   if (minValue > v) {\n   minElem = e\n   minValue = v\n   }\n   } while (iterator.hasNext())\n   return minElem\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the sequence.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOf(selector: (T) -> Double): Double {\n   val iterator = iterator()\n   if (!iterator.hasNext()) throw NoSuchElementException()\n   var minValue = selector(iterator.next())\n   while (iterator.hasNext()) {\n   val v = selector(iterator.next())\n   minValue = minOf(minValue, v)\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the sequence.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOf(selector: (T) -> Float): Float {\n   val iterator = iterator()\n   if (!iterator.hasNext()) throw NoSuchElementException()\n   var minValue = selector(iterator.next())\n   while (iterator.hasNext()) {\n   val v = selector(iterator.next())\n   minValue = minOf(minValue, v)\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the sequence.\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Sequence<T>.minOf(selector: (T) -> R): R {\n   val iterator = iterator()\n   if (!iterator.hasNext()) throw NoSuchElementException()\n   var minValue = selector(iterator.next())\n   while (iterator.hasNext()) {\n   val v = selector(iterator.next())\n   if (minValue > v) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOfOrNull(selector: (T) -> Double): Double? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var minValue = selector(iterator.next())\n   while (iterator.hasNext()) {\n   val v = selector(iterator.next())\n   minValue = minOf(minValue, v)\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minOfOrNull(selector: (T) -> Float): Float? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var minValue = selector(iterator.next())\n   while (iterator.hasNext()) {\n   val v = selector(iterator.next())\n   minValue = minOf(minValue, v)\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the sequence or `null` if there are no elements.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R : Comparable<R>> Sequence<T>.minOfOrNull(selector: (T) -> R): R? {\n   val iterator = iterator()\n   if (!iterator.hasNext()) return null\n   var minValue = selector(iterator.next())\n   while (iterator.hasNext()) {\n   val v =

selector(iterator.next())\n
if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the sequence.\n * \n * @throws NoSuchElementException if the sequence is empty.\n *\n * The operation is _terminal_.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>
Sequence<T>.minOfWith(comparator: Comparator<in R>, selector: (T) -> R): R {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest
value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the sequence or `null` if there are no elements.\n *\n * The operation is _terminal_.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <T, R>
Sequence<T>.minOfWithOrNull(comparator: Comparator<in R>, selector: (T) -> R): R? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var minValue = selector(iterator.next())\n while (iterator.hasNext()) {\n val v = selector(iterator.next())\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n *\n * The operation is _terminal_.\n
*/\n@SinceKotlin(\"1.4\")\npublic
fun Sequence<Double>.minOrNull(): Double? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n * \n * If any of elements is `NaN` returns `NaN`.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\npublic fun Sequence<Float>.minOrNull(): Float? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n min = minOf(min, e)\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T : Comparable<T>> Sequence<T>.minOrNull(): T? {\n val iterator = iterator()\n if (!iterator.hasNext())
return null\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n *\n * The operation is _terminal_.\n * \n * @throws NoSuchElementException if the sequence is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun <T> Sequence<T>.minWith(comparator: Comparator<in T>): T {\n val iterator = iterator()\n if (!iterator.hasNext()) throw NoSuchElementException()\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n *\n * The operation is _terminal_.\n
*/\n@SinceKotlin(\"1.4\")\npublic fun <T> Sequence<T>.minWithOrNull(comparator: Comparator<in T>): T? {\n val iterator = iterator()\n if (!iterator.hasNext()) return null\n var min = iterator.next()\n while (iterator.hasNext()) {\n val e = iterator.next()\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns `true` if the sequence has no elements.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\npublic fun <T> Sequence<T>.none(): Boolean {\n return !iterator().hasNext()\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun <T> Sequence<T>.none(predicate: (T) -> Boolean): Boolean {\n for (element in this) if

(predicate(element)) return false\n   return true\n}\n\n/**\n * Returns a sequence which performs the given [action] on each element of the original sequence as they pass through it.\n *\n * The operation is _intermediate_ and _stateless_.\n */\n@SinceKotlin(\"1.1\")\npublic fun <T> Sequence<T>.onEach(action: (T) -> Unit): Sequence<T> {\n   return map {\n   action(it)\n   it\n   }\n}\n\n/**\n * Returns a sequence which performs the given [action] on each element of the original sequence as they pass through it.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n *\n * The operation is _intermediate_ and _stateless_.\n */\n@SinceKotlin(\"1.4\")\npublic fun <T> Sequence<T>.onEachIndexed(action: (index: Int, T) -> Unit): Sequence<T> {\n   return mapIndexed { index, element ->\n   action(index, element)\n   element\n   }\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this sequence is empty. If the sequence can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun <S, T : S> Sequence<T>.reduce(operation: (acc: S, T) -> S): S {\n   val iterator = this.iterator()\n   if (!iterator.hasNext()) throw UnsupportedOperationException(\"Empty sequence can't be reduced.\")\n   var accumulator: S = iterator.next()\n   while (iterator.hasNext()) {\n   accumulator = operation(accumulator, iterator.next())\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original sequence.\n * \n * Throws an exception if this sequence is empty. If the sequence can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun <S, T : S> Sequence<T>.reduceIndexed(operation: (index: Int, acc: S, T) -> S): S {\n   val iterator = this.iterator()\n   if (!iterator.hasNext()) throw UnsupportedOperationException(\"Empty sequence can't be reduced.\")\n   var index = 1\n   var accumulator: S = iterator.next()\n   while (iterator.hasNext()) {\n   accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original sequence.\n * \n * Returns `null` if the sequence is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <S, T : S> Sequence<T>.reduceIndexedOrNull(operation: (index: Int, acc: S, T) -> S): S? {\n   val iterator = this.iterator()\n   if (!iterator.hasNext()) return null\n   var index = 1\n   var accumulator: S = iterator.next()\n   while (iterator.hasNext()) {\n   accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the sequence is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <S, T : S> Sequence<T>.reduceOrNull(operation: (acc: S, T) -> S): S? {\n   val iterator = this.iterator()\n   if (!iterator.hasNext()) return null\n   var accumulator: S = iterator.next()\n   while (iterator.hasNext()) {\n   accumulator = operation(accumulator, iterator.next())\n   }\n   return accumulator\n}\n\n/**\n * Returns a sequence containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n

* \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * The [initial] value should also be immutable (or should not be mutated)\n * as it may be passed to [operation] function later because of sequence's lazy nature.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\npublic fun <T, R> Sequence<T>.runningFold(initial: R, operation: (acc: R, T) -> R): Sequence<R> {\n return sequence {\n yield(initial)\n var accumulator = initial\n for (element in this@runningFold) {\n accumulator = operation(accumulator, element)\n yield(accumulator)\n }\n }\n}\n\n/**\n * Returns a sequence containing successive
accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original sequence and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * The [initial] value should also be immutable (or should not be mutated)\n * as it may be passed to [operation] function later because of sequence's lazy nature.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\npublic fun <T, R> Sequence<T>.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): Sequence<R> {\n return sequence {\n
yield(initial)\n var index = 0\n var accumulator = initial\n for (element in this@runningFoldIndexed) {\n accumulator = operation(checkIndexOverflow(index++), accumulator, element)\n yield(accumulator)\n }\n }\n}\n\n/**\n * Returns a sequence containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this sequence.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * \n * @param [operation] function that takes current accumulator value and the element, and calculates the next accumulator value.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
fun <S, T : S> Sequence<T>.runningReduce(operation: (acc: S, T) -> S): Sequence<S> {\n return sequence {\n val iterator = iterator()\n if (iterator.hasNext()) {\n var accumulator: S = iterator.next()\n yield(accumulator)\n while (iterator.hasNext()) {\n accumulator = operation(accumulator, iterator.next())\n yield(accumulator)\n }\n }\n}\n\n/**\n * Returns a sequence containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original sequence and current accumulator value that starts with the first element of this sequence.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates
the next accumulator value.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\npublic fun <S, T : S> Sequence<T>.runningReduceIndexed(operation: (index: Int, acc: S, T) -> S): Sequence<S> {\n return sequence {\n val iterator = iterator()\n if (iterator.hasNext()) {\n var accumulator: S = iterator.next()\n yield(accumulator)\n var index = 1\n while (iterator.hasNext()) {\n accumulator = operation(checkIndexOverflow(index++), accumulator, iterator.next())\n yield(accumulator)\n }\n }\n}\n\n/**\n * Returns a sequence containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should

not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * The [initial] value should also be immutable (or should not be mutated)\n * as it may be passed to [operation] function later because of sequence's lazy nature.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T, R> Sequence<T>.scan(initial: R, operation: (acc: R, T) -> R): Sequence<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a sequence containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original sequence and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value
passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting sequence.\n * The [initial] value should also be immutable (or should not be mutated)\n * as it may be passed to [operation] function later because of sequence's lazy nature.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T, R> Sequence<T>.scanIndexed(initial: R, operation: (index: Int, acc: R, T) -> R): Sequence<R> {\n return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the sequence.\n *\n * The operation is _terminal_.\n */\n@Deprecated(\"Use sumOf
instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun <T> Sequence<T>.sumBy(selector: (T) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the sequence.\n *\n * The operation is _terminal_.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun <T> Sequence<T>.sumByDouble(selector: (T) -> Double): Double {\n var sum: Double = 0.0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the sequence.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfDouble\")\n@kotlin.internal.InlineOnly\npublic
inline fun <T> Sequence<T>.sumOf(selector: (T) -> Double): Double {\n var sum: Double = 0.toDouble()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the sequence.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfInt\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.sumOf(selector: (T) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the sequence.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfLong\")\n@kotlin.internal.InlineOnly\npublic
inline fun <T> Sequence<T>.sumOf(selector: (T) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the sequence.\n *\n * The operation is _terminal_.\n */\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.sumOf(selector: (T) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return

sum\n}\n\n/**\n * Returns the sum of
all values produced by [selector] function applied to each element in the sequence.\n *\n * The operation is _terminal_.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.sumOf(selector: (T) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns an original collection containing all the non-`null` elements, throwing an [IllegalArgumentException] if there are any `null` elements.\n *\n * The operation is _intermediate_ and _stateless_.\n */\npublic fun <T : Any> Sequence<T?>.requireNoNulls(): Sequence<T> {\n return map { it ?: throw IllegalArgumentException(\"null element found in $this.\") }\n}\n\n/**\n * Splits this sequence into a sequence of lists
each not exceeding the given [size].\n * \n * The last list in the resulting sequence may have fewer elements than the given [size].\n * \n * @param size the number of elements to take in each list, must be positive and can be greater than the number of elements in this sequence.\n *\n * The operation is _intermediate_ and _stateful_.\n * \n * @sample samples.collections.Collections.Transformations.chunked\n */\n@SinceKotlin(\"1.2\")\npublic fun <T> Sequence<T>.chunked(size: Int): Sequence<List<T>> {\n return windowed(size, size, partialWindows = true)\n}\n\n/**\n * Splits this sequence into several lists each not exceeding the given [size]\n * and applies the given [transform] function to an each.\n * \n * @return sequence of results of the [transform] applied to an each list.\n * \n * Note that the list passed to the [transform] function is ephemeral and is valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a snapshot
of it.\n * The last list may have fewer elements than the given [size].\n * \n * @param size the number of elements to take in each list, must be positive and can be greater than the number of elements in this sequence.\n *\n * The operation is _intermediate_ and _stateful_.\n * \n * @sample samples.text.Strings.chunkedTransform\n */\n@SinceKotlin(\"1.2\")\npublic fun <T, R> Sequence<T>.chunked(size: Int, transform: (List<T>) -> R): Sequence<R> {\n return windowed(size, size, partialWindows = true, transform = transform)\n}\n\n/**\n * Returns a sequence containing all elements of the original sequence without the first occurrence of the given [element].\n *\n * The operation is _intermediate_ and _stateless_.\n */\npublic operator fun <T> Sequence<T>.minus(element: T): Sequence<T> {\n return object: Sequence<T> {\n override fun iterator(): Iterator<T> {\n var removed = false\n return this@minus.filter { if (!removed && it == element) { removed = true;
false } else true }.iterator()\n }\n }\n}\n\n/**\n * Returns a sequence containing all elements of original sequence except the elements contained in the given [elements] array.\n * \n * Note that the source sequence and the array being subtracted are iterated only when an `iterator` is requested from\n * the resulting sequence. Changing any of them between successive calls to `iterator` may affect the result.\n *\n * The operation is _intermediate_ and _stateful_.\n */\npublic operator fun <T> Sequence<T>.minus(elements: Array<out T>): Sequence<T> {\n if (elements.isEmpty()) return this\n return object: Sequence<T> {\n override fun iterator(): Iterator<T> {\n return this@minus.filterNot { it in elements }.iterator()\n }\n }\n}\n\n/**\n * Returns a sequence containing all elements of original sequence except the elements contained in the given [elements] collection.\n * \n * Note that the source sequence and the collection being subtracted are
iterated only when an `iterator` is requested from\n * the resulting sequence. Changing any of them between successive calls to `iterator` may affect the result.\n *\n * The operation is _intermediate_ and _stateful_.\n */\npublic operator fun <T> Sequence<T>.minus(elements: Iterable<T>): Sequence<T> {\n return object: Sequence<T> {\n override fun iterator(): Iterator<T> {\n val other = elements.convertToListIfNotCollection()\n if (other.isEmpty())\n return this@minus.iterator()\n else\n return this@minus.filterNot { it in other }.iterator()\n }\n }\n}\n\n/**\n * Returns a sequence containing all elements of original sequence except the elements contained in the given [elements] sequence.\n * \n * Note that the source sequence and the sequence being subtracted are iterated only when an `iterator` is requested

from\n * the resulting sequence. Changing any of them between successive calls to `iterator`
may affect the result.\n * \n * The operation is _intermediate_ for this sequence and _terminal_ and _stateful_ for the [elements] sequence.\n */\npublic operator fun <T> Sequence<T>.minus(elements: Sequence<T>): Sequence<T> {\n return object: Sequence<T> {\n override fun iterator(): Iterator<T> {\n val other = elements.toList()\n if (other.isEmpty())\n return this@minus.iterator()\n else\n return this@minus.filterNot { it in other }.iterator()\n }\n }\n}\n\n/**\n * Returns a sequence containing all elements of the original sequence without the first occurrence of the given [element].\n *\n * The operation is _intermediate_ and _stateless_.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.minusElement(element: T): Sequence<T> {\n return minus(element)\n}\n\n/**\n * Splits the original sequence into pair of lists,\n * where *first* list contains elements for which [predicate] yielded
`true`,\n * while *second* list contains elements for which [predicate] yielded `false`.\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Sequences.Transformations.partition\n */\npublic inline fun <T> Sequence<T>.partition(predicate: (T) -> Boolean): Pair<List<T>, List<T>> {\n val first = ArrayList<T>()\n val second = ArrayList<T>()\n for (element in this) {\n if (predicate(element)) {\n first.add(element)\n } else {\n second.add(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Returns a sequence containing all elements of the original sequence and then the given [element].\n *\n * The operation is _intermediate_ and _stateless_.\n */\npublic operator fun <T> Sequence<T>.plus(element: T): Sequence<T> {\n return sequenceOf(this, sequenceOf(element)).flatten()\n}\n\n/**\n * Returns a sequence containing all elements of original sequence and then all elements of the given [elements] array.\n
* \n * Note that the source sequence and the array being added are iterated only when an `iterator` is requested from\n * the resulting sequence. Changing any of them between successive calls to `iterator` may affect the result.\n *\n * The operation is _intermediate_ and _stateless_.\n */\npublic operator fun <T> Sequence<T>.plus(elements: Array<out T>): Sequence<T> {\n return this.plus(elements.asList())\n}\n\n/**\n * Returns a sequence containing all elements of original sequence and then all elements of the given [elements] collection.\n * \n * Note that the source sequence and the collection being added are iterated only when an `iterator` is requested from\n * the resulting sequence. Changing any of them between successive calls to `iterator` may affect the result.\n *\n * The operation is _intermediate_ and _stateless_.\n */\npublic operator fun <T> Sequence<T>.plus(elements: Iterable<T>): Sequence<T> {\n return sequenceOf(this, elements.asSequence()).flatten()\n}\n\n/**\n
* Returns a sequence containing all elements of original sequence and then all elements of the given [elements] sequence.\n * \n * Note that the source sequence and the sequence being added are iterated only when an `iterator` is requested from\n * the resulting sequence. Changing any of them between successive calls to `iterator` may affect the result.\n *\n * The operation is _intermediate_ and _stateless_.\n */\npublic operator fun <T> Sequence<T>.plus(elements: Sequence<T>): Sequence<T> {\n return sequenceOf(this, elements).flatten()\n}\n\n/**\n * Returns a sequence containing all elements of the original sequence and then the given [element].\n *\n * The operation is _intermediate_ and _stateless_.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.plusElement(element: T): Sequence<T> {\n return plus(element)\n}\n\n/**\n * Returns a sequence of snapshots of the window of the given [size]\n * sliding along this sequence with the given [step], where each\n
* snapshot is a list.\n * \n * Several last lists may have fewer elements than the given [size].\n * \n * Both [size] and [step] must be positive and can be greater than the number of elements in this sequence.\n * @param size the number of elements to take in each window\n * @param step the number of elements to move the window forward by on an each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the end if any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample samples.collections.Sequences.Transformations.takeWindows\n */\n@SinceKotlin(\"1.2\")\npublic fun <T> Sequence<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false): Sequence<List<T>> {\n return windowedSequence(size, step, partialWindows, reuseBuffer = false)\n}\n\n/**\n * Returns a sequence of results of applying the given [transform] function to\n * an each list representing a view over the window of the given [size]\n *

sliding along this sequence with the given [step].\n * \n * Note that the list passed to the [transform] function is ephemeral and is valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a snapshot of it.\n * Several last lists may have fewer elements than the given [size].\n * \n * Both [size] and [step] must be positive and can be greater than the number of elements in this sequence.\n * @param size the number of elements to take in each window\n * @param step the number of elements to move the window forward by on an each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the end if any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample samples.collections.Sequences.Transformations.averageWindows\n */\n@SinceKotlin(\"1.2\")\npublic fun <T, R> Sequence<T>.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform: (List<T>) -> R): Sequence<R> {\n return windowedSequence(size, step, partialWindows, reuseBuffer = true).map(transform)\n}\n\n/**\n * Returns a sequence of values built from the elements of `this` sequence and the [other] sequence with the same index.\n * The resulting sequence ends as soon as the shortest input sequence ends.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Sequences.Transformations.zip\n */\npublic infix fun <T, R> Sequence<T>.zip(other: Sequence<R>): Sequence<Pair<T, R>> {\n return MergingSequence(this, other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a sequence of values built from the elements of `this` sequence and the [other] sequence with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The resulting sequence ends as soon as the shortest input sequence ends.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Sequences.Transformations.zipWithTransform\n */\npublic fun <T, R, V> Sequence<T>.zip(other: Sequence<R>, transform: (a: T, b: R) -> V): Sequence<V> {\n return MergingSequence(this, other, transform)\n}\n\n/**\n * Returns a sequence of pairs of each two adjacent elements in this sequence.\n * \n * The returned sequence is empty if this sequence contains less than two elements.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Transformations.zipWithNext\n */\n@SinceKotlin(\"1.2\")\npublic fun <T> Sequence<T>.zipWithNext(): Sequence<Pair<T, T>> {\n return zipWithNext { a, b -> a to b }\n}\n\n/**\n * Returns a sequence containing the results of applying the given [transform] function\n * to an each pair of two adjacent elements in this sequence.\n * \n * The returned sequence is empty if this sequence contains less than two elements.\n *\n * The operation is _intermediate_ and _stateless_.\n * \n * @sample samples.collections.Collections.Transformations.zipWithNextToFindDeltas\n */\n@SinceKotlin(\"1.2\")\npublic fun <T, R> Sequence<T>.zipWithNext(transform: (a: T, b: T) -> R): Sequence<R> {\n return sequence result@ {\n val iterator = iterator()\n if (!iterator.hasNext()) return@result\n var current = iterator.next()\n while (iterator.hasNext()) {\n val next = iterator.next()\n yield(transform(current, next))\n current = next\n }\n }\n}\n\n/**\n * Appends the string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.joinTo\n */\npublic fun <T, A : Appendable> Sequence<T>.joinTo(buffer: A, separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((T) -> CharSequence)? = null): A {\n buffer.append(prefix)\n var count = 0\n for (element in this) {\n if (++count > 1) buffer.append(separator)\n if (limit < 0 || count <= limit) {\n buffer.appendElement(element, transform)\n } else break\n }\n if (limit >= 0 && count > limit) buffer.append(truncated)\n buffer.append(postfix)\n return buffer\n}\n\n/**\n * Creates a string from all the elements separated using [separator] and using the given [prefix] and [postfix] if supplied.\n * \n * If the collection could be huge, you can specify a non-negative value of [limit], in which case only the first [limit]\n * elements will be appended, followed by the [truncated] string (which defaults to \"...\").\n *\n * The operation is _terminal_.\n * \n * @sample samples.collections.Collections.Transformations.joinToString\n

```
*/\npublic fun <T> Sequence<T>.joinToString(separator: CharSequence = \", \", prefix: CharSequence = \"\", postfix: CharSequence = \"\", limit: Int = -1, truncated: CharSequence = \"...\", transform: ((T) -> CharSequence)? = null): String {\n    return joinTo(StringBuilder(), separator, prefix, postfix, limit, truncated, transform).toString()\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original sequence returning its elements when being iterated.\n */\npublic fun <T> Sequence<T>.asIterable(): Iterable<T> {\n    return Iterable { this.iterator() }\n}\n\n/**\n * Returns this sequence as a [Sequence].\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Sequence<T>.asSequence(): Sequence<T> {\n    return this\n}\n\n/**\n * Returns an average value of elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName(\"averageOfByte\")\npublic fun Sequence<Byte>.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName(\"averageOfShort\")\npublic fun Sequence<Short>.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName(\"averageOfInt\")\npublic fun Sequence<Int>.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName(\"averageOfLong\")\npublic fun Sequence<Long>.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName(\"averageOfFloat\")\npublic fun Sequence<Float>.average(): Double {\n    var sum: Double = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns an average value of elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName(\"averageOfDouble\")\npublic fun Sequence<Double>.average(): Double {\n    var sum: Double\n    = 0.0\n    var count: Int = 0\n    for (element in this) {\n        sum += element\n        checkCountOverflow(++count)\n    }\n    return if (count == 0) Double.NaN else sum / count\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName(\"sumOfByte\")\npublic fun Sequence<Byte>.sum(): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName(\"sumOfShort\")\npublic fun Sequence<Short>.sum(): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName(\"sumOfInt\")\npublic fun Sequence<Int>.sum(): Int {\n    var sum: Int = 0\n    for (element in this) {\n        sum += element\n\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName(\"sumOfLong\")\npublic fun Sequence<Long>.sum(): Long {\n    var sum: Long = 0L\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName(\"sumOfFloat\")\npublic fun Sequence<Float>.sum(): Float {\n    var sum: Float = 0.0f\n    for (element in this) {\n        sum += element\n    }\n    return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName(\"sumOfDouble\")\npublic fun Sequence<Double>.sum(): Double {\n    var sum: Double = 0.0\n    for (element in this) {\n        sum += element\n
```

}\n   return sum\n}\n\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this
 source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"SetsKt\")\n\npackage kotlin.collections\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.random.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns a set containing all elements of the original set except the given [element].\n * \n * The returned set preserves the element iteration order of the original set.\n */\npublic operator fun <T> Set<T>.minus(element: T): Set<T> {\n   val result = LinkedHashSet<T>(mapCapacity(size))\n   var removed = false\n   return this.filterTo(result) { if (!removed && it == element) { removed = true; false } else true }\n}\n\n/**\n * Returns a set containing all elements of the original set except the elements contained in the given [elements]
 array.\n * \n * The returned set preserves the element iteration order of the original set.\n */\npublic operator fun <T> Set<T>.minus(elements: Array<out T>): Set<T> {\n   val result = LinkedHashSet<T>(this)\n   result.removeAll(elements)\n   return result\n}\n\n/**\n * Returns a set containing all elements of the original set except the elements contained in the given [elements] collection.\n * \n * The returned set preserves the element iteration order of the original set.\n */\npublic operator fun <T> Set<T>.minus(elements: Iterable<T>): Set<T> {\n   val other = elements.convertToListIfNotCollection()\n   if (other.isEmpty())\n   return this.toSet()\n   if (other is Set)\n   return this.filterNotTo(LinkedHashSet<T>()) { it in other }\n   val result = LinkedHashSet<T>(this)\n   result.removeAll(other)\n   return result\n}\n\n/**\n * Returns a set containing all elements of the original set except the elements contained in the given [elements] sequence.\n
 * \n * The returned set preserves the element iteration order of the original set.\n */\npublic operator fun <T> Set<T>.minus(elements: Sequence<T>): Set<T> {\n   val result = LinkedHashSet<T>(this)\n   result.removeAll(elements)\n   return result\n}\n\n/**\n * Returns a set containing all elements of the original set except the given [element].\n * \n * The returned set preserves the element iteration order of the original set.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Set<T>.minusElement(element: T): Set<T> {\n   return minus(element)\n}\n\n/**\n * Returns a set containing all elements of the original set and then the given [element] if it isn't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n */\npublic operator fun <T> Set<T>.plus(element: T): Set<T> {\n   val result = LinkedHashSet<T>(mapCapacity(size + 1))\n   result.addAll(this)\n   result.add(element)\n   return result\n}\n\n/**\n * Returns a set
 containing all elements of the original set and the given [elements] array,\n * which aren't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n */\npublic operator fun <T> Set<T>.plus(elements: Array<out T>): Set<T> {\n   val result = LinkedHashSet<T>(mapCapacity(this.size + elements.size))\n   result.addAll(this)\n   result.addAll(elements)\n   return result\n}\n\n/**\n * Returns a set containing all elements of the original set and the given [elements] collection,\n * which aren't already in this set.\n * The returned set preserves the element iteration order of the original set.\n */\npublic operator fun <T> Set<T>.plus(elements: Iterable<T>): Set<T> {\n   val result = LinkedHashSet<T>(mapCapacity(elements.collectionSizeOrNull()?.let { this.size + it } ?: this.size * 2))\n   result.addAll(this)\n   result.addAll(elements)\n   return result\n}\n\n/**\n * Returns a set containing all elements of the original set and the
 given [elements] sequence,\n * which aren't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n */\npublic operator fun <T> Set<T>.plus(elements: Sequence<T>): Set<T> {\n   val result = LinkedHashSet<T>(mapCapacity(this.size * 2))\n   result.addAll(this)\n   result.addAll(elements)\n   return result\n}\n\n/**\n * Returns a set containing all elements of the original set and then the given [element] if it isn't already in this set.\n * \n * The returned set preserves the element iteration order of the original set.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Set<T>.plusElement(element: T): Set<T> {\n   return plus(element)\n}\n\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language

contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"StringsKt\")\n\npackage kotlin.text\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.random.*\n\n/**\n * Returns a character at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this char sequence.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic expect fun CharSequence.elementAt(index: Int): Char\n\n/**\n * Returns a character at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this char sequence.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrElse\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.elementAtOrElse(index: Int, defaultValue: (Int) -> Char): Char {\n   return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns a character at the given [index] or `null` if the [index] is out of bounds of this char sequence.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.elementAtOrNull(index: Int): Char? {\n   return this.getOrNull(index)\n}\n\n/**\n * Returns the first character matching the given [predicate], or `null` if no such character was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.find(predicate: (Char) -> Boolean): Char? {\n   return firstOrNull(predicate)\n}\n\n/**\n * Returns the last character matching the given [predicate], or `null` if no such character was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.findLast(predicate: (Char) -> Boolean): Char? {\n   return lastOrNull(predicate)\n}\n\n/**\n * Returns the first character.\n * \n * @throws NoSuchElementException if the char sequence is empty.\n */\npublic fun CharSequence.first(): Char {\n   if (isEmpty())\n   throw NoSuchElementException(\"Char sequence is empty.\")\n   return this[0]\n}\n\n/**\n * Returns the first character matching the given [predicate].\n * @throws [NoSuchElementException] if no such character is found.\n */\npublic inline fun CharSequence.first(predicate: (Char) -> Boolean): Char {\n   for (element in this) if (predicate(element)) return element\n   throw NoSuchElementException(\"Char sequence contains no character matching the predicate.\")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to characters of this char sequence in iteration order,\n * or throws [NoSuchElementException] if no non-null value was produced.\n * \n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic inline fun <R : Any> CharSequence.firstNotNullOf(transform: (Char) -> R?): R {\n   return firstNotNullOfOrNull(transform) ?: throw NoSuchElementException(\"No element of the char sequence was transformed to a non-null value.\")\n}\n\n/**\n * Returns the first non-null value produced by [transform] function being applied to characters of this char sequence in iteration order,\n * or `null` if no non-null value was produced.\n * \n * @sample samples.collections.Collections.Transformations.firstNotNullOf\n */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\npublic inline fun <R : Any> CharSequence.firstNotNullOfOrNull(transform: (Char) -> R?): R? {\n   for (element in this) {\n   val result = transform(element)\n   if (result != null) {\n   return result\n   }\n   }\n   return null\n}\n\n/**\n * Returns the first character, or `null` if the char sequence is empty.\n */\npublic fun CharSequence.firstOrNull(): Char? {\n   return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first character matching the given [predicate], or `null` if character was not found.\n */\npublic inline fun CharSequence.firstOrNull(predicate: (Char) -> Boolean): Char? {\n   for (element in this) if (predicate(element)) return element\n   return null\n}\n\n/**\n * Returns a character at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this char sequence.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.getOrElse(index: Int, defaultValue: (Int) -> Char): Char {\n   return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns a character at the given [index] or `null` if the [index] is out of bounds of this char sequence.\n * \n * @sample

samples.collections.Collections.Elements.getOrNull\n */\npublic fun CharSequence.getOrNull(index: Int): Char? {\n   return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns index of the first character matching the given [predicate],
or -1 if the char sequence does not contain such character.\n */\npublic inline fun CharSequence.indexOfFirst(predicate: (Char) -> Boolean): Int {\n   for (index in indices) {\n   if (predicate(this[index])) {\n   return index\n   }\n   }\n   return -1\n}\n\n/**\n * Returns index of the last character matching the given [predicate], or -1 if the char sequence does not contain such character.\n */\npublic inline fun CharSequence.indexOfLast(predicate: (Char) -> Boolean): Int {\n   for (index in indices.reversed()) {\n   if (predicate(this[index])) {\n   return index\n   }\n   }\n   return -1\n}\n\n/**\n * Returns the last character.\n * \n * @throws NoSuchElementException if the char sequence is empty.\n * \n * @sample samples.text.Strings.last\n */\npublic fun CharSequence.last(): Char {\n   if (isEmpty())\n   throw NoSuchElementException(\"Char sequence is empty.\")\n   return this[lastIndex]\n}\n\n/**\n * Returns the last character
matching the given [predicate].\n * \n * @throws NoSuchElementException if no such character is found.\n * \n * @sample samples.text.Strings.last\n */\npublic inline fun CharSequence.last(predicate: (Char) -> Boolean): Char {\n   for (index in this.indices.reversed()) {\n   val element = this[index]\n   if (predicate(element)) return element\n   }\n   throw NoSuchElementException(\"Char sequence contains no character matching the predicate.\")\n}\n\n/**\n * Returns the last character, or `null` if the char sequence is empty.\n * \n * @sample samples.text.Strings.last\n */\npublic fun CharSequence.lastOrNull(): Char? {\n   return if (isEmpty()) null else this[length - 1]\n}\n\n/**\n * Returns the last character matching the given [predicate], or `null` if no such character was found.\n * \n * @sample samples.text.Strings.last\n */\npublic inline fun CharSequence.lastOrNull(predicate: (Char) -> Boolean): Char? {\n   for (index in this.indices.reversed()) {\n   val
element = this[index]\n   if (predicate(element)) return element\n   }\n   return null\n}\n\n/**\n * Returns a random character from this char sequence.\n * \n * @throws NoSuchElementException if this char sequence is empty.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.random(): Char {\n   return random(Random)\n}\n\n/**\n * Returns a random character from this char sequence using the specified source of randomness.\n * \n * @throws NoSuchElementException if this char sequence is empty.\n */\n@SinceKotlin(\"1.3\")\npublic fun CharSequence.random(random: Random): Char {\n   if (isEmpty())\n   throw NoSuchElementException(\"Char sequence is empty.\")\n   return get(random.nextInt(length))\n}\n\n/**\n * Returns a random character from this char sequence, or `null` if this char sequence is empty.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.randomOrNull():
Char? {\n   return randomOrNull(Random)\n}\n\n/**\n * Returns a random character from this char sequence using the specified source of randomness, or `null` if this char sequence is empty.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun CharSequence.randomOrNull(random: Random): Char? {\n   if (isEmpty())\n   return null\n   return get(random.nextInt(length))\n}\n\n/**\n * Returns the single character, or throws an exception if the char sequence is empty or has more than one character.\n */\npublic fun CharSequence.single(): Char {\n   return when (length) {\n   0 -> throw NoSuchElementException(\"Char sequence is empty.\")\n   1 -> this[0]\n   else -> throw IllegalArgumentException(\"Char sequence has more than one element.\")\n   }\n}\n\n/**\n * Returns the single character matching the given [predicate], or throws exception if there is no or more than one matching character.\n */\npublic inline fun CharSequence.single(predicate:
(Char) -> Boolean): Char {\n   var single: Char? = null\n   var found = false\n   for (element in this) {\n   if (predicate(element)) {\n   if (found) throw IllegalArgumentException(\"Char sequence contains more than one matching element.\")\n   single = element\n   found = true\n   }\n   }\n   if (!found) throw NoSuchElementException(\"Char sequence contains no character matching the predicate.\")\n   @Suppress(\"UNCHECKED_CAST\")\n   return single as Char\n}\n\n/**\n * Returns single character, or `null` if the char sequence is empty or has more than one character.\n */\npublic fun CharSequence.singleOrNull(): Char?

{\n return if (length == 1) this[0] else null\n}\n\n/**\n * Returns the single character matching the given [predicate], or `null` if character was not found or more than one character was found.\n */\npublic inline fun CharSequence.singleOrNull(predicate: (Char) -> Boolean): Char? {\n var single: Char? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if (found) return null\n single = element\n found = true\n }\n }\n if (!found) return null\n return single\n}\n\n/**\n * Returns a subsequence of this char sequence with the first [n] characters removed.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.drop\n */\npublic fun CharSequence.drop(n: Int): CharSequence {\n require(n >= 0) { \"Requested character count $n is less than zero.\" }\n return subSequence(n.coerceAtMost(length), length)\n}\n\n/**\n * Returns a string with the first [n] characters removed.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.drop\n */\npublic fun String.drop(n: Int): String {\n require(n >= 0) { \"Requested character count $n is less than zero.\" }\n return substring(n.coerceAtMost(length))\n}\n\n/**\n * Returns a subsequence of this char sequence with the last [n] characters removed.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.drop\n */\npublic fun CharSequence.dropLast(n: Int): CharSequence {\n require(n >= 0) { \"Requested character count $n is less than zero.\" }\n return take((length - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a string with the last [n] characters removed.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.drop\n */\npublic fun String.dropLast(n: Int): String {\n require(n >= 0) { \"Requested character count $n is less than zero.\" }\n return take((length - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a subsequence of this char sequence containing all characters except last characters that satisfy the given [predicate].\n * \n * @sample samples.text.Strings.drop\n */\npublic inline fun CharSequence.dropLastWhile(predicate: (Char) -> Boolean): CharSequence {\n for (index in lastIndex downTo 0)\n if (!predicate(this[index]))\n return subSequence(0, index + 1)\n return \"\"\n}\n\n/**\n * Returns a string containing all characters except last characters that satisfy the given [predicate].\n * \n * @sample samples.text.Strings.drop\n */\npublic inline fun String.dropLastWhile(predicate: (Char) -> Boolean): String {\n for (index in lastIndex downTo 0)\n if (!predicate(this[index]))\n return substring(0, index + 1)\n return \"\"\n}\n\n/**\n * Returns a subsequence of this char sequence containing all characters except first characters that satisfy the given [predicate].\n * \n * @sample samples.text.Strings.drop\n */\npublic inline fun CharSequence.dropWhile(predicate: (Char) -> Boolean): CharSequence {\n for (index in this.indices)\n if (!predicate(this[index]))\n return subSequence(index, length)\n return \"\"\n}\n\n/**\n * Returns a string containing all characters except first characters that satisfy the given [predicate].\n * \n * @sample samples.text.Strings.drop\n */\npublic inline fun String.dropWhile(predicate: (Char) -> Boolean): String {\n for (index in this.indices)\n if (!predicate(this[index]))\n return substring(index)\n return \"\"\n}\n\n/**\n * Returns a char sequence containing only those characters from the original char sequence that match the given [predicate].\n * \n * @sample samples.text.Strings.filter\n */\npublic inline fun CharSequence.filter(predicate: (Char) -> Boolean): CharSequence {\n return filterTo(StringBuilder(), predicate)\n}\n\n/**\n * Returns a string containing only those characters from the original string that match the given [predicate].\n * \n * @sample samples.text.Strings.filter\n */\npublic inline fun String.filter(predicate: (Char) -> Boolean): String {\n return filterTo(StringBuilder(), predicate).toString()\n}\n\n/**\n * Returns a char sequence containing only those characters from the original char sequence that match the given [predicate].\n * @param [predicate] function that takes the index of a character and the character itself\n * and returns the result of predicate evaluation on the character.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun CharSequence.filterIndexed(predicate: (index: Int, Char) -> Boolean): CharSequence {\n return filterIndexedTo(StringBuilder(), predicate)\n}\n\n/**\n * Returns a string containing only those characters from the original string that match the given [predicate].\n * @param [predicate] function that takes the index of a character and the character itself\n * and returns the result of predicate evaluation on the character.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\npublic inline fun String.filterIndexed(predicate: (index:

```
Int, Char) -> Boolean): String {\n    return filterIndexedTo(StringBuilder(), predicate).toString()\n}\n\n/**\n
 * Appends all characters matching the given [predicate] to the given [destination].\n * @param [predicate] function
that takes the index of a character and the character itself\n * and returns the result of predicate evaluation on the
character.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\npublic inline fun <C :
Appendable> CharSequence.filterIndexedTo(destination: C, predicate: (index: Int, Char) -> Boolean): C {\n
forEachIndexed { index, element ->\n        if (predicate(index, element)) destination.append(element)\n    }\n
return destination\n}\n\n/**\n * Returns a char sequence containing only those characters from the original char
sequence that do not match the given [predicate].\n * \n * @sample samples.text.Strings.filterNot\n */\npublic inline
fun CharSequence.filterNot(predicate: (Char) -> Boolean): CharSequence {\n    return filterNotTo(StringBuilder(),
predicate)\n}\n\n/**\n * Returns a string containing only
 those characters from the original string that do not match the given [predicate].\n * \n * @sample
samples.text.Strings.filterNot\n */\npublic inline fun String.filterNot(predicate: (Char) -> Boolean): String {\n
return filterNotTo(StringBuilder(), predicate).toString()\n}\n\n/**\n * Appends all characters not matching the given
[predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n
*/\npublic inline fun <C : Appendable> CharSequence.filterNotTo(destination: C, predicate: (Char) -> Boolean): C
{\n    for (element in this) if (!predicate(element)) destination.append(element)\n    return destination\n}\n\n/**\n *
Appends all characters matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n */\npublic inline fun <C : Appendable>
CharSequence.filterTo(destination: C, predicate: (Char) -> Boolean): C {\n    for (index in 0 until length) {\n        val
element
 = get(index)\n        if (predicate(element)) destination.append(element)\n    }\n    return destination\n}\n\n/**\n *
Returns a char sequence containing characters of the original char sequence at the specified range of [indices].\n
*/\npublic fun CharSequence.slice(indices: IntRange): CharSequence {\n    if (indices.isEmpty()) return \"\"\n
return subSequence(indices)\n}\n\n/**\n * Returns a string containing characters of the original string at the
specified range of [indices].\n */\npublic fun String.slice(indices: IntRange): String {\n    if (indices.isEmpty())
return \"\"\n    return substring(indices)\n}\n\n/**\n * Returns a char sequence containing characters of the original
char sequence at specified [indices].\n */\npublic fun CharSequence.slice(indices: Iterable<Int>): CharSequence {\n
 val size = indices.collectionSizeOrDefault(10)\n    if (size == 0) return \"\"\n    val result = StringBuilder(size)\n
for (i in indices) {\n        result.append(get(i))\n    }\n
  return result\n}\n\n/**\n * Returns a string containing characters of the original string at specified [indices].\n
*/\n@kotlin.internal.InlineOnly\npublic inline fun String.slice(indices: Iterable<Int>): String {\n    return (this as
CharSequence).slice(indices).toString()\n}\n\n/**\n * Returns a subsequence of this char sequence containing the
first [n] characters from this char sequence, or the entire char sequence if this char sequence is shorter.\n * \n *
@throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.take\n */\npublic fun
CharSequence.take(n: Int): CharSequence {\n    require(n >= 0) { \"Requested character count $n is less than zero.\"
}\n    return subSequence(0, n.coerceAtMost(length))\n}\n\n/**\n * Returns a string containing the first [n]
characters from this string, or the entire string if this string is shorter.\n * \n * @throws IllegalArgumentException if
[n] is negative.\n * \n * @sample samples.text.Strings.take\n */\npublic
 fun String.take(n: Int): String {\n    require(n >= 0) { \"Requested character count $n is less than zero.\" }\n    return
substring(0, n.coerceAtMost(length))\n}\n\n/**\n * Returns a subsequence of this char sequence containing the last
[n] characters from this char sequence, or the entire char sequence if this char sequence is shorter.\n * \n * @throws
IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.take\n */\npublic fun
CharSequence.takeLast(n: Int): CharSequence {\n    require(n >= 0) { \"Requested character count $n is less than
zero.\" }\n    val length = length\n    return subSequence(length - n.coerceAtMost(length), length)\n}\n\n/**\n *
Returns a string containing the last [n] characters from this string, or the entire string if this string is shorter.\n * \n *
@throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.text.Strings.take\n */\npublic fun
String.takeLast(n: Int): String {\n    require(n >= 0) { \"Requested character
```

```
count $n is less than zero.\" }\n    val length = length\n    return substring(length -
n.coerceAtMost(length))\n}\n\n/**\n * Returns a subsequence of this char sequence containing last characters that
satisfy the given [predicate].\n * \n * @sample samples.text.Strings.take\n */\npublic inline fun
CharSequence.takeLastWhile(predicate: (Char) -> Boolean): CharSequence {\n    for (index in lastIndex downTo 0)
{\n        if (!predicate(this[index])) {\n            return subSequence(index + 1, length)\n        }\n    }\n    return
subSequence(0, length)\n}\n\n/**\n * Returns a string containing last characters that satisfy the given [predicate].\n
* \n * @sample samples.text.Strings.take\n */\npublic inline fun String.takeLastWhile(predicate: (Char) ->
Boolean): String {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n            return
substring(index + 1)\n        }\n    }\n    return this\n}\n\n/**\n * Returns a subsequence of this char sequence
containing the first characters that satisfy the given [predicate].\n * \n * @sample samples.text.Strings.take\n
*/\npublic inline fun CharSequence.takeWhile(predicate: (Char) -> Boolean): CharSequence {\n    for (index in 0
until length)\n        if (!predicate(get(index))) {\n            return subSequence(0, index)\n        }\n    return
subSequence(0, length)\n}\n\n/**\n * Returns a string containing the first characters that satisfy the given
[predicate].\n * \n * @sample samples.text.Strings.take\n */\npublic inline fun String.takeWhile(predicate: (Char) ->
Boolean): String {\n    for (index in 0 until length)\n        if (!predicate(get(index))) {\n            return substring(0,
index)\n        }\n    return this\n}\n\n/**\n * Returns a char sequence with characters in reversed order.\n */\npublic
fun CharSequence.reversed(): CharSequence {\n    return StringBuilder(this).reverse()\n}\n\n/**\n * Returns a string
with characters in reversed order.\n */\n@kotlin.internal.InlineOnly\npublic
inline fun String.reversed(): String {\n    return (this as CharSequence).reversed().toString()\n}\n\n/**\n * Returns a
[Map] containing key-value pairs provided by [transform] function\n * applied to characters of the given char
sequence.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * The
returned map preserves the entry iteration order of the original char sequence.\n * \n * @sample
samples.text.Strings.associate\n */\npublic inline fun <K, V> CharSequence.associate(transform: (Char) -> Pair<K,
V>): Map<K, V> {\n    val capacity = mapCapacity(length).coerceAtLeast(16)\n    return
associateTo(LinkedHashMap<K, V>(capacity), transform)\n}\n\n/**\n * Returns a [Map] containing the characters
from the given char sequence indexed by the key\n * returned from [keySelector] function applied to each
character.\n * \n * If any two characters would have the same key returned by [keySelector] the last one gets added
to the map.\n *
\n * The returned map preserves the entry iteration order of the original char sequence.\n * \n * @sample
samples.text.Strings.associateBy\n */\npublic inline fun <K> CharSequence.associateBy(keySelector: (Char) -> K):
Map<K, Char> {\n    val capacity = mapCapacity(length).coerceAtLeast(16)\n    return
associateByTo(LinkedHashMap<K, Char>(capacity), keySelector)\n}\n\n/**\n * Returns a [Map] containing the
values provided by [valueTransform] and indexed by [keySelector] functions applied to characters of the given char
sequence.\n * \n * If any two characters would have the same key returned by [keySelector] the last one gets added
to the map.\n * \n * The returned map preserves the entry iteration order of the original char sequence.\n * \n *
@sample samples.text.Strings.associateByWithValueTransform\n */\npublic inline fun <K, V>
CharSequence.associateBy(keySelector: (Char) -> K, valueTransform: (Char) -> V): Map<K, V> {\n    val capacity
= mapCapacity(length).coerceAtLeast(16)\n
    return associateByTo(LinkedHashMap<K, V>(capacity), keySelector, valueTransform)\n}\n\n/**\n * Populates
and returns the [destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector]
function applied to each character of the given char sequence\n * and value is the character itself.\n * \n * If any two
characters would have the same key returned by [keySelector] the last one gets added to the map.\n * \n * @sample
samples.text.Strings.associateByTo\n */\npublic inline fun <K, M : MutableMap<in K, in Char>>
CharSequence.associateByTo(destination: M, keySelector: (Char) -> K): M {\n    for (element in this) {\n
destination.put(keySelector(element), element)\n    }\n    return destination\n}\n\n/**\n * Populates and returns the
[destination] mutable map with key-value pairs,\n * where key is provided by the [keySelector] function and\n * and
value is provided by the [valueTransform] function applied to characters of the given char sequence.\n * \n
```

```
 * If any two characters would have the same key returned by [keySelector] the last one gets added to the map.\n *
\n * @sample samples.text.Strings.associateByToWithValueTransform\n */\npublic inline fun <K, V, M :
MutableMap<in K, in V>> CharSequence.associateByTo(destination: M, keySelector: (Char) -> K,
valueTransform: (Char) -> V): M {\n    for (element in this) {\n        destination.put(keySelector(element),
valueTransform(element))\n    }\n    return destination\n}\n\n/**\n * Populates and returns the [destination] mutable
map with key-value pairs\n * provided by [transform] function applied to each character of the given char
sequence.\n * \n * If any of two pairs would have the same key the last one gets added to the map.\n * \n * @sample
samples.text.Strings.associateTo\n */\npublic inline fun <K, V, M : MutableMap<in K, in V>>
CharSequence.associateTo(destination: M, transform: (Char) -> Pair<K, V>): M {\n    for (element in this) {\n
destination += transform(element)\n
    }\n    return destination\n}\n\n/**\n * Returns a [Map] where keys are characters from the given char sequence
and values are\n * produced by the [valueSelector] function applied to each character.\n * \n * If any two characters
are equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the
original char sequence.\n * \n * @sample samples.text.Strings.associateWith\n */\n@SinceKotlin(\"1.3\")\npublic
inline fun <V> CharSequence.associateWith(valueSelector: (Char) -> V): Map<Char, V> {\n    val result =
LinkedHashMap<Char, V>(mapCapacity(length.coerceAtMost(128)).coerceAtLeast(16))\n    return
associateWithTo(result, valueSelector)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-
value pairs for each character of the given char sequence,\n * where key is the character itself and value is provided
by the [valueSelector] function applied to that key.\n * \n * If any two characters are equal, the last one
 overwrites the former value in the map.\n * \n * @sample samples.text.Strings.associateWithTo\n
*/\n@SinceKotlin(\"1.3\")\npublic inline fun <V, M : MutableMap<in Char, in V>>
CharSequence.associateWithTo(destination: M, valueSelector: (Char) -> V): M {\n    for (element in this) {\n
destination.put(element, valueSelector(element))\n    }\n    return destination\n}\n\n/**\n * Appends all characters to
the given [destination] collection.\n */\npublic fun <C : MutableCollection<in Char>>
CharSequence.toCollection(destination: C): C {\n    for (item in this) {\n        destination.add(item)\n    }\n    return
destination\n}\n\n/**\n * Returns a new [HashSet] of all characters.\n */\npublic fun CharSequence.toHashSet():
HashSet<Char> {\n    return toCollection(HashSet<Char>(mapCapacity(length.coerceAtMost(128))))\n}\n\n/**\n *
Returns a [List] containing all characters.\n */\npublic fun CharSequence.toList(): List<Char> {\n    return when
(length) {\n        0 -> emptyList()\n
  1 -> listOf(this[0])\n        else -> this.toMutableList()\n    }\n}\n\n/**\n * Returns a new [MutableList] filled with
all characters of this char sequence.\n */\npublic fun CharSequence.toMutableList(): MutableList<Char> {\n
return toCollection(ArrayList<Char>(length))\n}\n\n/**\n * Returns a [Set] of all characters.\n * \n * The returned
set preserves the element iteration order of the original char sequence.\n */\npublic fun CharSequence.toSet():
Set<Char> {\n    return when (length) {\n        0 -> emptySet()\n        1 -> setOf(this[0])\n        else ->
toCollection(LinkedHashSet<Char>(mapCapacity(length.coerceAtMost(128))))\n    }\n}\n\n/**\n * Returns a single
list of all elements yielded from results of [transform] function being invoked on each character of original char
sequence.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\npublic inline fun <R>
CharSequence.flatMap(transform: (Char) -> Iterable<R>): List<R> {\n    return flatMapTo(ArrayList<R>(),
 transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each character\n * and its index in the original char sequence.\n * \n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterable\")\n@kotlin.internal.InlineOnly\npublic
inline fun <R> CharSequence.flatMapIndexed(transform: (index: Int, Char) -> Iterable<R>): List<R> {\n    return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each character\n * and its index in the original char sequence, to the given
[destination].\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
```

ByLambdaReturnType\n@kotlin.jvm.JvmName(\"flatMapIndexedIterableTo\")\n@kotlin.internal.InlineOnly\npubli
c
inline fun <R, C : MutableCollection<in R>> CharSequence.flatMapIndexedTo(destination: C, transform: (index:
Int, Char) -> Iterable<R>): C {\n var index = 0\n for (element in this) {\n val list = transform(index++,
element)\n destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Appends all elements yielded from
results of [transform] function being invoked on each character of original char sequence, to the given
[destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> CharSequence.flatMapTo(destination: C,
transform: (Char) -> Iterable<R>): C {\n for (element in this) {\n val list = transform(element)\n
destination.addAll(list)\n }\n return destination\n}\n\n/**\n * Groups characters of the original char sequence by
the key returned by the given [keySelector] function\n * applied to each character and returns a map where each
group key is associated with
a list of corresponding characters.\n * \n * The returned map preserves the entry iteration order of the keys produced
from the original char sequence.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n
*/\npublic inline fun <K> CharSequence.groupBy(keySelector: (Char) -> K): Map<K, List<Char>> {\n return
groupByTo(LinkedHashMap<K, MutableList<Char>>(), keySelector)\n}\n\n/**\n * Groups values returned by the
[valueTransform] function applied to each character of the original char sequence\n * by the key returned by the
given [keySelector] function applied to the character\n * and returns a map where each group key is associated with
a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced
from the original char sequence.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V>
CharSequence.groupBy(keySelector: (Char) -> K, valueTransform: (Char)
-> V): Map<K, List<V>> {\n return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector,
valueTransform)\n}\n\n/**\n * Groups characters of the original char sequence by the key returned by the given
[keySelector] function\n * applied to each character and puts to the [destination] map each group key associated
with a list of corresponding characters.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupBy\n */\npublic inline fun <K, M : MutableMap<in K,
MutableList<Char>>> CharSequence.groupByTo(destination: M, keySelector: (Char) -> K): M {\n for (element in
this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<Char>() }\n
list.add(element)\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function
applied to each character of the original char sequence\n * by the key returned by the given [keySelector] function
applied
to the character\n * and puts to the [destination] map each group key associated with a list of corresponding
values.\n * \n * @return The [destination] map.\n * \n * @sample
samples.collections.Collections.Transformations.groupByKeysAndValues\n */\npublic inline fun <K, V, M :
MutableMap<in K, MutableList<V>>> CharSequence.groupByTo(destination: M, keySelector: (Char) -> K,
valueTransform: (Char) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list =
destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return
destination\n}\n\n/**\n * Creates a [Grouping] source from a char sequence to be used later with one of group-and-
fold operations\n * using the specified [keySelector] function to extract a key from each character.\n * \n * @sample
samples.collections.Grouping.groupingByEachCount\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <K>
CharSequence.groupingBy(crossinline keySelector: (Char)
-> K): Grouping<Char, K> {\n return object : Grouping<Char, K> {\n override fun sourceIterator():
Iterator<Char> = this@groupingBy.iterator()\n override fun keyOf(element: Char): K = keySelector(element)\n
}\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each character
in the original char sequence.\n * \n * @sample samples.text.Strings.map\n */\npublic inline fun <R>
CharSequence.map(transform: (Char) -> R): List<R> {\n return mapTo(ArrayList<R>(length),
transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each
character and its index in the original char sequence.\n * @param [transform] function that takes the index of a

character and the character itself\n * and returns the result of the transform applied to the character.\n */\npublic inline fun <R> CharSequence.mapIndexed(transform: (index: Int, Char) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(length),
transform)\n}\n\n/**\n * Returns a list containing only the non-null results of applying the given [transform] function\n * to each character and its index in the original char sequence.\n * @param [transform] function that takes the index of a character and the character itself\n * and returns the result of the transform applied to the character.\n */\npublic inline fun <R : Any> CharSequence.mapIndexedNotNull(transform: (index: Int, Char) -> R?): List<R> {\n return mapIndexedNotNullTo(ArrayList<R>(), transform)\n}\n\n/**\n * Applies the given [transform] function to each character and its index in the original char sequence\n * and appends only the non-null results to the given [destination].\n * @param [transform] function that takes the index of a character and the character itself\n * and returns the result of the transform applied to the character.\n */\npublic inline fun <R : Any, C : MutableCollection<in R>> CharSequence.mapIndexedNotNullTo(destination:
C, transform: (index: Int, Char) -> R?): C {\n forEachIndexed { index, element -> transform(index, element)?.let { destination.add(it) } }\n return destination\n}\n\n/**\n * Applies the given [transform] function to each character and its index in the original char sequence\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of a character and the character itself\n * and returns the result of the transform applied to the character.\n */\npublic inline fun <R, C : MutableCollection<in R>> CharSequence.mapIndexedTo(destination: C, transform: (index: Int, Char) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Returns a list containing only the non-null results of applying the given [transform] function\n * to each character in the original char sequence.\n * \n * @sample samples.collections.Collections.Transformations.mapNotNull\n */\npublic inline fun <R : Any> CharSequence.mapNotNull(transform: (Char) -> R?): List<R> {\n return mapNotNullTo(ArrayList<R>(), transform)\n}\n\n/**\n * Applies the given [transform] function to each character in the original char sequence\n * and appends only the non-null results to the given [destination].\n */\npublic inline fun <R : Any, C : MutableCollection<in R>> CharSequence.mapNotNullTo(destination: C, transform: (Char) -> R?): C {\n forEach { element -> transform(element)?.let { destination.add(it) } }\n return destination\n}\n\n/**\n * Applies the given [transform] function to each character of the original char sequence\n * and appends the results to the given [destination].\n */\npublic inline fun <R, C : MutableCollection<in R>> CharSequence.mapTo(destination: C, transform: (Char) -> R): C {\n for (item in this)\n destination.add(transform(item))\n return destination\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each character of the original
char sequence\n * into an [IndexedValue] containing the index of that character and the character itself.\n */\npublic fun CharSequence.withIndex(): Iterable<IndexedValue<Char>> {\n return IndexingIterable { iterator() }\n}\n\n/**\n * Returns `true` if all characters match the given [predicate].\n * \n * Note that if the char sequence contains no characters, the function returns `true`\n * because there are no characters in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in [\"Vacuous truth\"](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\npublic inline fun CharSequence.all(predicate: (Char) -> Boolean): Boolean {\n for (element in this) if (!predicate(element)) return false\n return true\n}\n\n/**\n * Returns `true` if char sequence has at least one character.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\npublic fun CharSequence.any():
Boolean {\n return !isEmpty()\n}\n\n/**\n * Returns `true` if at least one character matches the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\npublic inline fun CharSequence.any(predicate: (Char) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return true\n return false\n}\n\n/**\n * Returns the length of this char sequence.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.count(): Int {\n return length\n}\n\n/**\n * Returns the number of characters matching the given [predicate].\n */\npublic inline fun CharSequence.count(predicate: (Char) -> Boolean): Int {\n var count = 0\n for (element in this) if

(predicate(element)) ++count\n   return count\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each character.\n * \n * Returns the specified [initial] value if the char sequence
is empty.\n * \n * @param [operation] function that takes current accumulator value and a character, and calculates the next accumulator value.\n */\npublic inline fun <R> CharSequence.fold(initial: R, operation: (acc: R, Char) -> R): R {\n   var accumulator = initial\n   for (element in this) accumulator = operation(accumulator, element)\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each character with its index in the original char sequence.\n * \n * Returns the specified [initial] value if the char sequence is empty.\n * \n * @param [operation] function that takes the index of a character, current accumulator value\n * and the character itself, and calculates the next accumulator value.\n */\npublic inline fun <R> CharSequence.foldIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): R {\n   var index = 0\n   var accumulator = initial\n   for (element
in this) accumulator = operation(index++, accumulator, element)\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each character and current accumulator value.\n * \n * Returns the specified [initial] value if the char sequence is empty.\n * \n * @param [operation] function that takes a character and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> CharSequence.foldRight(initial: R, operation: (Char, acc: R) -> R): R {\n   var index = lastIndex\n   var accumulator = initial\n   while (index >= 0) {\n   accumulator = operation(get(index--), accumulator)\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each character with its index in the original char sequence and current accumulator value.\n * \n * Returns the specified [initial] value if the
char sequence is empty.\n * \n * @param [operation] function that takes the index of a character, the character itself\n * and current accumulator value, and calculates the next accumulator value.\n */\npublic inline fun <R> CharSequence.foldRightIndexed(initial: R, operation: (index: Int, Char, acc: R) -> R): R {\n   var index = lastIndex\n   var accumulator = initial\n   while (index >= 0) {\n   accumulator = operation(index, get(index), accumulator)\n   --index\n   }\n   return accumulator\n}\n\n/**\n * Performs the given [action] on each character.\n */\npublic inline fun CharSequence.forEach(action: (Char) -> Unit): Unit {\n   for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each character, providing sequential index with the character.\n * @param [action] function that takes the index of a character and the character itself\n * and performs the action on the character.\n */\npublic inline fun CharSequence.forEachIndexed(action: (index:
Int, Char) -> Unit): Unit {\n   var index = 0\n   for (item in this) action(index++, item)\n}\n\n/**\n * Returns the largest character.\n * \n * @throws NoSuchElementException if the char sequence is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun CharSequence.max(): Char {\n   if (isEmpty()) throw NoSuchElementException()\n   var max = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (max < e) max = e\n   }\n   return max\n}\n\n/**\n * Returns the first character yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the char sequence is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.maxBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <R : Comparable<R>> CharSequence.maxBy(selector: (Char) -> R): Char {\n   if (isEmpty()) throw NoSuchElementException()\n
    var maxElem = this[0]\n   val lastIndex = this.lastIndex\n   if (lastIndex == 0) return maxElem\n   var maxValue = selector(maxElem)\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   val v = selector(e)\n   if (maxValue < v) {\n   maxElem = e\n   maxValue = v\n   }\n   }\n   return maxElem\n}\n\n/**\n * Returns the first character yielding the largest value of the given function or `null` if there are no characters.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> CharSequence.maxByOrNull(selector: (Char) -> R): Char? {\n   if (isEmpty()) return null\n   var maxElem = this[0]\n   val lastIndex = this.lastIndex\n   if (lastIndex == 0) return maxElem\n   var maxValue =

selector(maxElem)\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   val v = selector(e)\n   if (maxValue < v) {\n   maxElem = e\n
        maxValue = v\n   }\n   }\n   return maxElem\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each character in the char sequence.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the char sequence is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.maxOf(selector: (Char) ->
Double): Double {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue = selector(this[0])\n   for
(i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each character in the char sequence.\n * \n * If any of values produced
by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the char
sequence is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.maxOf(selector: (Char) ->
Float): Float {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue = selector(this[0])\n   for (i
in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each character in the char sequence.\n * \n * @throws NoSuchElementException if the char sequence is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharSequence.maxOf(selector:
(Char) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue = selector(this[0])\n   for
(i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (maxValue < v) {\n   maxValue = v\n   }\n   }\n
return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n *
applied to each character in the char sequence or `null` if there are no characters.\n * \n * If any of values produced
by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.maxOfOrNull(selector:
(Char) -> Double): Double? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in
1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each character in the char sequence or `null` if there are no characters.\n * \n * If any of values produced by
[selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.maxOfOrNull(selector:
(Char) -> Float): Float? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in
1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each character in the char sequence or `null` if there are no characters.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> CharSequence.maxOfOrNull(selector: (Char) -> R): R? {\n   if (isEmpty()) return
null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if
(maxValue < v) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value
according to the provided [comparator]\n * among all values produced by [selector] function applied to each
character in the char sequence.\n * \n * @throws NoSuchElementException if the char sequence is empty.\n

*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharSequence.maxOfWith(comparator: Comparator<in R>, selector: (Char) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each character in the char sequence or `null` if there are no characters.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharSequence.maxOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest character or `null` if there are no characters.\n */\n@SinceKotlin(\"1.4\")\npublic fun CharSequence.maxOrNull(): Char? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the first character having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the char sequence is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun CharSequence.maxWith(comparator: Comparator<in Char>): Char {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first character having the largest value according to the provided [comparator] or `null` if there are no characters.\n */\n@SinceKotlin(\"1.4\")\npublic fun CharSequence.maxWithOrNull(comparator: Comparator<in Char>): Char? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the smallest character.\n * \n * @throws NoSuchElementException if the char sequence is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun CharSequence.min(): Char {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the first character yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the char sequence is empty.\n * \n * @sample samples.collections.Collections.Aggregates.minBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <R : Comparable<R>> CharSequence.minBy(selector: (Char) -> R): Char {\n if (isEmpty()) throw NoSuchElementException()\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\n/**\n * Returns the first character yielding the smallest value of the given function or `null` if there are no characters.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R : Comparable<R>> CharSequence.minByOrNull(selector: (Char) -> R): Char? {\n if (isEmpty()) return null\n var minElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return minElem\n var minValue = selector(minElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (minValue > v) {\n minElem = e\n minValue = v\n }\n }\n return minElem\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each character in the char sequence.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException

if the char sequence is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.minOf(selector: (Char) ->
Double): Double {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue\n}\n\n/**\n * Returns the smallest
value among all values produced by [selector] function\n * applied to each character in the char sequence.\n * \n *
If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws
NoSuchElementException if the char sequence is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.minOf(selector: (Char) ->
Float): Float {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in
1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return
minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each character in the char sequence.\n * \n * @throws NoSuchElementException if the char sequence is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> CharSequence.minOf(selector: (Char) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v =
selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n *
Returns the smallest value among all values produced by [selector] function\n * applied to each character in the char
sequence or `null` if there are no characters.\n * \n * If any of values produced by [selector] function is `NaN`, the
returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.minOfOrNull(selector:
(Char) -> Double): Double? {\n if (isEmpty()) return null\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue =
minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each character in the char sequence or `null` if there are no characters.\n * \n *
If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.minOfOrNull(selector:
(Char) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex)
{\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n *
Returns the smallest value among all values produced by [selector] function\n * applied to each character
in the char sequence or `null` if there are no characters.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>>
CharSequence.minOfOrNull(selector: (Char) -> R): R? {\n if (isEmpty()) return null\n var minValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n
minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided
[comparator]\n * among all values produced by [selector] function applied to each character in the char sequence.\n
* \n * @throws NoSuchElementException if the char sequence is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharSequence.minOfWith(comparator:
Comparator<in R>, selector: (Char) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var
minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n *

Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each character in the char sequence or `null` if there are no characters.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.internal.InlineOnly\npublic inline fun <R> CharSequence.minOfWithOrNull(comparator: Comparator<in R>, selector: (Char) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0)\n {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest character or `null` if there are no characters.\n */\n@SinceKotlin(\"1.4\")\npublic fun CharSequence.minOrNull(): Char? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the first character having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the char sequence is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow\")\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun CharSequence.minWith(comparator: Comparator<in Char>): Char {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first character having the smallest value according to the provided [comparator] or `null` if there are no characters.\n */\n@SinceKotlin(\"1.4\")\npublic fun CharSequence.minWithOrNull(comparator: Comparator<in Char>): Char? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns `true` if the char sequence has no characters.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\npublic fun CharSequence.none(): Boolean {\n return isEmpty()\n}\n\n/**\n * Returns `true` if no characters match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n */\npublic inline fun CharSequence.none(predicate: (Char) -> Boolean): Boolean {\n for (element in this) if (predicate(element)) return false\n return true\n}\n\n/**\n * Performs the given [action] on each character and returns the char sequence itself afterwards.\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <S : CharSequence> S.onEach(action: (Char) -> Unit): S {\n return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each character, providing sequential index with the character,\n * and returns the char sequence itself afterwards.\n * @param [action] function that takes the index of a character and the character itself\n * and performs the action on the character.\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <S : CharSequence> S.onEachIndexed(action: (index: Int, Char) -> Unit): S {\n return apply { forEachIndexed(action) }\n}\n\n/**\n * Accumulates value starting with the first character and applying [operation] from left to right\n * to current accumulator value and each character.\n * \n * Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and a character,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun CharSequence.reduce(operation: (acc: Char, Char) -> Char): Char {\n if (isEmpty())\n throw UnsupportedOperationException(\"Empty char sequence can't be reduced.\")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first character and applying [operation] from left to right\n * to current accumulator value and each character with its index in the original char sequence.\n * \n * Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of a character, current accumulator value and the character itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\npublic inline fun CharSequence.reduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): Char {\n if (isEmpty())\n throw UnsupportedOperationException(\"Empty char sequence can't be reduced.\")\n var accumulator = this[0]\n for

(index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first character and applying [operation] from left to right\n * to current accumulator value and each character with its index in the original char sequence.\n * \n * Returns `null` if the char sequence is empty.\n * \n * @param [operation] function that takes the index of a character, current accumulator value and the character itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun CharSequence.reduceIndexedOrNull(operation: (index: Int, acc: Char, Char) -> Char): Char? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first character and applying [operation] from left to right\n * to current accumulator value and each character.\n * \n * Returns `null` if the char sequence is empty.\n * \n * @param [operation] function that takes current accumulator value and a character,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun CharSequence.reduceOrNull(operation: (acc: Char, Char) -> Char): Char? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last character and applying [operation] from right to left\n * to each character and current accumulator value.\n * \n * Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes a character and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun CharSequence.reduceRight(operation: (Char, acc: Char) -> Char): Char {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty char sequence can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last character and applying [operation] from right to left\n * to each character with its index in the original char sequence and current accumulator value.\n * \n * Throws an exception if this char sequence is empty. If the char sequence can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of a character, the character itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\npublic inline fun CharSequence.reduceRightIndexed(operation: (index: Int, Char, acc: Char) -> Char): Char {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty char sequence can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last character and applying [operation] from right to left\n * to each character with its index in the original char sequence and current accumulator value.\n * \n * Returns `null` if the char sequence is empty.\n * \n * @param [operation] function that takes the index of a character, the character itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\npublic inline fun CharSequence.reduceRightIndexedOrNull(operation: (index: Int, Char, acc: Char) -> Char): Char? {\n var index = lastIndex\n if (index < 0) return null\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last character and applying [operation] from right to left\n * to each character and current accumulator value.\n * \n * Returns `null` if the char sequence is empty.\n * \n * @param [operation] function that takes a character and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample

samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun CharSequence.reduceRightOrNull(operation: (Char, acc: Char) -> Char): Char? {\n var index = lastIndex\n if (index < 0) return null\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each character and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and a character, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R> CharSequence.runningFold(initial: R, operation: (acc: R, Char) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(length + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each character, its index in the original char sequence and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of a character, current accumulator value\n * and the character itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\npublic inline fun <R> CharSequence.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(length + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each character and current accumulator value that starts with the first character of this char sequence.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and a character, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\npublic inline fun CharSequence.runningReduce(operation: (acc: Char, Char) -> Char): List<Char> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<Char>(length).apply { add(accumulator) }\n for (index in 1 until length) {\n accumulator = operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each character, its index in the original char sequence and current accumulator value that starts with the first character of this char sequence.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of a character, current accumulator value\n * and the character itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\npublic inline fun CharSequence.runningReduceIndexed(operation: (index: Int, acc: Char, Char) -> Char): List<Char> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<Char>(length).apply { add(accumulator) }\n for (index in 1 until length) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each character and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and a character, and calculates the next

accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <R> CharSequence.scan(initial: R, operation: (acc: R, Char) -> R): List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each character, its index in the original char sequence and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of a character, current accumulator value\n * and the character itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <R> CharSequence.scanIndexed(initial: R, operation: (index: Int, acc: R, Char) -> R): List<R> {\n return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each character in the char sequence.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun CharSequence.sumBy(selector: (Char) -> Int): Int {\n var sum: Int = 0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each character in the char sequence.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic inline fun CharSequence.sumByDouble(selector: (Char) -> Double): Double {\n var sum: Double = 0.0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each character in the char sequence.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfDouble\")\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.sumOf(selector: (Char) -> Double): Double {\n var sum: Double = 0.toDouble()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each character in the char sequence.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfInt\")\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.sumOf(selector: (Char) -> Int): Int {\n var sum: Int = 0.toInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each character in the char sequence.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfLong\")\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.sumOf(selector: (Char) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each character in the char sequence.\n */\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.sumOf(selector: (Char) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each character in the char sequence.\n */\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic

inline fun CharSequence.sumOf(selector: (Char) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Splits this char sequence into a list of strings each not exceeding the given [size].\n * \n * The last string in the resulting list may have fewer characters than the given [size].\n * \n * @param size the number of elements to take in each string, must be positive and can be greater than the number of elements in this char sequence.\n * \n * @sample samples.text.Strings.chunked\n */\n@SinceKotlin(\"1.2\")\npublic fun CharSequence.chunked(size: Int): List<String> {\n return windowed(size, size, partialWindows = true)\n}\n\n/**\n * Splits this char sequence into several
char sequences each not exceeding the given [size]\n * and applies the given [transform] function to an each.\n * \n * @return list of results of the [transform] applied to an each char sequence.\n * \n * Note that the char sequence passed to the [transform] function is ephemeral and is valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a snapshot of it.\n * The last char sequence may have fewer characters than the given [size].\n * \n * @param size the number of elements to take in each char sequence, must be positive and can be greater than the number of elements in this char sequence.\n * \n * @sample samples.text.Strings.chunkedTransform\n */\n@SinceKotlin(\"1.2\")\npublic fun <R> CharSequence.chunked(size: Int, transform: (CharSequence) -> R): List<R> {\n return windowed(size, size, partialWindows = true, transform = transform)\n}\n\n/**\n * Splits this char sequence into a sequence of strings each not exceeding the
given [size].\n * \n * The last string in the resulting sequence may have fewer characters than the given [size].\n * \n * @param size the number of elements to take in each string, must be positive and can be greater than the number of elements in this char sequence.\n * \n * @sample samples.collections.Collections.Transformations.chunked\n */\n@SinceKotlin(\"1.2\")\npublic fun CharSequence.chunkedSequence(size: Int): Sequence<String> {\n return chunkedSequence(size) { it.toString() }\n}\n\n/**\n * Splits this char sequence into several char sequences each not exceeding the given [size]\n * and applies the given [transform] function to an each.\n * \n * @return sequence of results of the [transform] applied to an each char sequence.\n * \n * Note that the char sequence passed to the [transform] function is ephemeral and is valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a snapshot of it.\n * The last char sequence may
have fewer characters than the given [size].\n * \n * @param size the number of elements to take in each char sequence, must be positive and can be greater than the number of elements in this char sequence.\n * \n * @sample samples.text.Strings.chunkedTransformToSequence\n */\n@SinceKotlin(\"1.2\")\npublic fun <R> CharSequence.chunkedSequence(size: Int, transform: (CharSequence) -> R): Sequence<R> {\n return windowedSequence(size, size, partialWindows = true, transform = transform)\n}\n\n/**\n * Splits the original char sequence into pair of char sequences,\n * where *first* char sequence contains characters for which [predicate] yielded `true`,\n * while *second* char sequence contains characters for which [predicate] yielded `false`.\n * \n * @sample samples.text.Strings.partition\n */\npublic inline fun CharSequence.partition(predicate: (Char) -> Boolean): Pair<CharSequence, CharSequence> {\n val first = StringBuilder()\n val second = StringBuilder()\n for (element in this)
{\n if (predicate(element)) {\n first.append(element)\n } else {\n second.append(element)\n }\n }\n return Pair(first, second)\n}\n\n/**\n * Splits the original string into pair of strings,\n * where *first* string contains characters for which [predicate] yielded `true`,\n * while *second* string contains characters for which [predicate] yielded `false`.\n * \n * @sample samples.text.Strings.partition\n */\npublic inline fun String.partition(predicate: (Char) -> Boolean): Pair<String, String> {\n val first = StringBuilder()\n val second = StringBuilder()\n for (element in this) {\n if (predicate(element)) {\n first.append(element)\n } else {\n second.append(element)\n }\n }\n return Pair(first.toString(), second.toString())\n}\n\n/**\n * Returns a list of snapshots of the window of the given [size]\n * sliding along this char sequence with the given [step], where each\n * snapshot
is a string.\n * \n * Several last strings may have fewer characters than the given [size].\n * \n * Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.\n * @param size the number of elements to take in each window\n * @param step the number of elements to move the window forward

by on an each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the end if any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample samples.collections.Sequences.Transformations.takeWindows\n */\n@SinceKotlin(\"1.2\")\npublic fun CharSequence.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false): List<String> {\n   return windowed(size, step, partialWindows) { it.toString() }\n}\n\n/**\n * Returns a list of results of applying the given [transform] function to\n * an each char sequence representing a view over the window of the given [size]\n * sliding
along this char sequence with the given [step].\n * \n * Note that the char sequence passed to the [transform] function is ephemeral and is valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a snapshot of it.\n * Several last char sequences may have fewer characters than the given [size].\n * \n * Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.\n * @param size the number of elements to take in each window\n * @param step the number of elements to move the window forward by on an each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the end if any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample samples.collections.Sequences.Transformations.averageWindows\n */\n@SinceKotlin(\"1.2\")\npublic fun <R> CharSequence.windowed(size: Int, step: Int = 1, partialWindows: Boolean = false, transform:
(CharSequence) -> R): List<R> {\n   checkWindowSizeStep(size, step)\n   val thisSize = this.length\n   val resultCapacity = thisSize / step + if (thisSize % step == 0) 0 else 1\n   val result = ArrayList<R>(resultCapacity)\n   var index = 0\n   while (index in 0 until thisSize) {\n   val end = index + size\n   val coercedEnd = if (end < 0 || end > thisSize) { if (partialWindows) thisSize else break } else end\n   result.add(transform(subSequence(index, coercedEnd)))\n   index += step\n   }\n   return result\n}\n\n/**\n * Returns a sequence of snapshots of the window of the given [size]\n * sliding along this char sequence with the given [step], where each\n * snapshot is a string.\n * \n * Several last strings may have fewer characters than the given [size].\n * \n * Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.\n * @param size the number of elements to take in each window\n * @param step
the number of elements to move the window forward by on an each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the end if any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample samples.collections.Sequences.Transformations.takeWindows\n */\n@SinceKotlin(\"1.2\")\npublic fun CharSequence.windowedSequence(size: Int, step: Int = 1, partialWindows: Boolean = false): Sequence<String> {\n   return windowedSequence(size, step, partialWindows) { it.toString() }\n}\n\n/**\n * Returns a sequence of results of applying the given [transform] function to\n * an each char sequence representing a view over the window of the given [size]\n * sliding along this char sequence with the given [step].\n * \n * Note that the char sequence passed to the [transform] function is ephemeral and is valid only inside that function.\n * You should not store it or allow it to escape in some way, unless you made a snapshot of it.\n
* Several last char sequences may have fewer characters than the given [size].\n * \n * Both [size] and [step] must be positive and can be greater than the number of elements in this char sequence.\n * @param size the number of elements to take in each window\n * @param step the number of elements to move the window forward by on an each step, by default 1\n * @param partialWindows controls whether or not to keep partial windows in the end if any,\n * by default `false` which means partial windows won't be preserved\n * \n * @sample samples.collections.Sequences.Transformations.averageWindows\n */\n@SinceKotlin(\"1.2\")\npublic fun <R> CharSequence.windowedSequence(size: Int, step: Int = 1, partialWindows: Boolean = false, transform:
(CharSequence) -> R): Sequence<R> {\n   checkWindowSizeStep(size, step)\n   val windows = (if (partialWindows) indices else 0 until length - size + 1) step step\n   return windows.asSequence().map { index ->\n   val end = index + size\n
val coercedEnd = if (end < 0 || end > length) length else end\n   transform(subSequence(index, coercedEnd))\n   }\n}\n\n/**\n * Returns a list of pairs built from the characters of `this` and the [other] char sequences with the same index\n * The returned list has length of the shortest char sequence.\n * \n * @sample samples.text.Strings.zip\n

*/\npublic infix fun CharSequence.zip(other: CharSequence): List<Pair<Char, Char>> {\n   return zip(other) { c1, c2 -> c1 to c2 }\n}\n\n/**\n * Returns a list of values built from the characters of `this` and the [other] char sequences with the same index\n * using the provided [transform] function applied to each pair of characters.\n * The returned list has length of the shortest char sequence.\n * \n * @sample samples.text.Strings.zipWithTransform\n */\npublic inline fun <V> CharSequence.zip(other: CharSequence, transform: (a: Char, b: Char) -> V): List<V> {\n   val length = minOf(this.length, other.length)\n   val list = ArrayList<V>(length)\n   for (i in 0 until length) {\n   list.add(transform(this[i], other[i]))\n   }\n   return list\n}\n\n/**\n * Returns a list of pairs of each two adjacent characters in this char sequence.\n * \n * The returned list is empty if this char sequence contains less than two characters.\n * \n * @sample samples.collections.Collections.Transformations.zipWithNext\n */\n@SinceKotlin(\"1.2\")\npublic fun CharSequence.zipWithNext(): List<Pair<Char, Char>> {\n   return zipWithNext { a, b -> a to b }\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to an each pair of two adjacent characters in this char sequence.\n * \n * The returned list is empty if this char sequence contains less than two characters.\n * \n * @sample samples.collections.Collections.Transformations.zipWithNextToFindDeltas\n */\n@SinceKotlin(\"1.2\")\npublic inline fun <R> CharSequence.zipWithNext(transform: (a: Char, b: Char) -> R): List<R> {\n   val size = length - 1\n   if (size < 1) return emptyList()\n   val result = ArrayList<R>(size)\n   for (index in 0 until size) {\n   result.add(transform(this[index], this[index + 1]))\n   }\n   return result\n}\n\n/**\n * Creates an [Iterable] instance that wraps the original char sequence returning its characters when being iterated.\n */\npublic fun CharSequence.asIterable(): Iterable<Char> {\n   if (this is String && isEmpty()) return emptyList()\n   return Iterable { this.iterator() }\n}\n\n/**\n * Creates a [Sequence] instance that wraps the original char sequence returning its characters when being iterated.\n */\npublic fun CharSequence.asSequence(): Sequence<Char> {\n   if (this is String && isEmpty()) return emptySequence()\n   return Sequence { this.iterator() }\n}\n\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"StringsKt\")\n\npackage kotlin.text\n\nimport kotlin.contracts.contract\nimport kotlin.jvm.JvmName\n\n/**\n * Returns a copy of this string converted to upper case using the rules of the default locale.\n */\n@Deprecated(\"Use uppercase() instead.\", ReplaceWith(\"uppercase()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic expect fun String.toUpperCase(): String\n\n/**\n * Returns a copy of this string converted to upper case using Unicode mapping rules of the invariant locale.\n *\n * This function supports one-to-many and many-to-one character mapping,\n * thus the length of the returned string can be different from the length of the original string.\n *\n * @sample samples.text.Strings.uppercase\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun String.uppercase(): String\n\n/**\n * Returns a copy of this string converted to lower case using the rules of the default locale.\n */\n@Deprecated(\"Use lowercase() instead.\", ReplaceWith(\"lowercase()\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic expect fun String.toLowerCase(): String\n\n/**\n * Returns a copy of this string converted to lower case using Unicode mapping rules of the invariant locale.\n *\n * This function supports one-to-many and many-to-one character mapping,\n * thus the length of the returned string can be different from the length of the original string.\n *\n * @sample samples.text.Strings.lowercase\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun String.lowercase(): String\n\n/**\n * Returns a copy of this string having its first letter titlecased using the rules of the default locale,\n * or the original string if it's empty or already starts with a title case letter.\n *\n * The title case of a character is usually the same as its upper case with several exceptions.\n * The particular list of characters with the special title case form depends on the underlying platform.\n *\n * @sample samples.text.Strings.capitalize\n */\n@Deprecated(\"Use replaceFirstChar instead.\", ReplaceWith(\"replaceFirstChar { if (it.isLowerCase())

it.titlecase() else it.toString() }\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic expect fun String.capitalize(): String\n\n/**\n * Returns a copy of this string having its first letter lowercased using the rules of the default locale,\n * or the original string if it's empty or already starts with a lower case letter.\n *\n * @sample samples.text.Strings.decapitalize\n */\n@Deprecated(\"Use replaceFirstChar instead.\", ReplaceWith(\"replaceFirstChar { it.lowercase() }\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic expect fun String.decapitalize(): String\n\n/**\n * Returns a sub sequence of this char sequence having leading and trailing characters matching the [predicate] removed.\n */\npublic inline fun CharSequence.trim(predicate: (Char) -> Boolean): CharSequence {\n var startIndex = 0\n var endIndex = length - 1\n var startFound = false\n\n while (startIndex <= endIndex) {\n val index = if (!startFound) startIndex else endIndex\n val match = predicate(this[index])\n\n if (!startFound) {\n if (!match)\n startFound = true\n else\n startIndex += 1\n } else {\n if (!match)\n break\n else\n endIndex -= 1\n }\n }\n\n return subSequence(startIndex, endIndex + 1)\n}\n\n/**\n * Returns a string having leading and trailing characters matching the [predicate] removed.\n */\npublic inline fun String.trim(predicate: (Char) -> Boolean): String =\n (this as CharSequence).trim(predicate).toString()\n\n/**\n * Returns a sub sequence of this char sequence having leading characters matching the [predicate] removed.\n */\npublic inline fun CharSequence.trimStart(predicate: (Char) -> Boolean): CharSequence {\n for (index in this.indices)\n if (!predicate(this[index]))\n return subSequence(index, length)\n\n return \"\"\n}\n\n/**\n * Returns a string having leading characters matching the [predicate] removed.\n */\npublic inline fun String.trimStart(predicate: (Char) -> Boolean): String =\n (this as CharSequence).trimStart(predicate).toString()\n\n/**\n * Returns a sub sequence of this char sequence having trailing characters matching the [predicate] removed.\n */\npublic inline fun CharSequence.trimEnd(predicate: (Char) -> Boolean): CharSequence {\n for (index in this.indices.reversed())\n if (!predicate(this[index]))\n return subSequence(0, index + 1)\n\n return \"\"\n}\n\n/**\n * Returns a string having trailing characters matching the [predicate] removed.\n */\npublic inline fun String.trimEnd(predicate: (Char) -> Boolean): String =\n (this as CharSequence).trimEnd(predicate).toString()\n\n/**\n * Returns a sub sequence of this char sequence having leading and trailing characters from the [chars] array removed.\n */\npublic fun CharSequence.trim(vararg chars: Char): CharSequence = trim { it in chars }\n\n/**\n * Returns a string having leading and trailing characters from the [chars] array removed.\n */\npublic fun String.trim(vararg chars: Char): String = trim { it in chars }\n\n/**\n * Returns a sub sequence of this char sequence having leading characters from the [chars] array removed.\n */\npublic fun CharSequence.trimStart(vararg chars: Char): CharSequence = trimStart { it in chars }\n\n/**\n * Returns a string having leading characters from the [chars] array removed.\n */\npublic fun String.trimStart(vararg chars: Char): String = trimStart { it in chars }\n\n/**\n * Returns a sub sequence of this char sequence having trailing characters from the [chars] array removed.\n */\npublic fun CharSequence.trimEnd(vararg chars: Char): CharSequence = trimEnd { it in chars }\n\n/**\n * Returns a string having trailing characters from the [chars] array removed.\n */\npublic fun String.trimEnd(vararg chars: Char): String = trimEnd { it in chars }\n\n/**\n * Returns a sub sequence of this char sequence having leading and trailing whitespace removed.\n */\npublic fun CharSequence.trim(): CharSequence = trim(Char::isWhitespace)\n\n/**\n * Returns a string having leading and trailing whitespace removed.\n */\n@kotlin.internal.InlineOnly\npublic inline fun String.trim(): String = (this as CharSequence).trim().toString()\n\n/**\n * Returns a sub sequence of this char sequence having leading whitespace removed.\n */\npublic fun CharSequence.trimStart(): CharSequence = trimStart(Char::isWhitespace)\n\n/**\n * Returns a string having leading whitespace removed.\n */\n@kotlin.internal.InlineOnly\npublic inline fun String.trimStart(): String = (this as CharSequence).trimStart().toString()\n\n/**\n * Returns a sub sequence of this char sequence having trailing whitespace removed.\n */\npublic fun CharSequence.trimEnd(): CharSequence = trimEnd(Char::isWhitespace)\n\n/**\n * Returns a string having trailing whitespace removed.\n */\n@kotlin.internal.InlineOnly\npublic inline fun String.trimEnd(): String = (this as CharSequence).trimEnd().toString()\n\n/**\n * Returns a char sequence with content of this char sequence padded at the beginning\n * to the specified [length] with the specified character or space.\n *\n * @param length the desired

string length.\n * @param padChar the character to pad string with, if it has length less than the [length] specified. Space is used by default.\n * @return Returns a char sequence of length at least [length] consisting of `this` char sequence prepended with [padChar] as many times\n * as are necessary to reach that length.\n * @sample samples.text.Strings.padStart\n */\npublic fun CharSequence.padStart(length: Int, padChar: Char = ' '): CharSequence {\n if (length < 0)\n throw IllegalArgumentException(\"Desired length $length is less than zero.\")\n if (length <= this.length)\n return this.subSequence(0, this.length)\n\n val sb = StringBuilder(length)\n for (i in 1..(length - this.length))\n sb.append(padChar)\n sb.append(this)\n return sb\n}\n\n/**\n * Pads the string to the specified [length] at the beginning with the specified character or space.\n *\n * @param length the desired string length.\n * @param padChar the character to pad string with, if it has length less than the [length] specified. Space is used by default.\n * @return Returns a string of length at least [length] consisting of `this` string prepended with [padChar] as many times\n * as are necessary to reach that length.\n * @sample samples.text.Strings.padStart\n */\npublic fun String.padStart(length: Int, padChar: Char = ' '): String =\n (this as CharSequence).padStart(length, padChar).toString()\n\n/**\n * Returns a char sequence with content of this char sequence padded at the end\n * to the specified [length] with the specified character or space.\n *\n * @param length the desired string length.\n * @param padChar the character to pad string with, if it has length less than the [length] specified. Space is used by default.\n * @return Returns a char sequence of length at least [length] consisting of `this` char sequence appended with [padChar] as many times\n * as are necessary to reach that length.\n * @sample samples.text.Strings.padEnd\n */\npublic fun CharSequence.padEnd(length: Int, padChar: Char = ' '): CharSequence {\n if (length < 0)\n throw IllegalArgumentException(\"Desired length $length is less than zero.\")\n if (length <= this.length)\n return this.subSequence(0, this.length)\n\n val sb = StringBuilder(length)\n sb.append(this)\n for (i in 1..(length - this.length))\n sb.append(padChar)\n return sb\n}\n\n/**\n * Pads the string to the specified [length] at the end with the specified character or space.\n *\n * @param length the desired string length.\n * @param padChar the character to pad string with, if it has length less than the [length] specified. Space is used by default.\n * @return Returns a string of length at least [length] consisting of `this` string appended with [padChar] as many times\n * as are necessary to reach that length.\n * @sample samples.text.Strings.padEnd\n */\npublic fun String.padEnd(length: Int, padChar: Char = ' '): String =\n (this as CharSequence).padEnd(length, padChar).toString()\n\n/**\n * Returns `true` if this nullable char sequence is either `null` or empty.\n *\n * @sample samples.text.Strings.stringIsNullOrEmpty\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence?.isNullOrEmpty(): Boolean {\n contract {\n returns(false) implies (this@isNullOrEmpty != null)\n }\n\n return this == null || this.length == 0\n}\n\n/**\n * Returns `true` if this char sequence is empty (contains no characters).\n *\n * @sample samples.text.Strings.stringIsEmpty\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.isEmpty(): Boolean = length == 0\n\n/**\n * Returns `true` if this char sequence is not empty.\n *\n * @sample samples.text.Strings.stringIsNotEmpty\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.isNotEmpty(): Boolean = length > 0\n\n// implemented differently in JVM and JS\n//public fun String.isBlank(): Boolean = length() == 0 || all { it.isWhitespace() }\n\n/**\n * Returns `true` if this char sequence is not empty and contains some characters except of whitespace characters.\n *\n * @sample samples.text.Strings.stringIsNotBlank\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.isNotBlank(): Boolean = !isBlank()\n\n/**\n * Returns `true` if this nullable char sequence is either `null` or empty or consists solely of whitespace characters.\n *\n * @sample samples.text.Strings.stringIsNullOrBlank\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence?.isNullOrBlank(): Boolean {\n contract {\n returns(false) implies (this@isNullOrBlank != null)\n }\n\n return this == null || this.isBlank()\n}\n\n/**\n * Iterator for characters of the given char sequence.\n */\npublic operator fun CharSequence.iterator(): CharIterator = object : CharIterator() {\n private var index = 0\n\n public override fun nextChar(): Char = get(index++)\n\n public override fun hasNext(): Boolean = index < length\n}\n\n/** Returns the string if it is not `null`, or the empty string otherwise. */\n@kotlin.internal.InlineOnly\npublic inline fun

```
String?.orEmpty(): String = this ?: \"\"\n\n/**\n * Returns this char sequence if it's not empty\n * or the result of calling [defaultValue] function if the char sequence is empty.\n *\n * @sample samples.text.Strings.stringIfEmpty\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun <C, R> C.ifEmpty(defaultValue: () -> R): R where C : CharSequence, C : R =\n    if (isEmpty()) defaultValue() else this\n\n/**\n * Returns this char sequence if it is not empty and doesn't consist solely of whitespace characters,\n * or the result of calling [defaultValue] function otherwise.\n *\n * @sample samples.text.Strings.stringIfBlank\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun <C, R> C.ifBlank(defaultValue: () -> R): R where C : CharSequence, C : R =\n    if (isBlank()) defaultValue() else this\n\n/**\n * Returns the range of valid character indices for this char sequence.\n */\npublic val CharSequence.indices: IntRange\n    get() = 0..length - 1\n\n/**\n * Returns the index of the last character in the char sequence or -1 if it is empty.\n */\npublic val CharSequence.lastIndex: Int\n    get() = this.length - 1\n\n/**\n * Returns `true` if this CharSequence has Unicode surrogate pair at the specified [index].\n */\npublic fun CharSequence.hasSurrogatePairAt(index: Int): Boolean {\n    return index in 0..length - 2\n            && this[index].isHighSurrogate()\n            && this[index + 1].isLowSurrogate()\n}\n\n/**\n * Returns a substring specified by the given [range] of indices.\n */\npublic fun String.substring(range: IntRange): String = substring(range.start, range.endInclusive + 1)\n\n/**\n * Returns a subsequence of this char sequence specified by the given [range] of indices.\n */\npublic fun CharSequence.subSequence(range: IntRange): CharSequence = subSequence(range.start, range.endInclusive + 1)\n\n/**\n * Returns a subsequence of this char sequence.\n *\n * This extension is chosen only for invocation with old-named parameters.\n * Replace parameter names with the same as those of [CharSequence.subSequence].\n */\n@kotlin.internal.InlineOnly\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\") // false warning\n@Deprecated(\"Use parameters named startIndex and endIndex.\", ReplaceWith(\"subSequence(startIndex = start, endIndex = end)\"))\npublic inline fun String.subSequence(start: Int, end: Int): CharSequence = subSequence(start, end)\n\n/**\n * Returns a substring of chars from a range of this char sequence starting at the [startIndex] and ending right before the [endIndex].\n *\n * @param startIndex the start index (inclusive).\n * @param endIndex the end index (exclusive). If not specified, the length of the char sequence is used.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.substring(startIndex: Int, endIndex: Int = length): String = subSequence(startIndex, endIndex).toString()\n\n/**\n * Returns a substring of chars at indices from the specified [range] of this char sequence.\n */\npublic fun CharSequence.substring(range: IntRange): String = subSequence(range.start, range.endInclusive + 1).toString()\n\n/**\n * Returns a substring before the first occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.substringBefore(delimiter: Char, missingDelimiterValue: String = this): String {\n    val index = indexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else substring(0, index)\n}\n\n/**\n * Returns a substring before the first occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.substringBefore(delimiter: String, missingDelimiterValue: String = this): String {\n    val index = indexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else substring(0, index)\n}\n\n/**\n * Returns a substring after the first occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.substringAfter(delimiter: Char, missingDelimiterValue: String = this): String {\n    val index = indexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else substring(index + 1, length)\n}\n\n/**\n * Returns a substring after the first occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.substringAfter(delimiter: String, missingDelimiterValue: String = this): String {\n    val index = indexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else substring(index + delimiter.length, length)\n}\n\n/**\n * Returns a substring before the last occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.substringBeforeLast(delimiter: Char, missingDelimiterValue: String = this): String {\n    val index = lastIndexOf(delimiter)\n    return if (index == -1) missingDelimiterValue else
```

substring(0, index)\n}\n\n/**\n * Returns a substring before the last occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to
the original string.\n */\npublic fun String.substringBeforeLast(delimiter: String, missingDelimiterValue: String = this): String {\n val index = lastIndexOf(delimiter)\n return if (index == -1) missingDelimiterValue else substring(0, index)\n}\n\n/**\n * Returns a substring after the last occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.substringAfterLast(delimiter: Char, missingDelimiterValue: String = this): String {\n val index = lastIndexOf(delimiter)\n return if (index == -1) missingDelimiterValue else substring(index + 1, length)\n}\n\n/**\n * Returns a substring after the last occurrence of [delimiter].\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.substringAfterLast(delimiter: String, missingDelimiterValue: String = this): String {\n val index = lastIndexOf(delimiter)\n return if (index == -1) missingDelimiterValue else substring(index + delimiter.length, length)\n}\n\n/**\n * Returns a char sequence with content of this char sequence where its part at the given range\n * is replaced with the [replacement] char sequence.\n * @param startIndex the index of the first character to be replaced.\n * @param endIndex the index of the first character after the replacement to keep in the string.\n */\npublic fun CharSequence.replaceRange(startIndex: Int, endIndex: Int, replacement: CharSequence): CharSequence {\n if (endIndex < startIndex)\n throw IndexOutOfBoundsException(\"End index ($endIndex) is less than start index ($startIndex).\")\n val sb = StringBuilder()\n sb.appendRange(this, 0, startIndex)\n sb.append(replacement)\n sb.appendRange(this, endIndex, length)\n return sb\n}\n\n/**\n * Replaces the part of the string at the given range with the [replacement] char sequence.\n * @param startIndex the
index of the first character to be replaced.\n * @param endIndex the index of the first character after the replacement to keep in the string.\n */\n@kotlin.internal.InlineOnly\npublic inline fun String.replaceRange(startIndex: Int, endIndex: Int, replacement: CharSequence): String =\n (this as CharSequence).replaceRange(startIndex, endIndex, replacement).toString()\n\n/**\n * Returns a char sequence with content of this char sequence where its part at the given [range]\n * is replaced with the [replacement] char sequence.\n *\n * The end index of the [range] is included in the part to be replaced.\n */\npublic fun CharSequence.replaceRange(range: IntRange, replacement: CharSequence): CharSequence =\n replaceRange(range.start, range.endInclusive + 1, replacement)\n\n/**\n * Replace the part of string at the given [range] with the [replacement] string.\n *\n * The end index of the [range] is included in the part to be replaced.\n */\n@kotlin.internal.InlineOnly\npublic inline fun
String.replaceRange(range: IntRange, replacement: CharSequence): String =\n (this as CharSequence).replaceRange(range, replacement).toString()\n\n/**\n * Returns a char sequence with content of this char sequence where its part at the given range is removed.\n *\n * @param startIndex the index of the first character to be removed.\n * @param endIndex the index of the first character after the removed part to keep in the string.\n *\n * [endIndex] is not included in the removed part.\n */\npublic fun CharSequence.removeRange(startIndex: Int, endIndex: Int): CharSequence {\n if (endIndex < startIndex)\n throw IndexOutOfBoundsException(\"End index ($endIndex) is less than start index ($startIndex).\")\n\n if (endIndex == startIndex)\n return this.subSequence(0, length)\n\n val sb = StringBuilder(length - (endIndex - startIndex))\n sb.appendRange(this, 0, startIndex)\n sb.appendRange(this, endIndex, length)\n return sb\n}\n\n/**\n * Removes the part of a
string at a given range.\n * @param startIndex the index of the first character to be removed.\n * @param endIndex the index of the first character after the removed part to keep in the string.\n *\n * [endIndex] is not included in the removed part.\n */\n@kotlin.internal.InlineOnly\npublic inline fun String.removeRange(startIndex: Int, endIndex: Int): String =\n (this as CharSequence).removeRange(startIndex, endIndex).toString()\n\n/**\n * Returns a char sequence with content of this char sequence where its part at the given [range] is removed.\n *\n * The end index of the [range] is included in the removed part.\n */\npublic fun CharSequence.removeRange(range: IntRange): CharSequence = removeRange(range.start, range.endInclusive + 1)\n\n/**\n * Removes the part of a string at the given [range].\n *\n * The end index of the [range] is included in the removed part.\n

*/\n@kotlin.internal.InlineOnly\npublic inline fun String.removeRange(range: IntRange): String =\n   (this as CharSequence).removeRange(range).toString()\n\n/**\n
 * If this char sequence starts with the given [prefix], returns a new char sequence\n * with the prefix removed. Otherwise, returns a new char sequence with the same characters.\n */\npublic fun CharSequence.removePrefix(prefix: CharSequence): CharSequence {\n   if (startsWith(prefix)) {\n   return subSequence(prefix.length, length)\n   }\n   return subSequence(0, length)\n}\n\n/**\n * If this string starts with the given [prefix], returns a copy of this string\n * with the prefix removed. Otherwise, returns this string.\n */\npublic fun String.removePrefix(prefix: CharSequence): String {\n   if (startsWith(prefix)) {\n   return substring(prefix.length)\n   }\n   return this\n}\n\n/**\n * If this char sequence ends with the given [suffix], returns a new char sequence\n * with the suffix removed. Otherwise, returns a new char sequence with the same characters.\n */\npublic fun CharSequence.removeSuffix(suffix: CharSequence):
 CharSequence {\n   if (endsWith(suffix)) {\n   return subSequence(0, length - suffix.length)\n   }\n   return subSequence(0, length)\n}\n\n/**\n * If this string ends with the given [suffix], returns a copy of this string\n * with the suffix removed. Otherwise, returns this string.\n */\npublic fun String.removeSuffix(suffix: CharSequence): String {\n   if (endsWith(suffix)) {\n   return substring(0, length - suffix.length)\n   }\n   return this\n}\n\n/**\n * When this char sequence starts with the given [prefix] and ends with the given [suffix],\n * returns a new char sequence having both the given [prefix] and [suffix] removed.\n * Otherwise returns a new char sequence with the same characters.\n */\npublic fun CharSequence.removeSurrounding(prefix: CharSequence, suffix: CharSequence): CharSequence {\n   if ((length >= prefix.length + suffix.length) && startsWith(prefix) && endsWith(suffix)) {\n
 return subSequence(prefix.length, length - suffix.length)\n
  }\n   return subSequence(0, length)\n}\n\n/**\n * Removes from a string both the given [prefix] and [suffix] if and only if\n * it starts with the [prefix] and ends with the [suffix].\n * Otherwise returns this string unchanged.\n */\npublic fun String.removeSurrounding(prefix: CharSequence, suffix: CharSequence): String {\n   if ((length >= prefix.length + suffix.length) && startsWith(prefix) && endsWith(suffix)) {\n   return substring(prefix.length, length - suffix.length)\n   }\n   return this\n}\n\n/**\n * When this char sequence starts with and ends with the given [delimiter],\n * returns a new char sequence having this [delimiter] removed both from the start and end.\n * Otherwise returns a new char sequence with the same characters.\n */\npublic fun CharSequence.removeSurrounding(delimiter: CharSequence): CharSequence = removeSurrounding(delimiter, delimiter)\n\n/**\n * Removes the given [delimiter] string from both the start and the end of this string\n * if and only
 if it starts with and ends with the [delimiter].\n * Otherwise returns this string unchanged.\n */\npublic fun String.removeSurrounding(delimiter: CharSequence): String = removeSurrounding(delimiter, delimiter)\n\n/**\n * Replace part of string before the first occurrence of given delimiter with the [replacement] string.\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.replaceBefore(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String {\n   val index = indexOf(delimiter)\n   return if (index == -1) missingDelimiterValue else replaceRange(0, index, replacement)\n}\n\n/**\n * Replace part of string before the first occurrence of given delimiter with the [replacement] string.\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.replaceBefore(delimiter: String, replacement:
 String, missingDelimiterValue: String = this): String {\n   val index = indexOf(delimiter)\n   return if (index == -1) missingDelimiterValue else replaceRange(0, index, replacement)\n}\n\n/**\n * Replace part of string after the first occurrence of given delimiter with the [replacement] string.\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.replaceAfter(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String {\n   val index = indexOf(delimiter)\n   return if (index == -1) missingDelimiterValue else replaceRange(index + 1, length, replacement)\n}\n\n/**\n * Replace part of string after the first occurrence of given delimiter with the [replacement] string.\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.replaceAfter(delimiter: String, replacement: String, missingDelimiterValue:

String = this): String {\n val index = indexOf(delimiter)\n return if (index == -1) missingDelimiterValue else replaceRange(index + delimiter.length, length, replacement)\n}\n\n/**\n * Replace part of string after the last occurrence of given delimiter with the [replacement] string.\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.replaceAfterLast(delimiter: String, replacement: String, missingDelimiterValue: String = this): String {\n val index = lastIndexOf(delimiter)\n return if (index == -1) missingDelimiterValue else replaceRange(index + delimiter.length, length, replacement)\n}\n\n/**\n * Replace part of string after the last occurrence of given delimiter with the [replacement] string.\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.replaceAfterLast(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String {\n val index = lastIndexOf(delimiter)\n return if (index == -1) missingDelimiterValue else replaceRange(index + 1, length, replacement)\n}\n\n/**\n * Replace part of string before the last occurrence of given delimiter with the [replacement] string.\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.replaceBeforeLast(delimiter: Char, replacement: String, missingDelimiterValue: String = this): String {\n val index = lastIndexOf(delimiter)\n return if (index == -1) missingDelimiterValue else replaceRange(0, index, replacement)\n}\n\n/**\n * Replace part of string before the last occurrence of given delimiter with the [replacement] string.\n * If the string does not contain the delimiter, returns [missingDelimiterValue] which defaults to the original string.\n */\npublic fun String.replaceBeforeLast(delimiter: String, replacement: String, missingDelimiterValue: String = this): String {\n val index = lastIndexOf(delimiter)\n return if (index == -1) missingDelimiterValue else replaceRange(0, index, replacement)\n}\n\n\n// public fun String.replace(oldChar: Char, newChar: Char, ignoreCase: Boolean): String // JVM- and JS-specific\n// public fun String.replace(oldValue: String, newValue: String, ignoreCase: Boolean): String // JVM- and JS-specific\n\n/**\n * Returns a new string obtained by replacing each substring of this char sequence that matches the given regular expression\n * with the given [replacement].\n *\n * The [replacement] can consist of any combination of literal text and $-substitutions. To treat the replacement string\n * literally escape it with the [kotlin.text.Regex.Companion.escapeReplacement] method.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.replace(regex: Regex, replacement: String): String = regex.replace(this, replacement)\n\n/**\n * Returns a
new string obtained by replacing each substring of this char sequence that matches the given regular expression\n * with the result of the given function [transform] that takes [MatchResult] and returns a string to be used as a\n * replacement for that match.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.replace(regex: Regex, noinline transform: (MatchResult) -> CharSequence): String =\n regex.replace(this, transform)\n\n/**\n * Replaces the first occurrence of the given regular expression [regex] in this char sequence with specified [replacement] expression.\n *\n * @param replacement A replacement expression that can include substitutions. See [Regex.replaceFirst] for details.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.replaceFirst(regex: Regex, replacement: String): String = regex.replaceFirst(this, replacement)\n\n/**\n * Returns a copy of this string having its first character replaced with the result of the specified [transform],\n *
or the original string if it's empty.\n *\n * @param transform function that takes the first character and returns the result of the transform applied to the character.\n *\n * @sample samples.text.Strings.replaceFirstChar\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@JvmName(\"replaceFirstCharWithChar\")\n@kotlin.internal.InlineOnly\npublic inline fun String.replaceFirstChar(transform: (Char) -> Char): String {\n return if (isNotEmpty()) transform(this[0]) + substring(1) else this\n}\n\n/**\n * Returns a copy of this string having its first character replaced with the result of the specified [transform],\n * or the original string if it's empty.\n *\n * @param transform function that takes the first character and returns the result of the transform applied to the character.\n *\n * @sample samples.text.Strings.replaceFirstChar\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@OptIn(kotlin.experimental.Exper

imentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@JvmName(\"replaceFirstCharWithCharSequence\")\n@kotlin.internal.InlineOnly\npublic inline fun String.replaceFirstChar(transform: (Char) -> CharSequence): String {\n return if (isNotEmpty()) transform(this[0]).toString() + substring(1) else this\n}\n\n\n/**\n * Returns `true` if this char sequence matches the given regular expression.\n */\n@kotlin.internal.InlineOnly\npublic inline infix fun CharSequence.matches(regex: Regex): Boolean = regex.matches(this)\n\n/**\n * Implementation of [regionMatches] for CharSequences.\n * Invoked when it's already known that arguments are not Strings, so that no additional type checks are performed.\n */\ninternal fun CharSequence.regionMatchesImpl(thisOffset: Int, other: CharSequence, otherOffset: Int, length: Int, ignoreCase: Boolean): Boolean {\n if ((otherOffset < 0) || (thisOffset < 0) || (thisOffset > this.length - length) || (otherOffset > other.length - length)) {\n return false\n }\n\n for (index in 0 until length) {\n if (!this[thisOffset + index].equals(other[otherOffset + index], ignoreCase))\n return false\n }\n return true\n}\n\n/**\n * Returns `true` if this char sequence starts with the specified character.\n */\npublic fun CharSequence.startsWith(char: Char, ignoreCase: Boolean = false): Boolean =\n this.length > 0 && this[0].equals(char, ignoreCase)\n\n/**\n * Returns `true` if this char sequence ends with the specified character.\n */\npublic fun CharSequence.endsWith(char: Char, ignoreCase: Boolean = false): Boolean =\n this.length > 0 && this[lastIndex].equals(char, ignoreCase)\n\n/**\n * Returns `true` if this char sequence starts with the specified prefix.\n */\npublic fun CharSequence.startsWith(prefix: CharSequence, ignoreCase: Boolean = false): Boolean {\n if (!ignoreCase && this is String && prefix is String)\n return this.startsWith(prefix)\n else\n return regionMatchesImpl(0, prefix, 0, prefix.length, ignoreCase)\n}\n\n/**\n * Returns `true` if a substring of this char sequence starting at the specified offset [startIndex] starts with the specified prefix.\n */\npublic fun CharSequence.startsWith(prefix: CharSequence, startIndex: Int, ignoreCase: Boolean = false): Boolean {\n if (!ignoreCase && this is String && prefix is String)\n return this.startsWith(prefix, startIndex)\n else\n return regionMatchesImpl(startIndex, prefix, 0, prefix.length, ignoreCase)\n}\n\n/**\n * Returns `true` if this char sequence ends with the specified suffix.\n */\npublic fun CharSequence.endsWith(suffix: CharSequence, ignoreCase: Boolean = false): Boolean {\n if (!ignoreCase && this is String && suffix is String)\n return this.endsWith(suffix)\n else\n return regionMatchesImpl(length - suffix.length, suffix, 0, suffix.length, ignoreCase)\n}\n\n\n// common prefix and suffix\n\n/**\n * Returns the longest string `prefix` such that this char sequence and [other] char sequence both start with this prefix,\n * taking care not to split surrogate pairs.\n * If this and [other] have no common prefix, returns the empty string.\n\n * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @sample samples.text.Strings.commonPrefixWith\n */\npublic fun CharSequence.commonPrefixWith(other: CharSequence, ignoreCase: Boolean = false): String {\n val shortestLength = minOf(this.length, other.length)\n\n var i = 0\n while (i < shortestLength && this[i].equals(other[i], ignoreCase = ignoreCase)) {\n i++\n }\n if (this.hasSurrogatePairAt(i - 1) || other.hasSurrogatePairAt(i - 1)) {\n i--\n }\n return subSequence(0, i).toString()\n}\n\n/**\n * Returns the longest string `suffix` such that this char sequence and [other] char sequence both end with this suffix,\n * taking care not to split surrogate pairs.\n * If this and [other] have no common suffix, returns the empty string.\n\n * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @sample samples.text.Strings.commonSuffixWith\n */\npublic fun CharSequence.commonSuffixWith(other: CharSequence, ignoreCase: Boolean = false): String {\n val thisLength = this.length\n val otherLength = other.length\n val shortestLength = minOf(thisLength, otherLength)\n\n var i = 0\n while (i < shortestLength && this[thisLength - i - 1].equals(other[otherLength - i - 1], ignoreCase = ignoreCase)) {\n i++\n }\n if (this.hasSurrogatePairAt(thisLength - i - 1) || other.hasSurrogatePairAt(otherLength - i - 1)) {\n i--\n }\n return subSequence(thisLength - i, thisLength).toString()\n}\n\n\n// indexOfAny()\n\n/**\n * Finds the index of the first occurrence of any of the specified [chars] in this char sequence,\n * starting from the specified [startIndex] and optionally ignoring the case.\n *\n * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @return An index of the first occurrence of matched character from [chars] or -1 if none of [chars] are found.\n *\n */\npublic fun CharSequence.indexOfAny(chars: CharArray,

```
startIndex: Int = 0, ignoreCase: Boolean = false): Int {\n    if (!ignoreCase && chars.size == 1 && this is String) {\n        val char = chars.single()\n        return nativeIndexOf(char, startIndex)\n    }\n\n    for (index in startIndex.coerceAtLeast(0)..lastIndex) {\n        val charAtIndex = get(index)\n        if (chars.any { it.equals(charAtIndex, ignoreCase) })\n            return index\n    }\n    return -1\n}\n\n/**\n * Finds the index of the last occurrence of any of the specified [chars] in this char sequence,\n * starting from the specified [startIndex] and optionally ignoring the case.\n *\n * @param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @return An index of the last occurrence of matched character from [chars] or -1 if none of [chars] are found.\n *\n */\npublic fun CharSequence.lastIndexOfAny(chars: CharArray, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {\n    if (!ignoreCase && chars.size == 1 && this is String) {\n        val char = chars.single()\n        return nativeLastIndexOf(char, startIndex)\n    }\n\n\n    for (index in startIndex.coerceAtMost(lastIndex) downTo 0) {\n        val charAtIndex = get(index)\n        if (chars.any { it.equals(charAtIndex, ignoreCase) })\n            return index\n    }\n\n    return -1\n}\n\n\nprivate fun CharSequence.indexOf(other: CharSequence, startIndex: Int, endIndex: Int, ignoreCase: Boolean, last: Boolean = false): Int {\n    val indices = if (!last)\n        startIndex.coerceAtLeast(0)..endIndex.coerceAtMost(length)\n    else\n        startIndex.coerceAtMost(lastIndex) downTo endIndex.coerceAtLeast(0)\n\n    if (this is String && other is String) { // smart cast\n        for (index in indices) {\n            if (other.regionMatches(0, this, index, other.length, ignoreCase))\n                return index\n        }\n    } else {\n        for (index in indices) {\n            if (other.regionMatchesImpl(0, this, index, other.length, ignoreCase))\n                return index\n        }\n    }\n    return -1\n}\n\nprivate fun CharSequence.findAnyOf(strings: Collection<String>, startIndex: Int, ignoreCase: Boolean, last: Boolean): Pair<Int, String>? {\n    if (!ignoreCase && strings.size == 1) {\n        val string = strings.single()\n        val index = if (!last) indexOf(string, startIndex) else lastIndexOf(string, startIndex)\n        return if (index < 0) null else index to string\n    }\n\n    val indices = if (!last) startIndex.coerceAtLeast(0)..length\n    else startIndex.coerceAtMost(lastIndex) downTo 0\n\n    if (this is String) {\n        for (index in indices) {\n            val matchingString = strings.firstOrNull { it.regionMatches(0, this, index, it.length, ignoreCase) }\n            if (matchingString != null)\n                return index to matchingString\n        }\n    } else {\n        for (index in indices) {\n            val matchingString = strings.firstOrNull { it.regionMatchesImpl(0, this, index, it.length, ignoreCase) }\n            if (matchingString != null)\n                return index to matchingString\n        }\n    }\n\n    return null\n}\n\n/**\n * Finds the first occurrence of any of the specified [strings] in this char sequence,\n * starting from the specified [startIndex] and optionally ignoring the case.\n *\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return A pair of an index of the first occurrence of matched string from [strings] and the string matched\n * or `null` if none of [strings] are found.\n *\n * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from\n * the beginning to the end of this string, and finds at each position the first element in [strings]\n * that matches this string at that position.\n */\npublic fun CharSequence.findAnyOf(strings: Collection<String>, startIndex: Int = 0, ignoreCase: Boolean = false): Pair<Int, String>? =\n    findAnyOf(strings, startIndex, ignoreCase, last = false)\n\n/**\n * Finds the last occurrence of any of the specified [strings] in this char sequence,\n * starting from the specified [startIndex] and optionally ignoring the case.\n *\n * @param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return A pair of an index of the last occurrence of matched string from [strings] and the string matched or `null` if none of [strings] are found.\n *\n * To avoid ambiguous results when strings in [strings] have characters in common, this method proceeds from\n * the end toward the beginning of this string, and finds at each position the first element in [strings]\n * that matches this string at that position.\n */\npublic fun CharSequence.findLastAnyOf(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Pair<Int, String>? =\n    findAnyOf(strings, startIndex, ignoreCase, last = true)\n\n/**\n * Finds the index of the first occurrence of any of the specified [strings] in this char sequence,\n * starting from the specified
```

[startIndex] and optionally ignoring the case.\n *\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return An index of the first occurrence of matched string from [strings] or -1 if none of [strings] are found.\n *\n * To avoid ambiguous
results when strings in [strings] have characters in common, this method proceeds from\n * the beginning to the end of this string, and finds at each position the first element in [strings]\n * that matches this string at that position.\n */\npublic fun CharSequence.indexOfAny(strings: Collection<String>, startIndex: Int = 0, ignoreCase: Boolean = false): Int =\n findAnyOf(strings, startIndex, ignoreCase, last = false)?.first ?: -1\n\n/**\n * Finds the index of the last occurrence of any of the specified [strings] in this char sequence,\n * starting from the specified [startIndex] and optionally ignoring the case.\n *\n * @param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return An index of the last occurrence of matched string from [strings] or -1 if none of [strings] are found.\n *\n * To avoid ambiguous
results when strings in [strings] have characters in common, this method proceeds from\n * the end toward the beginning of this string, and finds at each position the first element in [strings]\n * that matches this string at that position.\n */\npublic fun CharSequence.lastIndexOfAny(strings: Collection<String>, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int =\n findAnyOf(strings, startIndex, ignoreCase, last = true)?.first ?: -1\n\n\n// indexOf\n\n/**\n * Returns the index within this string of the first occurrence of the specified character, starting from the specified [startIndex].\n *\n * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @return An index of the first occurrence of [char] or -1 if none is found.\n */\npublic fun CharSequence.indexOf(char: Char, startIndex: Int = 0, ignoreCase: Boolean = false): Int {\n return if (ignoreCase || this !is String)\n indexOfAny(charArrayOf(char), startIndex,
ignoreCase)\n else\n nativeIndexOf(char, startIndex)\n}\n\n/**\n * Returns the index within this char sequence of the first occurrence of the specified [string],\n * starting from the specified [startIndex].\n *\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return An index of the first occurrence of [string] or `-1` if none is found.\n * @sample samples.text.Strings.indexOf\n */\npublic fun CharSequence.indexOf(string: String, startIndex: Int = 0, ignoreCase: Boolean = false): Int {\n return if (ignoreCase || this !is String)\n indexOf(string, startIndex, length, ignoreCase)\n else\n nativeIndexOf(string, startIndex)\n}\n\n/**\n * Returns the index within this char sequence of the last occurrence of the specified character,\n * starting from the specified [startIndex].\n *\n * @param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to ignore character case when matching a character. By default `false`.\n * @return An index of the last occurrence of [char] or -1 if none is found.\n */\npublic fun CharSequence.lastIndexOf(char: Char, startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {\n return if (ignoreCase || this !is String)\n lastIndexOfAny(charArrayOf(char), startIndex, ignoreCase)\n else\n nativeLastIndexOf(char, startIndex)\n}\n\n/**\n * Returns the index within this char sequence of the last occurrence of the specified [string],\n * starting from the specified [startIndex].\n *\n * @param startIndex The index of character to start searching at. The search proceeds backward toward the beginning of the string.\n * @param ignoreCase `true` to ignore character case when matching a string. By default `false`.\n * @return An index of the last occurrence of [string] or -1 if none is found.\n */\npublic fun CharSequence.lastIndexOf(string: String,
startIndex: Int = lastIndex, ignoreCase: Boolean = false): Int {\n return if (ignoreCase || this !is String)\n indexOf(string, startIndex, 0, ignoreCase, last = true)\n else\n nativeLastIndexOf(string, startIndex)\n}\n\n/**\n * Returns `true` if this char sequence contains the specified [other] sequence of characters as a substring.\n *\n * @param ignoreCase `true` to ignore character case when comparing strings. By default `false`.\n */\n@Suppress(\"INAPPLICABLE_OPERATOR_MODIFIER\")\npublic operator fun CharSequence.contains(other: CharSequence, ignoreCase: Boolean = false): Boolean =\n if (other is String)\n indexOf(other, ignoreCase = ignoreCase) >= 0\n else\n indexOf(other, 0, length, ignoreCase) >= 0\n\n\n\n/**\n * Returns `true` if this char sequence contains the specified character [char].\n *\n * @param ignoreCase `true` to ignore character case when comparing characters. By default `false`.\n

```
*/\n@Suppress(\"INAPPLICABLE_OPERATOR_MODIFIER\")\npublic
 operator fun CharSequence.contains(char: Char, ignoreCase: Boolean = false): Boolean =\n    indexOf(char,
ignoreCase = ignoreCase) >= 0\n\n/**\n * Returns `true` if this char sequence contains at least one match of the
specified regular expression [regex].\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun
CharSequence.contains(regex: Regex): Boolean = regex.containsMatchIn(this)\n\n\n//
rangesDelimitedBy\n\n\nprivate class DelimitedRangesSequence(\n    private val input: CharSequence,\n    private
val startIndex: Int,\n    private val limit: Int,\n    private val getNextMatch: CharSequence.(currentIndex: Int) ->
Pair<Int, Int>?\n) : Sequence<IntRange> {\n\n    override fun iterator(): Iterator<IntRange> = object :
Iterator<IntRange> {\n        var nextState: Int = -1 // -1 for unknown, 0 for done, 1 for continue\n        var
currentStartIndex: Int = startIndex.coerceIn(0, input.length)\n        var nextSearchIndex: Int = currentStartIndex\n
 var nextItem:
 IntRange? = null\n        var counter: Int = 0\n\n        private fun calcNext() {\n            if (nextSearchIndex < 0) {\n
            nextState = 0\n                nextItem = null\n            } else {\n                if (limit > 0 && ++counter >= limit ||
nextSearchIndex > input.length) {\n                    nextItem = currentStartIndex..input.lastIndex\n
nextSearchIndex = -1\n                } else {\n                    val match = input.getNextMatch(nextSearchIndex)\n
    if (match == null) {\n                        nextItem = currentStartIndex..input.lastIndex\n
nextSearchIndex = -1\n                    } else {\n                        val (index, length) = match\n                        nextItem =
currentStartIndex until index\n                        currentStartIndex = index + length\n                        nextSearchIndex =
currentStartIndex + if (length == 0) 1 else 0\n                    }\n
   }\n                nextState = 1\n            }\n        }\n\n        override fun next(): IntRange {\n            if (nextState == -
1)\n                calcNext()\n            if (nextState == 0)\n                throw NoSuchElementException()\n            val
result = nextItem as IntRange\n            // Clean next to avoid keeping reference on yielded instance\n
nextItem = null\n            nextState = -1\n            return result\n        }\n\n        override fun hasNext(): Boolean {\n
    if (nextState == -1)\n                calcNext()\n            return nextState == 1\n        }\n    }\n}\n\n/**\n * Returns a
sequence of index ranges of substrings in this char sequence around occurrences of the specified [delimiters].\n *\n *
@param delimiters One or more characters to be used as delimiters.\n * @param startIndex The index to start
searching delimiters from.\n *  No range having its start value less than [startIndex] is returned.\n *  [startIndex] is
coerced
 to be non-negative and not greater than length of this string.\n * @param ignoreCase `true` to ignore character case
when matching a delimiter. By default `false`.\n * @param limit The maximum number of substrings to return. Zero
by default means no limit is set.\n */\nprivate fun CharSequence.rangesDelimitedBy(delimiters: CharArray,
startIndex: Int = 0, ignoreCase: Boolean = false, limit: Int = 0): Sequence<IntRange> {\n
requireNonNegativeLimit(limit)\n\n    return DelimitedRangesSequence(this, startIndex, limit, { currentIndex ->\n
   indexOfAny(delimiters, currentIndex, ignoreCase = ignoreCase).let { if (it < 0) null else it to 1 }\n
})\n}\n\n\n/**\n * Returns a sequence of index ranges of substrings in this char sequence around occurrences of the
specified [delimiters].\n *\n * @param delimiters One or more strings to be used as delimiters.\n * @param
startIndex The index to start searching delimiters from.\n *  No range having its start value less than [startIndex] is
 returned.\n *  [startIndex] is coerced to be non-negative and not greater than length of this string.\n * @param
ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.\n * @param limit The
maximum number of substrings to return. Zero by default means no limit is set.\n *\n * To avoid ambiguous results
when strings in [delimiters] have characters in common, this method proceeds from\n * the beginning to the end of
this string, and finds at each position the first element in [delimiters]\n * that matches this string at that position.\n
*/\nprivate fun CharSequence.rangesDelimitedBy(delimiters: Array<out String>, startIndex: Int = 0, ignoreCase:
Boolean = false, limit: Int = 0): Sequence<IntRange> {\n    requireNonNegativeLimit(limit)\n    val delimitersList =
delimiters.asList()\n\n    return DelimitedRangesSequence(this, startIndex, limit, { currentIndex ->
findAnyOf(delimitersList, currentIndex, ignoreCase = ignoreCase, last = false)?.let { it.first
 to it.second.length } })\n\n}\n\ninternal fun requireNonNegativeLimit(limit: Int) =\n    require(limit >= 0) { \"Limit
must be non-negative, but was $limit\" }\n\n\n// split\n\n/**\n * Splits this char sequence to a sequence of strings
```

around occurrences of the specified [delimiters].\n *\n * @param delimiters One or more strings to be used as delimiters.\n * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.\n * @param limit The maximum number of substrings to return. Zero by default means no limit is set.\n *\n * To avoid ambiguous results when strings in [delimiters] have characters in common, this method proceeds from\n * the beginning to the end of this string, and finds at each position the first element in [delimiters]\n * that matches this string at that position.\n */\npublic fun CharSequence.splitToSequence(vararg delimiters: String, ignoreCase: Boolean = false, limit: Int = 0): Sequence<String> =\n rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).map { substring(it) }\n\n/**\n * Splits this char sequence to a list of strings around occurrences of the specified [delimiters].\n *\n * @param delimiters One or more strings to be used as delimiters.\n * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.\n * @param limit The maximum number of substrings to return. Zero by default means no limit is set.\n *\n * To avoid ambiguous results when strings in [delimiters] have characters in common, this method proceeds from\n * the beginning to the end of this string, and matches at each position the first element in [delimiters]\n * that is equal to a delimiter in this instance at that position.\n */\npublic fun CharSequence.split(vararg delimiters: String, ignoreCase: Boolean = false, limit: Int = 0): List<String> {\n if (delimiters.size == 1) {\n val delimiter = delimiters[0]\n if (!delimiter.isEmpty()) {\n return split(delimiter, ignoreCase, limit)\n }\n }\n\n return rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).asIterable().map { substring(it) }\n}\n\n/**\n * Splits this char sequence to a sequence of strings around occurrences of the specified [delimiters].\n *\n * @param delimiters One or more characters to be used as delimiters.\n * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.\n * @param limit The maximum number of substrings to return.\n */\npublic fun CharSequence.splitToSequence(vararg delimiters: Char, ignoreCase: Boolean = false, limit: Int = 0): Sequence<String> =\n rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).map { substring(it) }\n\n/**\n * Splits this char sequence to a list of strings around occurrences of the specified [delimiters].\n *\n * @param delimiters One or more characters to be used as delimiters.\n * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.\n * @param limit The maximum number of substrings to return.\n */\npublic fun CharSequence.split(vararg delimiters: Char, ignoreCase: Boolean = false, limit: Int = 0): List<String> {\n if (delimiters.size == 1) {\n return split(delimiters[0].toString(), ignoreCase, limit)\n }\n\n return rangesDelimitedBy(delimiters, ignoreCase = ignoreCase, limit = limit).asIterable().map { substring(it) }\n}\n\n/**\n * Splits this char sequence to a list of strings around occurrences of the specified [delimiter].\n * This is specialized version of split which receives single non-empty delimiter and offers better performance\n *\n * @param delimiter String used as delimiter\n * @param ignoreCase `true` to ignore character case when matching a delimiter. By default `false`.\n * @param limit The maximum number of substrings to return.\n */\nprivate fun CharSequence.split(delimiter: String, ignoreCase: Boolean, limit: Int): List<String> {\n requireNonNegativeLimit(limit)\n\n var currentOffset = 0\n var nextIndex = indexOf(delimiter, currentOffset, ignoreCase)\n if (nextIndex == -1 || limit == 1) {\n return listOf(this.toString())\n }\n\n val isLimited = limit > 0\n val result = ArrayList<String>(if (isLimited) limit.coerceAtMost(10) else 10)\n do {\n result.add(substring(currentOffset, nextIndex))\n currentOffset = nextIndex + delimiter.length\n // Do not search for next occurrence if we're reaching limit\n if (isLimited && result.size == limit - 1) break\n nextIndex = indexOf(delimiter, currentOffset, ignoreCase)\n } while (nextIndex != -1)\n\n result.add(substring(currentOffset, length))\n return result\n}\n\n/**\n * Splits this char sequence to a list of strings around matches of the given regular expression.\n *\n * @param limit Non-negative value specifying the maximum number of substrings to return.\n * Zero by default means no limit is set.\n */\n@kotlin.internal.InlineOnly\npublic inline fun CharSequence.split(regex: Regex, limit: Int = 0): List<String> = regex.split(this, limit)\n\n/**\n * Splits this char sequence to a sequence of strings around matches of the given regular expression.\n *\n * @param limit Non-negative value specifying the maximum number of substrings to return.\n * Zero by default means no limit is set.\n * @sample samples.text.Strings.splitToSequence\n

```
*/\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npubli
c inline fun CharSequence.splitToSequence(regex: Regex, limit: Int = 0): Sequence<String> =
regex.splitToSequence(this, limit)\n\n/**\n * Splits this char sequence to a sequence of lines delimited by any of the
following character sequences: CRLF, LF or CR.\n *\n * The lines returned do not include terminating line
separators.\n */\npublic fun CharSequence.lineSequence(): Sequence<String> = splitToSequence(\"\\r\\n\", \"\\n\",
\"\\r\")\n\n/**\n * Splits this char
 sequence to a list of lines delimited by any of the following character sequences: CRLF, LF or CR.\n *\n * The
lines returned do not include terminating line separators.\n */\npublic fun CharSequence.lines(): List<String> =
lineSequence().toList()\n\n/**\n * Returns `true` if the contents of this char sequence are equal to the contents of the
specified [other],\n * i.e. both char sequences contain the same number of the same characters in the same order.\n
*\n * @sample samples.text.Strings.contentEquals\n */\n@SinceKotlin(\"1.5\")\npublic expect infix fun
CharSequence?.contentEquals(other: CharSequence?): Boolean\n\n/**\n * Returns `true` if the contents of this char
sequence are equal to the contents of the specified [other], optionally ignoring case difference.\n *\n * @param
ignoreCase `true` to ignore character case when comparing contents.\n *\n * @sample
samples.text.Strings.contentEquals\n */\n@SinceKotlin(\"1.5\")\npublic expect fun
CharSequence?.contentEquals(other: CharSequence?,
 ignoreCase: Boolean): Boolean\n\ninternal fun CharSequence?.contentEqualsIgnoreCaseImpl(other:
CharSequence?): Boolean {\n    if (this is String && other is String) {\n        return this.equals(other, ignoreCase =
true)\n    }\n\n    if (this === other) return true\n    if (this == null || other == null || this.length != other.length) return
false\n\n    for (i in 0 until length) {\n        if (!this[i].equals(other[i], ignoreCase = true)) {\n            return false\n
}\n    }\n\n    return true\n}\n\ninternal fun CharSequence?.contentEqualsImpl(other: CharSequence?): Boolean {\n
if (this is String && other is String) {\n        return this == other\n    }\n\n    if (this === other) return true\n    if (this
== null || other == null || this.length != other.length) return false\n\n    for (i in 0 until length) {\n        if (this[i] !=
other[i]) {\n            return false\n        }\n    }\n\n    return true\n}\n\n/**\n * Returns `true` if the content of this
string
 is equal to the word \"true\", `false` if it is equal to \"false\",\n * and throws an exception otherwise.\n *\n * There is
also a lenient version of the function available on nullable String, [String?.toBoolean].\n * Note that this function is
case-sensitive.\n *\n * @sample samples.text.Strings.toBooleanStrict\n */\n@SinceKotlin(\"1.5\")\npublic fun
String.toBooleanStrict(): Boolean = when (this) {\n    \"true\" -> true\n    \"false\" -> false\n    else -> throw
IllegalArgumentException(\"The string doesn't represent a boolean value: $this\")\n}\n\n/**\n * Returns `true` if the
content of this string is equal to the word \"true\", `false` if it is equal to \"false\",\n * and `null` otherwise.\n *\n *
There is also a lenient version of the function available on nullable String, [String?.toBoolean].\n * Note that this
function is case-sensitive.\n *\n * @sample samples.text.Strings.toBooleanStrictOrNull\n
*/\n@SinceKotlin(\"1.5\")\npublic fun String.toBooleanStrictOrNull(): Boolean?
 = when (this) {\n    \"true\" -> true\n    \"false\" -> false\n    else -> null\n}","/*\n * Copyright 2010-2023 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT
EDIT!\n\npackage kotlin\n\nimport
kotlin.jvm.*\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@JvmInline\npublic value class
UByteArray\n@PublishedApi\ninternal constructor(@PublishedApi internal val storage: ByteArray) :
Collection<UByte> {\n\n    /** Creates a new array of the specified [size], with all elements initialized to zero. */\n
public constructor(size: Int) : this(ByteArray(size))\n\n    /**\n     * Returns the array element at the given [index].
This method can be called using the index operator.\n     *\n     * If the [index] is out of bounds of this array, throws
an [IndexOutOfBoundsException] except in Kotlin/JS\n     * where the behavior
 is unspecified.\n     */\n    public operator fun get(index: Int): UByte = storage[index].toUByte()\n\n    /**\n     *
Sets the element at the given [index] to the given [value]. This method can be called using the index operator.\n
*\n     * If the [index] is out of bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n
 * where the behavior is unspecified.\n     */\n    public operator fun set(index: Int, value: UByte) {\n
```

storage[index] = value.toByte()\n   }\n\n   /** Returns the number of elements in the array. */\n   public override val size: Int get() = storage.size\n\n   /** Creates an iterator over the elements of the array. */\n   public override operator fun iterator(): kotlin.collections.Iterator<UByte> = Iterator(storage)\n\n   private class Iterator(private val array: ByteArray) : kotlin.collections.Iterator<UByte> {\n   private var index = 0\n   override fun hasNext() = index < array.size\n   override fun next() = if (index < array.size) array[index++].toUByte() else throw NoSuchElementException(index.toString())\n   }\n\n   override fun contains(element: UByte): Boolean {\n   // TODO: Eliminate this check after KT-30016 gets fixed.\n   // Currently JS BE does not generate special bridge method for this method.\n   @Suppress(\"USELESS_CAST\")\n   if ((element as Any?) !is UByte) return false\n\n   return storage.contains(element.toByte())\n   }\n\n   override fun containsAll(elements: Collection<UByte>): Boolean {\n   return (elements as Collection<*>).all { it is UByte && storage.contains(it.toByte()) }\n   }\n\n   override fun isEmpty(): Boolean = this.storage.size == 0\n}\n\n/**\n * Creates a new array of the specified [size], where each element is calculated by calling the specified\n * [init] function.\n *\n * The function [init] is called for each array element sequentially starting from the first one.\n * It should return the value for an array element given its index.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray(size: Int, init: (Int) -> UByte): UByteArray {\n   return UByteArray(ByteArray(size) { index -> init(index).toByte() })\n}\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ubyteArrayOf(vararg elements: UByte): UByteArray = elements\n","/*\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin\n\nimport kotlin.jvm.*\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@JvmInline\npublic value class UIntArray\n@PublishedApi\ninternal constructor(@PublishedApi internal val storage: IntArray) : Collection<UInt> {\n\n   /** Creates a new array of the specified [size], with all elements initialized to zero. */\n   public constructor(size: Int) : this(IntArray(size))\n\n   /**\n   * Returns the array element at the given [index]. This method can be called using the index operator.\n   *\n   * If the [index] is out of bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n   * where the behavior is unspecified.\n   */\n   public operator fun get(index: Int): UInt = storage[index].toUInt()\n\n   /**\n   * Sets the element at the given [index] to the given [value]. This method can be called using the index operator.\n   *\n   * If the [index] is out of bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n   * where the behavior is unspecified.\n   */\n   public operator fun set(index: Int, value: UInt) {\n   storage[index] = value.toInt()\n   }\n\n   /** Returns the number of elements in the array. */\n   public override val size: Int get() = storage.size\n\n   /** Creates an iterator over the elements of the array. */\n   public override operator fun iterator(): kotlin.collections.Iterator<UInt> = Iterator(storage)\n\n   private class Iterator(private val array: IntArray) : kotlin.collections.Iterator<UInt> {\n   private var index = 0\n   override fun hasNext() = index < array.size\n   override fun next() = if (index < array.size) array[index++].toUInt() else throw NoSuchElementException(index.toString())\n   }\n\n   override fun contains(element: UInt): Boolean {\n   // TODO: Eliminate this check after KT-30016 gets fixed.\n   // Currently JS BE does not generate special bridge method for this method.\n   @Suppress(\"USELESS_CAST\")\n   if ((element as Any?) !is UInt) return false\n\n   return storage.contains(element.toInt())\n   }\n\n   override fun containsAll(elements: Collection<UInt>): Boolean {\n   return (elements as Collection<*>).all { it is UInt && storage.contains(it.toInt()) }\n   }\n\n   override fun isEmpty(): Boolean = this.storage.size == 0\n}\n\n/**\n * Creates a new array of the specified [size], where each element is calculated by calling the specified\n * [init] function.\n *\n * The function [init] is called for each array element sequentially starting from the first one.\n * It should return the value for an array element given its index.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun

UIntArray(size: Int, init: (Int) -> UInt): UIntArray {\n   return UIntArray(IntArray(size) { index -> init(index).toInt()\n})\n}\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun uintArrayOf(vararg elements: UInt): UIntArray = elements\n","/*\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin\n\nimport kotlin.jvm.*\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@JvmInline\npublic value class ULongArray\n@PublishedApi\ninternal constructor(@PublishedApi internal val storage: LongArray) : Collection<ULong> {\n\n   /** Creates a new array of the specified [size], with all elements initialized to zero. */\n   public constructor(size: Int) : this(LongArray(size))\n\n   /**\n   * Returns the array element at the given [index]. This method can be called using the index operator.\n   *\n   * If the [index] is out of bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n   * where the behavior is unspecified.\n   */\n   public operator fun get(index: Int): ULong = storage[index].toULong()\n\n   /**\n   * Sets the element at the given [index] to the given [value]. This method can be called using the index operator.\n   *\n   * If the [index] is out of bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n   * where the behavior is unspecified.\n   */\n   public operator fun set(index: Int, value: ULong) {\n   storage[index] = value.toLong()\n   }\n\n   /** Returns the number of elements in the array. */\n   public override val size: Int get() = storage.size\n\n   /** Creates an iterator over the elements of the array. */\n   public override operator fun iterator(): kotlin.collections.Iterator<ULong> = Iterator(storage)\n\n   private class Iterator(private val array: LongArray) : kotlin.collections.Iterator<ULong> {\n   private var index = 0\n   override fun hasNext() = index < array.size\n   override fun next() = if (index < array.size) array[index++].toULong() else throw NoSuchElementException(index.toString())\n   }\n\n   override fun contains(element: ULong): Boolean {\n   // TODO: Eliminate this check after KT-30016 gets fixed.\n   // Currently JS BE does not generate special bridge method for this method.\n   @Suppress(\"USELESS_CAST\")\n   if ((element as Any?) !is ULong) return false\n\n   return storage.contains(element.toLong())\n   }\n\n   override fun containsAll(elements: Collection<ULong>): Boolean {\n   return (elements as Collection<*>).all { it is ULong && storage.contains(it.toLong()) }\n   }\n\n   override fun isEmpty(): Boolean = this.storage.size == 0\n}\n\n/**\n * Creates a new array of the specified [size], where each element is calculated by calling the specified\n * [init] function.\n *\n * The function [init] is called for each array element sequentially starting from the first one.\n * It should return the value for an array element given its index.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray(size: Int, init: (Int) -> ULong): ULongArray {\n   return ULongArray(LongArray(size) { index -> init(index).toLong()\n})\n}\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ulongArrayOf(vararg elements: ULong): ULongArray = elements\n","/*\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin\n\nimport kotlin.jvm.*\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@JvmInline\npublic value class UShortArray\n@PublishedApi\ninternal constructor(@PublishedApi internal val storage: ShortArray) : Collection<UShort> {\n\n   /** Creates a new array of the specified [size], with all elements initialized to zero. */\n   public constructor(size: Int) : this(ShortArray(size))\n\n   /**\n   * Returns the array element at the given [index]. This method can be called using the index operator.\n   *\n   * If the [index] is out of bounds of this array, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n   * where the behavior is unspecified.\n   */\n   public operator fun get(index: Int): UShort = storage[index].toUShort()\n\n   /**\n   * Sets the element at the given [index] to the given [value]. This method can be called using the index operator.\n   *\n   * If the [index] is out of bounds of this array, throws an [IndexOutOfBoundsException] except

in Kotlin/JS\n * where the behavior is unspecified.\n */\n public operator fun set(index: Int, value: UShort) {\n storage[index] = value.toShort()\n }\n\n /** Returns the number of elements in the array. */\n public override val size: Int get() = storage.size\n\n /** Creates an iterator over the elements of the array. */\n public override operator fun iterator(): kotlin.collections.Iterator<UShort> = Iterator(storage)\n\n private class Iterator(private val array: ShortArray) : kotlin.collections.Iterator<UShort> {\n private var index = 0\n override fun hasNext() = index < array.size\n
override fun next() = if (index < array.size) array[index++].toUShort() else throw NoSuchElementException(index.toString())\n }\n\n override fun contains(element: UShort): Boolean {\n // TODO: Eliminate this check after KT-30016 gets fixed.\n // Currently JS BE does not generate special bridge method for this method.\n @Suppress(\"USELESS_CAST\")\n if ((element as Any?) !is UShort) return false\n\n return storage.contains(element.toShort())\n }\n\n override fun containsAll(elements: Collection<UShort>): Boolean {\n return (elements as Collection<*>).all { it is UShort && storage.contains(it.toShort()) }\n }\n\n override fun isEmpty(): Boolean = this.storage.size == 0\n}\n\n/**\n * Creates a new array of the specified [size], where each element is calculated by calling the specified\n * [init] function.\n *\n * The function [init] is called for each array element sequentially starting from the first one.\n * It should return
the value for an array element given its index.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray(size: Int, init: (Int) -> UShort): UShortArray {\n return UShortArray(ShortArray(size) { index -> init(index).toShort()
})\n}\n\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ushortArrayOf(vararg elements: UShort): UShortArray = elements\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"UArraysKt\")\n@file:kotlin.jvm.JvmPackageName(\"kotlin.collections.unsigned\")\n\npackage kotlin.collections\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport
kotlin.random.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UIntArray.component1(): UInt {\n return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun ULongArray.component1(): ULong {\n return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior
is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UByteArray.component1(): UByte {\n return get(0)\n}\n\n/**\n * Returns 1st *element* from the array.\n * \n * If the size of this array is less than 1, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UShortArray.component1(): UShort {\n return get(0)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UIntArray.component2(): UInt {\n return get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If

the size of this array is
less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun ULongArray.component2(): ULong {\n   return get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UByteArray.component2(): UByte {\n   return get(1)\n}\n\n/**\n * Returns 2nd *element* from the array.\n * \n * If the size of this array is less than 2, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UShortArray.component2(): UShort {\n   return
get(1)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UIntArray.component3(): UInt {\n   return get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun ULongArray.component3(): ULong {\n   return get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline operator fun UByteArray.component3(): UByte {\n   return get(2)\n}\n\n/**\n * Returns 3rd *element* from the array.\n * \n * If the size of this array is less than 3, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UShortArray.component3(): UShort {\n   return get(2)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UIntArray.component4(): UInt {\n   return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where
the behavior is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun ULongArray.component4(): ULong {\n   return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UByteArray.component4(): UByte {\n   return get(3)\n}\n\n/**\n * Returns 4th *element* from the array.\n * \n * If the size of this array is less than 4, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UShortArray.component4(): UShort {\n   return get(3)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size
of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UIntArray.component5(): UInt {\n   return get(4)\n}\n\n/**\n * Returns 5th *element* from the

array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun ULongArray.component5(): ULong {\n return get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UByteArray.component5(): UByte
{\n return get(4)\n}\n\n/**\n * Returns 5th *element* from the array.\n * \n * If the size of this array is less than 5, throws an [IndexOutOfBoundsException] except in Kotlin/JS\n * where the behavior is unspecified.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UShortArray.component5(): UShort {\n return get(4)\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic expect fun UIntArray.elementAt(index: Int): UInt\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic expect fun ULongArray.elementAt(index:
Int): ULong\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic expect fun UByteArray.elementAt(index: Int): UByte\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic expect fun UShortArray.elementAt(index: Int): UShort\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun UIntArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UInt): UInt {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.elementAtOrElse(index: Int, defaultValue: (Int) -> ULong): ULong {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.elementAtOrElse(index:
Int, defaultValue: (Int) -> UByte): UByte {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrElse\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.elementAtOrElse(index: Int, defaultValue: (Int) -> UShort): UShort {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample
samples.collections.Collections.Elements.elementAtOrNull\n

*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.elementAtOrNull(index: Int): UInt? {\n
return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.elementAtOrNull(index: Int): ULong? {\n return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.elementAtOrNull(index: Int): UByte? {\n return this.getOrNull(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAtOrNull\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UShortArray.elementAtOrNull(index: Int): UShort? {\n return this.getOrNull(index)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.find(predicate: (UInt) -> Boolean): UInt? {\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.find(predicate: (ULong) -> Boolean): ULong? {\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n
* \n * @sample samples.collections.Collections.Elements.find\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.find(predicate: (UByte) -> Boolean): UByte? {\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.find(predicate: (UShort) -> Boolean): UShort? {\n return firstOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.findLast(predicate: (UInt) -> Boolean): UInt? {\n return lastOrNull(predicate)\n}\n\n/**\n
* Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.findLast(predicate: (ULong) -> Boolean): ULong? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.findLast(predicate: (UByte) -> Boolean): UByte? {\n return lastOrNull(predicate)\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.find\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UShortArray.findLast(predicate: (UShort) -> Boolean): UShort? {\n return
lastOrNull(predicate)\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic

inline fun UIntArray.first(): UInt {\n   return storage.first().toUInt()\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.first(): ULong {\n   return storage.first().toULong()\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.first(): UByte {\n   return storage.first().toUByte()\n}\n\n/**\n * Returns the first element.\n * \n * @throws
 NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.first(): UShort {\n   return storage.first().toUShort()\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.first(predicate: (UInt) -> Boolean): UInt {\n   for (element in this) if (predicate(element)) return element\n   throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.first(predicate: (ULong) -> Boolean): ULong {\n   for (element in this) if
 (predicate(element)) return element\n   throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws
 [NoSuchElementException] if no such element is found.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.first(predicate: (UByte) -> Boolean): UByte {\n   for (element in this) if (predicate(element)) return element\n   throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the first element matching the given [predicate].\n * @throws [NoSuchElementException] if no such element is found.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.first(predicate: (UShort) -> Boolean): UShort {\n   for (element in this) if (predicate(element)) return element\n   throw NoSuchElementException(\"Array contains no
 element matching the predicate.\")\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.firstOrNull(): UInt? {\n   return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.firstOrNull(): ULong? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.firstOrNull(): UByte? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element, or `null` if the array is empty.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.firstOrNull(): UShort? {\n return if (isEmpty()) null else this[0]\n}\n\n/**\n * Returns the first element matching the given [predicate], or
 `null` if element was not found.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.firstOrNull(predicate: (UInt) -> Boolean): UInt? {\n   for (element in this) if (predicate(element)) return element\n   return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.firstOrNull(predicate: (ULong) -> Boolean): ULong? {\n   for (element in this) if (predicate(element)) return element\n   return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun

UByteArray.firstOrNull(predicate: (UByte) -> Boolean): UByte? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns the first element matching the given [predicate], or `null` if element was not found.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.firstOrNull(predicate: (UShort) -> Boolean): UShort? {\n for (element in this) if (predicate(element)) return element\n return null\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.getOrElse(index: Int, defaultValue: (Int) -> UInt): UInt {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.getOrElse(index: Int, defaultValue: (Int) -> ULong): ULong {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.getOrElse(index: Int, defaultValue: (Int) -> UByte): UByte {\n return if (index >= 0 && index <= lastIndex) get(index) else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or the result of calling the [defaultValue] function if the [index] is out of bounds of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.getOrElse(index: Int, defaultValue: (Int) -> UShort): UShort {\n return if (index >= 0 && index <= lastIndex) get(index)
else defaultValue(index)\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.getOrNull(index: Int): UInt? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.getOrNull(index: Int): ULong? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.getOrNull(index:
Int): UByte? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns an element at the given [index] or `null` if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.getOrNull\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.getOrNull(index: Int): UShort? {\n return if (index >= 0 && index <= lastIndex) get(index) else null\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.indexOf(element: UInt): Int {\n return storage.indexOf(element.toInt())\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.indexOf(element: ULong): Int {\n return
storage.indexOf(element.toLong())\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.indexOf(element: UByte): Int {\n return storage.indexOf(element.toByte())\n}\n\n/**\n * Returns first index of [element], or -1 if the array does not contain element.\n

*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.indexOf(element: UShort): Int {\n   return storage.indexOf(element.toShort())\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.indexOfFirst(predicate: (UInt) -> Boolean): Int {\n   return storage.indexOfFirst { predicate(it.toUInt()) }\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.indexOfFirst(predicate: (ULong) -> Boolean): Int {\n   return storage.indexOfFirst { predicate(it.toULong()) }\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.indexOfFirst(predicate: (UByte) -> Boolean): Int {\n   return storage.indexOfFirst { predicate(it.toUByte()) }\n}\n\n/**\n * Returns index of the first element matching the given [predicate], or -1 if the array does not contain such element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.indexOfFirst(predicate: (UShort) -> Boolean): Int {\n   return storage.indexOfFirst { predicate(it.toUShort()) }\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.indexOfLast(predicate: (UInt) -> Boolean): Int {\n   return storage.indexOfLast { predicate(it.toUInt()) }\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.indexOfLast(predicate: (ULong) -> Boolean): Int {\n   return storage.indexOfLast { predicate(it.toULong()) }\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.indexOfLast(predicate: (UByte) -> Boolean): Int {\n   return storage.indexOfLast { predicate(it.toUByte()) }\n}\n\n/**\n * Returns index of the last element matching the given [predicate], or -1 if the array does not contain such element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.indexOfLast(predicate: (UShort) -> Boolean): Int {\n   return storage.indexOfLast { predicate(it.toUShort()) }\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.last(): UInt {\n   return storage.last().toUInt()\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.last(): ULong {\n   return storage.last().toULong()\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.last(): UByte {\n   return storage.last().toUByte()\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.last(): UShort {\n   return storage.last().toUShort()\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample

samples.collections.Collections.Elements.last\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.last(predicate: (UInt) -> Boolean): UInt {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.last(predicate: (ULong) -> Boolean): ULong {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.last(predicate: (UByte) -> Boolean): UByte {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns the last element matching the given [predicate].\n * \n * @throws NoSuchElementException if no such element is found.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.last(predicate: (UShort) -> Boolean): UShort {\n for (index in this.indices.reversed()) {\n val element = this[index]\n
if (predicate(element)) return element\n }\n throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.lastIndexOf(element: UInt): Int {\n return storage.lastIndexOf(element.toInt())\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.lastIndexOf(element: ULong): Int {\n return storage.lastIndexOf(element.toLong())\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.lastIndexOf(element: UByte): Int {\n return storage.lastIndexOf(element.toByte())\n}\n\n/**\n * Returns last index of [element], or -1 if the array does not contain element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.lastIndexOf(element: UShort): Int {\n return storage.lastIndexOf(element.toShort())\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.lastOrNull(): UInt? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.lastOrNull(): ULong? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.lastOrNull(): UByte? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element, or `null` if the array is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.lastOrNull(): UShort? {\n return if (isEmpty()) null else this[size - 1]\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n

*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.lastOrNull(predicate: (UInt) -> Boolean): UInt? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last element
matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.lastOrNull(predicate: (ULong) -> Boolean): ULong? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last element matching the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.lastOrNull(predicate: (UByte) -> Boolean): UByte? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns the last element matching
the given [predicate], or `null` if no such element was found.\n * \n * @sample samples.collections.Collections.Elements.last\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.lastOrNull(predicate: (UShort) -> Boolean): UShort? {\n for (index in this.indices.reversed()) {\n val element = this[index]\n if (predicate(element)) return element\n }\n return null\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.random(): UInt {\n return random(Random)\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.random(): ULong {\n return random(Random)\n}\n\n/**\n
* Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.random(): UByte {\n return random(Random)\n}\n\n/**\n * Returns a random element from this array.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.random(): UShort {\n return random(Random)\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.random(random: Random): UInt {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from
this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.random(random: Random): ULong {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.random(random: Random): UByte {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness.\n * \n * @throws NoSuchElementException if this array is empty.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.random(random: Random): UShort {\n if (isEmpty())\n throw NoSuchElementException(\"Array is empty.\")\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n

*/\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.randomOrNull(): UInt? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.randomOrNull(): ULong? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.randomOrNull(): UByte? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array, or `null` if this array is empty.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.randomOrNull(): UShort? {\n return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun UIntArray.randomOrNull(random: Random): UInt? {\n if (isEmpty())\n return null\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun ULongArray.randomOrNull(random: Random): ULong? {\n if (isEmpty())\n return null\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun UByteArray.randomOrNull(random: Random): UByte? {\n if (isEmpty())\n return null\n return get(random.nextInt(size))\n}\n\n/**\n * Returns a random element from this array using the specified source of randomness, or `null` if this array is empty.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun UShortArray.randomOrNull(random: Random): UShort? {\n if (isEmpty())\n return null\n return get(random.nextInt(size))\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or has more than one element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.single(): UInt {\n return storage.single().toUInt()\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or has more than one element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.single(): ULong {\n return storage.single().toULong()\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or has more than one element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.single(): UByte {\n return storage.single().toUByte()\n}\n\n/**\n * Returns the single element, or throws an exception if the array is empty or has more than one element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.single(): UShort {\n return storage.single().toUShort()\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.single(predicate: (UInt) -> Boolean): UInt {\n var single: UInt? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if (found) throw IllegalArgumentException(\"Array contains more than one matching element.\")\n single = element\n found = true\n }\n }\n if (!found) throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n

@Suppress(\"UNCHECKED_CAST\")\n   return single as UInt\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.single(predicate: (ULong) -> Boolean): ULong {\n   var single: ULong? = null\n   var found = false\n   for (element in this) {\n   if (predicate(element)) {\n   if (found) throw IllegalArgumentException(\"Array contains more than one matching element.\")\n   single = element\n   found = true\n   }\n   }\n   if (!found) throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n   @Suppress(\"UNCHECKED_CAST\")\n   return single as ULong\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.single(predicate: (UByte) -> Boolean): UByte {\n   var single: UByte? = null\n   var found = false\n   for (element in this) {\n   if (predicate(element)) {\n   if (found) throw IllegalArgumentException(\"Array contains more than one matching element.\")\n   single = element\n   found = true\n   }\n   }\n   if (!found) throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n   @Suppress(\"UNCHECKED_CAST\")\n   return single as UByte\n}\n\n/**\n * Returns the single element matching the given [predicate], or throws exception if there is no or more than one matching element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.single(predicate: (UShort) -> Boolean): UShort {\n   var single: UShort? = null\n   var found = false\n   for (element in this) {\n   if (predicate(element)) {\n   if (found) throw IllegalArgumentException(\"Array contains more than one matching element.\")\n   single = element\n   found = true\n   }\n   }\n   if (!found) throw NoSuchElementException(\"Array contains no element matching the predicate.\")\n   @Suppress(\"UNCHECKED_CAST\")\n   return single as UShort\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.singleOrNull(): UInt? {\n   return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.singleOrNull(): ULong? {\n   return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.singleOrNull(): UByte? {\n   return if (size == 1) this[0] else null\n}\n\n/**\n * Returns single element, or `null` if the array is empty or has more than one element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.singleOrNull(): UShort? {\n   return if (size == 1) this[0] else null\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.singleOrNull(predicate: (UInt) -> Boolean): UInt? {\n   var single: UInt? = null\n   var found = false\n   for (element in this) {\n   if (predicate(element)) {\n   if (found) return null\n   single = element\n   found = true\n   }\n   }\n   if (!found) return null\n   return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.singleOrNull(predicate: (ULong) -> Boolean): ULong? {\n   var single: ULong? = null\n   var found = false\n   for (element in this) {\n   if (predicate(element)) {\n   if (found) return null\n   single = element\n   found = true\n   }\n   }\n   if (!found) return null\n   return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.singleOrNull(predicate: (UByte) -> Boolean): UByte? {\n   var single: UByte? = null\n   var found = false\n   for (element in this) {\n   if (predicate(element)) {\n   if (found) return null\n   single =

element\n found = true\n }\n }\n if (!found) return null\n return single\n}\n\n/**\n * Returns the single element matching the given [predicate], or `null` if element was not found or more than one element was found.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.singleOrNull(predicate: (UShort) -> Boolean): UShort? {\n var single: UShort? = null\n var found = false\n for (element in this) {\n if (predicate(element)) {\n if (found) return null\n single = element\n found = true\n }\n }\n if (!found) return null\n return single\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.drop(n: Int): List<UInt> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.drop(n: Int): List<ULong> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.drop(n: Int): List<UByte> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.drop(n: Int): List<UShort> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return takeLast((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.dropLast(n: Int): List<UInt> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.dropLast(n: Int): List<ULong> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.dropLast(n: Int): List<UByte> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n * Returns a list containing all elements except last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.dropLast(n: Int): List<UShort> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n return take((size - n).coerceAtLeast(0))\n}\n\n/**\n

```
 * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.dropLastWhile(predicate: (UInt) -> Boolean): List<UInt> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n            return take(index + 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.dropLastWhile(predicate: (ULong) -> Boolean): List<ULong> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n            return take(index + 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.dropLastWhile(predicate: (UByte) -> Boolean): List<UByte> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n            return take(index + 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n * Returns a list containing all elements except last elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.dropLastWhile(predicate: (UShort) -> Boolean): List<UShort> {\n    for (index in lastIndex downTo 0) {\n        if (!predicate(this[index])) {\n            return take(index + 1)\n        }\n    }\n    return emptyList()\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.dropWhile(predicate: (UInt) -> Boolean): List<UInt> {\n    var yielding = false\n    val list = ArrayList<UInt>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n            list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.dropWhile(predicate: (ULong) -> Boolean): List<ULong> {\n    var yielding = false\n    val list = ArrayList<ULong>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n            list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.dropWhile(predicate: (UByte) -> Boolean): List<UByte> {\n    var yielding = false\n    val list = ArrayList<UByte>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n            list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing all elements except first elements that satisfy the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.drop\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.dropWhile(predicate: (UShort) -> Boolean): List<UShort> {\n    var yielding = false\n    val list = ArrayList<UShort>()\n    for (item in this)\n        if (yielding)\n            list.add(item)\n        else if (!predicate(item)) {\n            list.add(item)\n            yielding = true\n        }\n    return list\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.filter(predicate: (UInt) -> Boolean): List<UInt> {\n    return filterTo(ArrayList<UInt>(),
```

predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.filter(predicate: (ULong) -> Boolean): List<ULong> {\n return filterTo(ArrayList<ULong>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.filter(predicate: (UByte) -> Boolean): List<UByte> {\n return filterTo(ArrayList<UByte>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.filter(predicate: (UShort) -> Boolean): List<UShort> {\n return filterTo(ArrayList<UShort>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.filterIndexed(predicate: (index: Int, UInt) -> Boolean): List<UInt> {\n return filterIndexedTo(ArrayList<UInt>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.filterIndexed(predicate:\n (index: Int, ULong) -> Boolean): List<ULong> {\n return filterIndexedTo(ArrayList<ULong>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.filterIndexed(predicate: (index: Int, UByte) -> Boolean): List<UByte> {\n return filterIndexedTo(ArrayList<UByte>(), predicate)\n}\n\n/**\n * Returns a list containing only elements matching the given [predicate].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexed\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic\n inline fun UShortArray.filterIndexed(predicate: (index: Int, UShort) -> Boolean): List<UShort> {\n return filterIndexedTo(ArrayList<UShort>(), predicate)\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C : MutableCollection<in UInt>> UIntArray.filterIndexedTo(destination: C, predicate: (index: Int, UInt) -> Boolean): C {\n forEachIndexed { index, element ->\n if (predicate(index, element)) destination.add(element)\n }\n return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the\n given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :

MutableCollection<in ULong>> ULongArray.filterIndexedTo(destination: C, predicate: (index: Int, ULong) -> Boolean): C {\n   forEachIndexed { index, element ->\n   if (predicate(index, element)) destination.add(element)\n   }\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C : MutableCollection<in UByte>> UByteArray.filterIndexedTo(destination: C, predicate: (index: Int, UByte) -> Boolean): C {\n   forEachIndexed { index, element ->\n   if (predicate(index, element)) destination.add(element)\n   }\n   return destination\n}\n\n/**\n * Appends all elements matching the given [predicate] to the given [destination].\n * @param [predicate] function that takes the index of an element and the element itself\n * and returns the result of predicate evaluation on the element.\n * \n * @sample samples.collections.Collections.Filtering.filterIndexedTo\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C : MutableCollection<in UShort>> UShortArray.filterIndexedTo(destination: C, predicate: (index: Int, UShort) -> Boolean): C {\n   forEachIndexed { index, element ->\n   if (predicate(index, element)) destination.add(element)\n   }\n   return destination\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.filterNot(predicate: (UInt) -> Boolean): List<UInt> {\n   return filterNotTo(ArrayList<UInt>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.filterNot(predicate: (ULong) -> Boolean): List<ULong> {\n   return filterNotTo(ArrayList<ULong>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.filterNot(predicate: (UByte) -> Boolean): List<UByte> {\n   return filterNotTo(ArrayList<UByte>(), predicate)\n}\n\n/**\n * Returns a list containing all elements not matching the given [predicate].\n * \n * @sample samples.collections.Collections.Filtering.filter\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.filterNot(predicate: (UShort) -> Boolean): List<UShort> {\n   return filterNotTo(ArrayList<UShort>(), predicate)\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C : MutableCollection<in UInt>> UIntArray.filterNotTo(destination: C, predicate: (UInt) -> Boolean): C {\n   for (element in this) if (!predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C : MutableCollection<in ULong>> ULongArray.filterNotTo(destination: C, predicate: (ULong) -> Boolean): C {\n   for (element in this) if (!predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements not matching the given [predicate] to the given [destination].\n * \n * @sample samples.collections.Collections.Filtering.filterTo\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C : MutableCollection<in UByte>> UByteArray.filterNotTo(destination: C, predicate: (UByte) -> Boolean): C {\n   for (element in this) if (!predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all

elements
not matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UShort>> UShortArray.filterNotTo(destination: C, predicate: (UShort) -> Boolean): C {\n
for (element in this) if (!predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends
all elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UInt>> UIntArray.filterTo(destination: C, predicate: (UInt) -> Boolean): C {\n   for (element
in this) if (predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all elements
matching the given [predicate] to the given [destination].\n
* \n * @sample samples.collections.Collections.Filtering.filterTo\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in ULong>> ULongArray.filterTo(destination: C, predicate: (ULong) -> Boolean): C {\n   for
(element in this) if (predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all
elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UByte>> UByteArray.filterTo(destination: C, predicate: (UByte) -> Boolean): C {\n   for
(element in this) if (predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Appends all
elements matching the given [predicate] to the given [destination].\n * \n * @sample
samples.collections.Collections.Filtering.filterTo\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <C :
MutableCollection<in UShort>> UShortArray.filterTo(destination: C, predicate: (UShort) -> Boolean): C {\n   for
(element in this) if (predicate(element)) destination.add(element)\n   return destination\n}\n\n/**\n * Returns a list
containing elements at indices in the specified [indices] range.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.slice(indices: IntRange):
List<UInt> {\n   if (indices.isEmpty()) return listOf()\n   return copyOfRange(indices.start, indices.endInclusive +
1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.slice(indices: IntRange):
List<ULong> {\n   if (indices.isEmpty()) return listOf()\n   return copyOfRange(indices.start, indices.endInclusive
+ 1).asList()\n}\n\n/**\n
* Returns a list containing elements at indices in the specified [indices] range.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.slice(indices: IntRange):
List<UByte> {\n   if (indices.isEmpty()) return listOf()\n   return copyOfRange(indices.start, indices.endInclusive
+ 1).asList()\n}\n\n/**\n * Returns a list containing elements at indices in the specified [indices] range.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.slice(indices: IntRange):
List<UShort> {\n   if (indices.isEmpty()) return listOf()\n   return copyOfRange(indices.start, indices.endInclusive
+ 1).asList()\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.slice(indices: Iterable<Int>):
List<UInt> {\n   val size = indices.collectionSizeOrDefault(10)\n   if (size == 0) return emptyList()\n   val list =
ArrayList<UInt>(size)\n   for (index
in indices) {\n   list.add(get(index))\n   }\n   return list\n}\n\n/**\n * Returns a list containing elements at
specified [indices].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun
ULongArray.slice(indices: Iterable<Int>): List<ULong> {\n   val size = indices.collectionSizeOrDefault(10)\n   if
(size == 0) return emptyList()\n   val list = ArrayList<ULong>(size)\n   for (index in indices) {\n
list.add(get(index))\n   }\n   return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n

*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.slice(indices: Iterable<Int>): List<UByte> {\n   val size = indices.collectionSizeOrDefault(10)\n   if (size == 0) return emptyList()\n   val list = ArrayList<UByte>(size)\n   for (index in indices) {\n   list.add(get(index))\n   }\n   return list\n}\n\n/**\n * Returns a list containing elements at specified [indices].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.slice(indices: Iterable<Int>): List<UShort> {\n   val size = indices.collectionSizeOrDefault(10)\n   if (size == 0) return emptyList()\n   val list = ArrayList<UShort>(size)\n   for (index in indices) {\n   list.add(get(index))\n   }\n   return list\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sliceArray(indices: Collection<Int>): UIntArray {\n   return UIntArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sliceArray(indices: Collection<Int>): ULongArray {\n   return ULongArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sliceArray(indices: Collection<Int>): UByteArray {\n   return UByteArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array containing elements of this array at specified [indices].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sliceArray(indices: Collection<Int>): UShortArray {\n   return UShortArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sliceArray(indices: IntRange): UIntArray {\n   return UIntArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sliceArray(indices: IntRange): ULongArray {\n   return ULongArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sliceArray(indices: IntRange): UByteArray {\n   return UByteArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns an array containing elements at indices in the specified [indices] range.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sliceArray(indices: IntRange): UShortArray {\n   return UShortArray(storage.sliceArray(indices))\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.take(n: Int): List<UInt> {\n   require(n >= 0) { \"Requested element count $n is less than zero.\" }\n   if (n == 0) return emptyList()\n   if (n >= size) return toList()\n   if (n == 1) return listOf(this[0])\n   var count = 0\n   val list = ArrayList<UInt>(n)\n   for (item in this) {\n   list.add(item)\n   if (++count == n)\n   break\n   }\n   return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.take(n: Int): List<ULong> {\n   require(n >= 0) { \"Requested element count $n is less than zero.\" }\n   if (n == 0) return emptyList()\n   if (n >= size) return toList()\n   if (n == 1) return listOf(this[0])\n   var count = 0\n   val list = ArrayList<ULong>(n)\n   for (item in this) {\n   list.add(item)\n   if (++count == n)\n   break\n   }\n   return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic

fun UByteArray.take(n: Int): List<UByte> {\n   require(n >= 0) { \"Requested element count $n is less than zero.\" }\n   if (n == 0) return emptyList()\n   if (n >= size) return toList()\n   if (n == 1) return listOf(this[0])\n   var count = 0\n   val list = ArrayList<UByte>(n)\n   for (item in this) {\n   list.add(item)\n   if (++count == n)\n   break\n   }\n   return list\n}\n\n/**\n * Returns a list containing first [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.take(n: Int): List<UShort> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n   if (n == 0) return emptyList()\n   if (n >= size) return toList()\n   if (n == 1) return listOf(this[0])\n   var count = 0\n   val list = ArrayList<UShort>(n)\n   for (item in this) {\n   list.add(item)\n   if (++count == n)\n   break\n   }\n   return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.takeLast(n: Int): List<UInt> {\n require(n >= 0) { \"Requested element count $n is less than zero.\" }\n   if (n == 0) return emptyList()\n   val size = size\n   if (n >= size) return toList()\n   if (n == 1) return listOf(this[size - 1])\n   val list = ArrayList<UInt>(n)\n   for (index in size - n until size)\n   list.add(this[index])\n   return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic
fun ULongArray.takeLast(n: Int): List<ULong> {\n   require(n >= 0) { \"Requested element count $n is less than zero.\" }\n   if (n == 0) return emptyList()\n   val size = size\n   if (n >= size) return toList()\n   if (n == 1) return listOf(this[size - 1])\n   val list = ArrayList<ULong>(n)\n   for (index in size - n until size)\n   list.add(this[index])\n   return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.takeLast(n: Int): List<UByte> {\n   require(n >= 0) { \"Requested element count $n is less than zero.\" }\n   if (n == 0) return emptyList()\n   val size = size\n   if (n >= size) return toList()\n   if (n == 1) return listOf(this[size - 1])\n   val list = ArrayList<UByte>(n)\n   for (index in size - n until size)\n   list.add(this[index])\n   return list\n}\n\n/**\n * Returns a list containing last [n] elements.\n * \n * @throws IllegalArgumentException if [n] is negative.\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.takeLast(n: Int): List<UShort> {\n   require(n >= 0) { \"Requested element count $n is less than zero.\" }\n   if (n == 0) return emptyList()\n   val size = size\n   if (n >= size) return toList()\n   if (n == 1) return listOf(this[size - 1])\n   val list = ArrayList<UShort>(n)\n   for (index in size - n until size)\n   list.add(this[index])\n   return list\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.takeLastWhile(predicate: (UInt) -> Boolean): List<UInt>
 {\n   for (index in lastIndex downTo 0) {\n   if (!predicate(this[index])) {\n   return drop(index + 1)\n   }\n   }\n   return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.takeLastWhile(predicate: (ULong) -> Boolean): List<ULong> {\n   for (index in lastIndex downTo 0) {\n   if (!predicate(this[index])) {\n   return drop(index + 1)\n   }\n   }\n   return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun

UByteArray.takeLastWhile(predicate: (UByte) -> Boolean): List<UByte> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing last elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.takeLastWhile(predicate: (UShort) -> Boolean): List<UShort> {\n for (index in lastIndex downTo 0) {\n if (!predicate(this[index])) {\n return drop(index + 1)\n }\n }\n return toList()\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.takeWhile(predicate: (UInt) -> Boolean): List<UInt> {\n val list = ArrayList<UInt>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.takeWhile(predicate: (ULong) -> Boolean): List<ULong> {\n val list = ArrayList<ULong>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.takeWhile(predicate: (UByte) -> Boolean): List<UByte> {\n val list = ArrayList<UByte>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Returns a list containing first elements satisfying the given [predicate].\n * \n * @sample samples.collections.Collections.Transformations.take\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.takeWhile(predicate: (UShort) -> Boolean): List<UShort> {\n val list = ArrayList<UShort>()\n for (item in this) {\n if (!predicate(item))\n break\n list.add(item)\n }\n return list\n}\n\n/**\n * Reverses elements in the array in-place.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reverse(): Unit {\n storage.reverse()\n}\n\n/**\n * Reverses elements in the array in-place.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reverse(): Unit {\n storage.reverse()\n}\n\n/**\n * Reverses elements in the array in-place.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reverse(): Unit {\n storage.reverse()\n}\n\n/**\n * Reverses elements in the array in-place.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reverse(): Unit {\n storage.reverse()\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n storage.reverse(fromIndex, toIndex)\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic

inline fun ULongArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n storage.reverse(fromIndex, toIndex)\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n storage.reverse(fromIndex, toIndex)\n}\n\n/**\n * Reverses elements of the array in the specified range in-place.\n * \n * @param fromIndex the start of the range (inclusive) to reverse.\n * @param toIndex the end of the range (exclusive) to reverse.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reverse(fromIndex: Int, toIndex: Int): Unit {\n storage.reverse(fromIndex, toIndex)\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.reversed(): List<UInt> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.reversed(): List<ULong> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.reversed(): List<UByte> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns a list with elements in reversed order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.reversed(): List<UShort> {\n if (isEmpty()) return emptyList()\n val list = toMutableList()\n list.reverse()\n return list\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reversedArray(): UIntArray {\n return UIntArray(storage.reversedArray())\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reversedArray(): ULongArray {\n return ULongArray(storage.reversedArray())\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reversedArray(): UByteArray {\n return UByteArray(storage.reversedArray())\n}\n\n/**\n * Returns an array with elements of this array in reversed order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reversedArray(): UShortArray {\n return UShortArray(storage.reversedArray())\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.shuffle(): Unit {\n shuffle(Random)\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.shuffle(random: Random): Unit {\n for (i in lastIndex downTo 1) {\n val j = random.nextInt(i + 1)\n val copy = this[i]\n this[i] =

this[j]\n   this[j] = copy\n   }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.shuffle(random: Random): Unit {\n   for (i in lastIndex downTo 1) {\n   val j = random.nextInt(i + 1)\n   val copy = this[i]\n   this[i] = this[j]\n   this[j] = copy\n   }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.shuffle(random: Random): Unit {\n   for (i in lastIndex downTo 1) {\n   val j = random.nextInt(i + 1)\n   val copy = this[i]\n   this[i] = this[j]\n   this[j] = copy\n   }\n}\n\n/**\n * Randomly shuffles elements in this array in-place using the specified [random] instance as the source of randomness.\n * \n * See: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle#The_modern_algorithm\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.shuffle(random: Random): Unit {\n   for (i in lastIndex downTo 1) {\n   val j = random.nextInt(i + 1)\n   val copy = this[i]\n   this[i] = this[j]\n   this[j] = copy\n   }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sortDescending(): Unit {\n   if (size > 1) {\n   sort()\n   reverse()\n   }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sortDescending(): Unit {\n   if (size > 1) {\n   sort()\n   reverse()\n   }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sortDescending(): Unit {\n   if (size > 1) {\n   sort()\n   reverse()\n   }\n}\n\n/**\n * Sorts elements in the array in-place descending according to their natural sort order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sortDescending(): Unit {\n   if (size > 1) {\n   sort()\n   reverse()\n   }\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort\n order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sorted(): List<UInt> {\n   return copyOf().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sorted(): List<ULong> {\n   return copyOf().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sorted(): List<UByte> {\n   return copyOf().apply { sort() }.asList()\n}\n\n/**\n * Returns a list of all elements sorted according to their natural sort order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sorted(): List<UShort> {\n   return copyOf().apply { sort() }.asList()\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sortedArray(): UIntArray {\n   if (isEmpty()) return this\n   return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sortedArray(): ULongArray {\n   if (isEmpty()) return this\n   return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sortedArray(): UByteArray {\n   if (isEmpty()) return this\n   return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted according to their natural sort order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sortedArray(): UShortArray {\n   if (isEmpty()) return this\n

```
    return this.copyOf().apply { sort() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sortedArrayDescending(): UIntArray {\n    if (isEmpty()) return this\n    return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sortedArrayDescending(): ULongArray {\n    if (isEmpty()) return this\n    return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sortedArrayDescending(): UByteArray {\n    if (isEmpty()) return this\n    return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns an array with all elements of this array sorted descending according to their natural sort order.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sortedArrayDescending(): UShortArray {\n    if (isEmpty()) return this\n    return this.copyOf().apply { sortDescending() }\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sortedDescending(): List<UInt> {\n    return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sortedDescending(): List<ULong> {\n    return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sortedDescending(): List<UByte> {\n    return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns a list of all elements sorted descending according to their natural sort order.\n * \n * The sort is _stable_. It means that equal elements preserve their order relative to each other after sorting.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sortedDescending(): List<UShort> {\n    return copyOf().apply { sort() }.reversed()\n}\n\n/**\n * Returns an array of type [ByteArray], which is a view of this array where each element is a signed reinterpretation\n * of
    the corresponding element of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.asByteArray(): ByteArray {\n    return storage\n}\n\n/**\n * Returns an array of type [IntArray], which is a view of this array where each element is a signed reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.asIntArray(): IntArray {\n    return storage\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic expect fun UIntArray.asList(): List<UInt>\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic expect fun ULongArray.asList(): List<ULong>\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic expect
    fun UByteArray.asList(): List<UByte>\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic expect fun UShortArray.asList(): List<UShort>\n\n/**\n * Returns an array of type [LongArray], which is a view of this array where each element is a signed reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.asLongArray(): LongArray {\n    return storage\n}\n\n/**\n * Returns an array of type [ShortArray], which is a view of this array where each element is a signed reinterpretation\n * of the corresponding element of this
```

array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.asShortArray(): ShortArray {\n   return storage\n}\n\n/**\n * Returns an array of type [UByteArray], which is a view of this array where each element is an
 unsigned reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.asUByteArray(): UByteArray {\n   return UByteArray(this)\n}\n\n/**\n * Returns an array of type [UIntArray], which is a view of this array where each element is an unsigned reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.asUIntArray(): UIntArray {\n   return UIntArray(this)\n}\n\n/**\n * Returns an array of type [ULongArray], which is a view of this array where each element is an unsigned reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.asULongArray(): ULongArray {\n   return ULongArray(this)\n}\n\n/**\n * Returns an array of type [UShortArray],
 which is a view of this array where each element is an unsigned reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.asUShortArray(): UShortArray {\n   return UShortArray(this)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.3\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@ExperimentalUnsignedTypes\npublic infix fun UIntArray.contentEquals(other: UIntArray): Boolean {\n
 return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation
 warning.\")\n@SinceKotlin(\"1.3\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@ExperimentalUnsignedTypes\npublic infix fun ULongArray.contentEquals(other: ULongArray): Boolean {\n   return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.3\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@ExperimentalUnsignedTypes\npublic infix fun UByteArray.contentEquals(other: UByteArray): Boolean {\n   return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.3\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@ExperimentalUnsignedTypes\npublic
 infix fun UShortArray.contentEquals(other: UShortArray): Boolean {\n   return this.contentEquals(other)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic infix fun UIntArray?.contentEquals(other: UIntArray?): Boolean {\n   return this?.storage.contentEquals(other?.storage)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic infix fun ULongArray?.contentEquals(other: ULongArray?): Boolean {\n   return this?.storage.contentEquals(other?.storage)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements

in the same order.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic infix fun UByteArray?.contentEquals(other: UByteArray?): Boolean {\n   return this?.storage.contentEquals(other?.storage)\n}\n\n/**\n * Returns `true` if the two specified arrays are *structurally* equal to one another,\n * i.e. contain the same number of the same elements in the same order.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic infix fun UShortArray?.contentEquals(other: UShortArray?): Boolean {\n   return this?.storage.contentEquals(other?.storage)\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.3\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.contentHashCode(): Int {\n   return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.3\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.contentHashCode(): Int {\n   return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.3\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.contentHashCode(): Int {\n   return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.3\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.contentHashCode(): Int {\n   return this.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray?.contentHashCode(): Int {\n   return this?.storage.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray?.contentHashCode(): Int {\n return this?.storage.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray?.contentHashCode(): Int {\n   return this?.storage.contentHashCode()\n}\n\n/**\n * Returns a hash code based on the contents of this array as if it is [List].\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray?.contentHashCode(): Int {\n return this?.storage.contentHashCode()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.3\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.contentToString(): String {\n   return this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.3\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.contentToString(): String {\n   return this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.3\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.contentToString(): String {\n   return this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is

[List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@SinceKotlin(\"1.3\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.contentToString(): String {\n return this.contentToString()\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray?.contentToString(): String {\n return this?.joinToString(\", \", \"[\", \"]\") ?: \"null\"\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray?.contentToString(): String {\n return this?.joinToString(\", \", \"[\", \"]\") ?: \"null\"\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray?.contentToString(): String {\n return this?.joinToString(\", \", \"[\", \"]\") ?: \"null\"\n}\n\n/**\n * Returns a string representation of the contents of the specified array as if it is [List].\n * \n * @sample samples.collections.Arrays.ContentOperations.contentToString\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray?.contentToString(): String {\n return this?.joinToString(\", \", \"[\", \"]\") ?: \"null\"\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.copyInto(destination: UIntArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): UIntArray {\n storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy,\n * size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.copyInto(destination: ULongArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): ULongArray {\n storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the

subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.copyInto(destination: UByteArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): UByteArray {\n storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Copies this array or its subrange into the [destination] array and returns that array.\n * \n * It's allowed to pass the same array in the [destination] and even specify the subrange so that it overlaps with the destination range.\n * \n * @param destination the array to copy to.\n * @param destinationOffset the position in the [destination] array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the subrange to copy, 0 by default.\n * @param endIndex the end (exclusive) of the subrange to copy, size of this array by default.\n * \n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n * \n * @return the [destination] array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.copyInto(destination: UShortArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = size): UShortArray {\n storage.copyInto(destination.storage, destinationOffset, startIndex, endIndex)\n return destination\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.copyOf(): UIntArray {\n return UIntArray(storage.copyOf())\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.copyOf(): ULongArray {\n return ULongArray(storage.copyOf())\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.copyOf(): UByteArray {\n return UByteArray(storage.copyOf())\n}\n\n/**\n * Returns new array which is a copy of the original array.\n * \n * @sample samples.collections.Arrays.CopyOfOperations.copyOf\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.copyOf(): UShortArray {\n return UShortArray(storage.copyOf())\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.copyOf(newSize: Int): UIntArray {\n return UIntArray(storage.copyOf(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n

*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.copyOf(newSize: Int): ULongArray
{\n return ULongArray(storage.copyOf(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize] is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.copyOf(newSize: Int): UByteArray {\n return UByteArray(storage.copyOf(newSize))\n}\n\n/**\n * Returns new array which is a copy of the original array, resized to the given [newSize].\n * The copy is either truncated or padded at the end with zero values if necessary.\n * \n * - If [newSize] is less than the size of the original array, the copy array is truncated to the [newSize].\n * - If [newSize]
is greater than the size of the original array, the extra elements in the copy array are filled with zero values.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.copyOf(newSize: Int): UShortArray {\n return UShortArray(storage.copyOf(newSize))\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.copyOfRange(fromIndex: Int, toIndex: Int): UIntArray {\n return UIntArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n
* Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.copyOfRange(fromIndex: Int, toIndex: Int): ULongArray {\n return ULongArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of
this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.copyOfRange(fromIndex: Int, toIndex: Int): UByteArray {\n return UByteArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n * Returns a new array which is a copy of the specified range of the original array.\n * \n * @param fromIndex the start of the range (inclusive) to copy.\n * @param toIndex the end of the range (exclusive) to copy.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.copyOfRange(fromIndex: Int, toIndex: Int): UShortArray {\n return UShortArray(storage.copyOfRange(fromIndex, toIndex))\n}\n\n/**\n
* Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.fill(element: UInt, fromIndex:

Int = 0, toIndex: Int = size): Unit {\n storage.fill(element.toInt(), fromIndex, toIndex)\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.fill(element: ULong, fromIndex: Int = 0, toIndex: Int = size): Unit {\n storage.fill(element.toLong(), fromIndex, toIndex)\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.fill(element: UByte, fromIndex: Int = 0, toIndex: Int = size): Unit {\n storage.fill(element.toByte(), fromIndex, toIndex)\n}\n\n/**\n * Fills this array or its subrange with the specified [element] value.\n * \n * @param fromIndex the start of the range (inclusive) to fill, 0 by default.\n * @param toIndex the end of the range (exclusive) to fill, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.fill(element: UShort, fromIndex: Int = 0, toIndex: Int = size): Unit {\n storage.fill(element.toShort(), fromIndex, toIndex)\n}\n\n/**\n * Returns the range of valid indices for the array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic inline val UIntArray.indices: IntRange\n get() = storage.indices\n\n/**\n * Returns the range of valid indices for the array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic inline val ULongArray.indices: IntRange\n get() = storage.indices\n\n/**\n * Returns the range of valid indices for the array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic inline val UByteArray.indices: IntRange\n get() = storage.indices\n\n/**\n * Returns the range of valid indices for the array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic inline val UShortArray.indices: IntRange\n get() = storage.indices\n\n/**\n * Returns the last valid index for the array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic inline val UIntArray.lastIndex: Int\n get() = storage.lastIndex\n\n/**\n * Returns the last valid index for the array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic inline val ULongArray.lastIndex: Int\n get() = storage.lastIndex\n\n/**\n * Returns the last valid index for the array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic inline val UByteArray.lastIndex: Int\n get() = storage.lastIndex\n\n/**\n * Returns the last valid index for the array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic inline val UShortArray.lastIndex: Int\n get() = storage.lastIndex\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UIntArray.plus(element: UInt): UIntArray {\n return UIntArray(storage + element.toInt())\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun ULongArray.plus(element: ULong): ULongArray {\n return ULongArray(storage + element.toLong())\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UByteArray.plus(element: UByte): UByteArray {\n return UByteArray(storage + element.toByte())\n}\n\n/**\n * Returns an array containing all elements of the original array and then the given [element].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UShortArray.plus(element: UShort): UShortArray {\n return UShortArray(storage +

element.toShort())\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic operator fun UIntArray.plus(elements: Collection<UInt>): UIntArray {\n   var index = size\n   val result = storage.copyOf(size + elements.size)\n   for (element in elements) result[index++] = element.toInt()\n   return UIntArray(result)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic operator fun ULongArray.plus(elements: Collection<ULong>): ULongArray {\n   var index = size\n   val result = storage.copyOf(size + elements.size)\n   for (element in elements) result[index++] = element.toLong()\n   return ULongArray(result)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic operator fun UByteArray.plus(elements: Collection<UByte>): UByteArray {\n   var index = size\n   val result = storage.copyOf(size + elements.size)\n   for (element in elements) result[index++] = element.toByte()\n   return UByteArray(result)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] collection.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic operator fun UShortArray.plus(elements: Collection<UShort>): UShortArray {\n   var index = size\n   val result = storage.copyOf(size + elements.size)\n   for (element in elements) result[index++] = element.toShort()\n   return UShortArray(result)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UIntArray.plus(elements: UIntArray): UIntArray {\n   return UIntArray(storage + elements.storage)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun ULongArray.plus(elements: ULongArray): ULongArray {\n   return ULongArray(storage + elements.storage)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UByteArray.plus(elements: UByteArray): UByteArray {\n   return UByteArray(storage + elements.storage)\n}\n\n/**\n * Returns an array containing all elements of the original array and then all elements of the given [elements] array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline operator fun UShortArray.plus(elements: UShortArray): UShortArray {\n   return UShortArray(storage + elements.storage)\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sort(): Unit {\n   if (size > 1) sortArray(this, 0, size)\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sort(): Unit {\n   if (size > 1) sortArray(this, 0, size)\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sort(): Unit {\n   if (size > 1) sortArray(this, 0, size)\n}\n\n/**\n * Sorts the array in-place.\n * \n * @sample samples.collections.Arrays.Sorting.sortArray\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sort(): Unit {\n   if (size > 1) sortArray(this, 0, size)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if

[fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n sortArray(this, fromIndex, toIndex)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n sortArray(this, fromIndex, toIndex)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive) to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n sortArray(this, fromIndex, toIndex)\n}\n\n/**\n * Sorts a range in the array in-place.\n * \n * @param fromIndex the start of the range (inclusive) to sort, 0 by default.\n * @param toIndex the end of the range (exclusive)
to sort, size of this array by default.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n * \n * @sample samples.collections.Arrays.Sorting.sortRangeOfArray\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sort(fromIndex: Int = 0, toIndex: Int = size): Unit {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, size)\n sortArray(this, fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException
if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range
in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Sorts elements of the array in the specified range in-place.\n * The elements are sorted descending according to their natural sort order.\n * \n * @param fromIndex the start of the range (inclusive) to sort.\n * @param toIndex the end of the range (exclusive) to sort.\n * \n * @throws IndexOutOfBoundsException if [fromIndex] is less than zero or [toIndex] is

greater than the size of this array.\n * @throws IllegalArgumentException if [fromIndex] is greater than [toIndex].\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.sortDescending(fromIndex: Int, toIndex: Int): Unit {\n sort(fromIndex, toIndex)\n reverse(fromIndex, toIndex)\n}\n\n/**\n * Returns an array of type [ByteArray], which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.toByteArray(): ByteArray {\n return storage.copyOf()\n}\n\n/**\n * Returns an array of type [IntArray], which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.toIntArray(): IntArray {\n return storage.copyOf()\n}\n\n/**\n * Returns an array of type [LongArray], which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.toLongArray(): LongArray {\n return storage.copyOf()\n}\n\n/**\n * Returns an array of type [ShortArray], which is a copy of this array where each element is a signed reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.toShortArray(): ShortArray {\n return storage.copyOf()\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.toTypedArray(): Array<UInt> {\n return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.toTypedArray(): Array<ULong> {\n return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.toTypedArray(): Array<UByte> {\n return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns a *typed* object array containing all of the elements of this primitive array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.toTypedArray(): Array<UShort> {\n return Array(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of UByte containing all of the elements of this generic array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun Array<out UByte>.toUByteArray(): UByteArray {\n return UByteArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of type [UByteArray], which is a copy of this array where each element is an unsigned reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ByteArray.toUByteArray(): UByteArray {\n return UByteArray(this.copyOf())\n}\n\n/**\n * Returns an array of UInt containing all of the elements of this generic array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun Array<out UInt>.toUIntArray(): UIntArray {\n return UIntArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of type [UIntArray], which is a copy of this array where each element is an unsigned reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun IntArray.toUIntArray(): UIntArray {\n return UIntArray(this.copyOf())\n}\n\n/**\n * Returns an array of ULong containing all of the elements of this generic array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun Array<out ULong>.toULongArray(): ULongArray {\n return ULongArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of type [ULongArray], which is a copy of this array where each element is an unsigned reinterpretation\n * of the corresponding element of this array.\n

*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun LongArray.toULongArray(): ULongArray {\n   return ULongArray(this.copyOf())\n}\n\n/**\n * Returns an array of UShort containing all of the elements of this generic array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun Array<out UShort>.toUShortArray(): UShortArray {\n   return UShortArray(size) { index -> this[index] }\n}\n\n/**\n * Returns an array of type [UShortArray], which is a copy of this array where each element is an unsigned reinterpretation\n * of the corresponding element of this array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ShortArray.toUShortArray(): UShortArray {\n   return UShortArray(this.copyOf())\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V> UIntArray.associateWith(valueSelector: (UInt) -> V): Map<UInt, V> {\n   val result = LinkedHashMap<UInt, V>(mapCapacity(size).coerceAtLeast(16))\n   return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V> ULongArray.associateWith(valueSelector: (ULong) -> V): Map<ULong, V> {\n   val result = LinkedHashMap<ULong, V>(mapCapacity(size).coerceAtLeast(16))\n   return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V> UByteArray.associateWith(valueSelector: (UByte) -> V): Map<UByte, V> {\n   val result = LinkedHashMap<UByte, V>(mapCapacity(size).coerceAtLeast(16))\n   return associateWithTo(result, valueSelector)\n}\n\n/**\n * Returns a [Map] where keys are elements from the given array and values are\n * produced by the [valueSelector] function applied to each element.\n * \n * If any two elements are equal, the last one gets added to the map.\n * \n * The returned map preserves the entry iteration order of the original array.\n * \n * @sample samples.collections.Collections.Transformations.associateWith\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V> UShortArray.associateWith(valueSelector: (UShort) -> V): Map<UShort, V> {\n   val result = LinkedHashMap<UShort, V>(mapCapacity(size).coerceAtLeast(16))\n   return associateWithTo(result, valueSelector)\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n * @sample samples.collections.Collections.Transformations.associateWithTo\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in UInt, in V>> UIntArray.associateWithTo(destination: M, valueSelector: (UInt) -> V): M {\n   for (element in this) {\n   destination.put(element, valueSelector(element))\n   }\n   return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element itself and value is provided

by the [valueSelector] function applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n * @sample samples.collections.Collections.Transformations.associateWithTo\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in ULong, in V>> ULongArray.associateWithTo(destination: M, valueSelector: (ULong) -> V): M {\n for (element in this) {\n destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n * @sample samples.collections.Collections.Transformations.associateWithTo\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in UByte, in V>> UByteArray.associateWithTo(destination: M, valueSelector: (UByte) -> V): M {\n for (element in this) {\n destination.put(element, valueSelector(element))\n }\n return destination\n}\n\n/**\n * Populates and returns the [destination] mutable map with key-value pairs for each element of the given array,\n * where key is the element itself and value is provided by the [valueSelector] function applied to that key.\n * \n * If any two elements are equal, the last one overwrites the former value in the map.\n * \n * @sample samples.collections.Collections.Transformations.associateWithTo\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V, M : MutableMap<in UShort, in V>> UShortArray.associateWithTo(destination: M, valueSelector: (UShort) -> V): M {\n for (element in this) {\n destination.put(element,\n valueSelector(element))\n }\n return destination\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.flatMap(transform: (UInt) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.flatMap(transform: (ULong) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded\n from results of [transform] function being invoked on each element of original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.flatMap(transform: (UByte) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element of original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMap\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.flatMap(transform: (UShort) -> Iterable<R>): List<R> {\n return flatMapTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.flatMapIndexed(transform: (index: Int, UInt) -> Iterable<R>): List<R> {\n return flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n * @sample samples.collections.Collections.Transformations.flatMapIndexed\n

*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
ULongArray.flatMapIndexed(transform: (index: Int, ULong) -> Iterable<R>): List<R> {\n   return
flatMapIndexedTo(ArrayList<R>(),
 transform)\n}\n\n/**\n * Returns a single list of all elements yielded from results of [transform] function being
invoked on each element\n * and its index in the original array.\n * \n * @sample
samples.collections.Collections.Transformations.flatMapIndexed\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UByteArray.flatMapIndexed(transform: (index: Int, UByte) -> Iterable<R>): List<R> {\n   return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Returns a single list of all elements yielded from
results of [transform] function being invoked on each element\n * and its index in the original array.\n * \n *
@sample samples.collections.Collections.Transformations.flatMapIndexed\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
 inline fun <R> UShortArray.flatMapIndexed(transform: (index: Int, UShort) -> Iterable<R>): List<R> {\n   return
flatMapIndexedTo(ArrayList<R>(), transform)\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UIntArray.flatMapIndexedTo(destination: C, transform: (index: Int, UInt) ->
Iterable<R>): C {\n   var index = 0\n   for (element in this) {\n   val list = transform(index++, element)\n
destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and
 its index in the original array, to the given [destination].\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> ULongArray.flatMapIndexedTo(destination: C, transform: (index: Int, ULong) ->
Iterable<R>): C {\n   var index = 0\n   for (element in this) {\n   val list = transform(index++, element)\n
destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked on each element\n * and its index in the original array, to the given
[destination].\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UByteArray.flatMapIndexedTo(destination:
 C, transform: (index: Int, UByte) -> Iterable<R>): C {\n   var index = 0\n   for (element in this) {\n   val list =
transform(index++, element)\n   destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Appends all
elements yielded from results of [transform] function being invoked on each element\n * and its index in the original
array, to the given [destination].\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :
MutableCollection<in R>> UShortArray.flatMapIndexedTo(destination: C, transform: (index: Int, UShort) ->
Iterable<R>): C {\n   var index = 0\n   for (element in this) {\n   val list = transform(index++, element)\n
destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Appends all elements yielded from results of
[transform] function being invoked
 on each element of original array, to the given [destination].\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :

MutableCollection<in R>> UIntArray.flatMapTo(destination: C, transform: (UInt) -> Iterable<R>): C {\n   for (element in this) {\n   val list = transform(element)\n   destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> ULongArray.flatMapTo(destination: C, transform: (ULong) -> Iterable<R>): C {\n   for (element in this) {\n   val list = transform(element)\n   destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> UByteArray.flatMapTo(destination: C, transform: (UByte) -> Iterable<R>): C {\n   for (element in this) {\n   val list = transform(element)\n   destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Appends all elements yielded from results of [transform] function being invoked on each element of original array, to the given [destination].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> UShortArray.flatMapTo(destination: C, transform: (UShort) -> Iterable<R>): C {\n   for (element in this) {\n   val list = transform(element)\n   destination.addAll(list)\n   }\n   return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K> UIntArray.groupBy(keySelector: (UInt) -> K): Map<K, List<UInt>> {\n   return groupByTo(LinkedHashMap<K, MutableList<UInt>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K> ULongArray.groupBy(keySelector: (ULong) -> K): Map<K, List<ULong>> {\n   return groupByTo(LinkedHashMap<K, MutableList<ULong>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K> UByteArray.groupBy(keySelector: (UByte) -> K): Map<K, List<UByte>> {\n   return groupByTo(LinkedHashMap<K, MutableList<UByte>>(), keySelector)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector]\n function\n * applied to each element and returns a map where each group key is associated with a list of corresponding elements.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K> UShortArray.groupBy(keySelector: (UShort) -> K): Map<K, List<UShort>> {\n   return groupByTo(LinkedHashMap<K, MutableList<UShort>>(), keySelector)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n

*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> UIntArray.groupBy(keySelector: (UInt) -> K, valueTransform: (UInt) -> V): Map<K, List<V>> {\n return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> ULongArray.groupBy(keySelector: (ULong) -> K, valueTransform: (ULong) -> V): Map<K, List<V>> {\n return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> UByteArray.groupBy(keySelector: (UByte) -> K, valueTransform: (UByte) -> V): Map<K, List<V>> {\n return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and returns a map where each group key is associated with a list of corresponding values.\n * \n * The returned map preserves the entry iteration order of the keys produced from the original array.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V> UShortArray.groupBy(keySelector: (UShort) -> K, valueTransform: (UShort) -> V): Map<K, List<V>> {\n return groupByTo(LinkedHashMap<K, MutableList<V>>(), keySelector, valueTransform)\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M : MutableMap<in K, MutableList<UInt>>> UIntArray.groupByTo(destination: M, keySelector: (UInt) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<UInt>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M : MutableMap<in K, MutableList<ULong>>> ULongArray.groupByTo(destination: M, keySelector: (ULong) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<ULong>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M : MutableMap<in K, MutableList<UByte>>> UByteArray.groupByTo(destination: M, keySelector: (UByte) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<UByte>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups elements

of the original array by the key returned by the given [keySelector] function\n * applied to each element and puts to the [destination] map each group key associated with a list of corresponding elements.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupBy\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, M : MutableMap<in K, MutableList<UShort>>> UShortArray.groupByTo(destination: M, keySelector: (UShort) -> K): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<UShort>() }\n list.add(element)\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, M : MutableMap<in K, MutableList<V>>> UIntArray.groupByTo(destination: M, keySelector: (UInt) -> K, valueTransform: (UInt) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, M : MutableMap<in K, MutableList<V>>> ULongArray.groupByTo(destination: M, keySelector: (ULong) -> K, valueTransform: (ULong) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, M : MutableMap<in K, MutableList<V>>> UByteArray.groupByTo(destination: M, keySelector: (UByte) -> K, valueTransform: (UByte) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Groups values returned by the [valueTransform] function applied to each element of the original array\n * by the key returned by the given [keySelector] function applied to the element\n * and puts to the [destination] map each group key associated with a list of corresponding values.\n * \n * @return The [destination] map.\n * \n * @sample samples.collections.Collections.Transformations.groupByKeysAndValues\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <K, V, M : MutableMap<in K, MutableList<V>>> UShortArray.groupByTo(destination: M, keySelector: (UShort) -> K, valueTransform: (UShort) -> V): M {\n for (element in this) {\n val key = keySelector(element)\n val list = destination.getOrPut(key) { ArrayList<V>() }\n list.add(valueTransform(element))\n }\n return destination\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.map(transform: (UInt) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>

ULongArray.map(transform:
(ULong) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.map(transform: (UByte) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element in the original array.\n * \n * @sample samples.collections.Collections.Transformations.map\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.map(transform: (UShort) -> R): List<R> {\n return mapTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given
[transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.mapIndexed(transform: (index: Int, UInt) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.mapIndexed(transform: (index: Int, ULong) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size),
transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.mapIndexed(transform: (index: Int, UByte) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Returns a list containing the results of applying the given [transform] function\n * to each element and its index in the original array.\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.mapIndexed(transform:
(index: Int, UShort) -> R): List<R> {\n return mapIndexedTo(ArrayList<R>(size), transform)\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> UIntArray.mapIndexedTo(destination: C, transform: (index: Int, UInt) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element
itself\n * and returns the result of the transform applied to the element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> ULongArray.mapIndexedTo(destination: C, transform: (index: Int, ULong) -> R): C {\n var index = 0\n for (item in this)\n destination.add(transform(index++, item))\n return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C :

```
MutableCollection<in R>> UByteArray.mapIndexedTo(destination: C, transform: (index: Int, UByte) -> R): C {\n var index = 0\n    for (item in this)\n
    destination.add(transform(index++, item))\n    return destination\n}\n\n/**\n * Applies the given [transform] function to each element and its index in the original array\n * and appends the results to the given [destination].\n * @param [transform] function that takes the index of an element and the element itself\n * and returns the result of the transform applied to the element.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> UShortArray.mapIndexedTo(destination: C, transform: (index: Int, UShort) -> R): C {\n var index = 0\n    for (item in this)\n        destination.add(transform(index++, item))\n    return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in
R>> UIntArray.mapTo(destination: C, transform: (UInt) -> R): C {\n    for (item in this)\n
destination.add(transform(item))\n    return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> ULongArray.mapTo(destination: C, transform: (ULong) -> R): C {\n    for (item in this)\n        destination.add(transform(item))\n    return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> UByteArray.mapTo(destination: C, transform: (UByte) -> R): C {\n    for (item in this)\n
    destination.add(transform(item))\n
   return destination\n}\n\n/**\n * Applies the given [transform] function to each element of the original array\n * and appends the results to the given [destination].\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, C : MutableCollection<in R>> UShortArray.mapTo(destination: C, transform: (UShort) -> R): C {\n    for (item in this)\n        destination.add(transform(item))\n    return destination\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.withIndex(): Iterable<IndexedValue<UInt>> {\n    return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic
 fun ULongArray.withIndex(): Iterable<IndexedValue<ULong>> {\n    return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.withIndex(): Iterable<IndexedValue<UByte>> {\n    return IndexingIterable { iterator() }\n}\n\n/**\n * Returns a lazy [Iterable] that wraps each element of the original array\n * into an [IndexedValue] containing the index of that element and the element itself.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.withIndex(): Iterable<IndexedValue<UShort>> {\n    return IndexingIterable { iterator() }\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the array contains no elements, the function returns `true`\n
 * because there are no elements in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in [\"Vacuous truth\"](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.all(predicate: (UInt) -> Boolean): Boolean {\n    for (element in this) if (!predicate(element)) return
```

false\n   return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the array contains no elements, the function returns `true`\n * because there are no elements in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in [\"Vacuous truth\"](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.all(predicate: (ULong) -> Boolean): Boolean {\n   for (element in this) if (!predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the array contains no elements, the function returns `true`\n * because there are no elements in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in [\"Vacuous truth\"](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.all(predicate: (UByte) -> Boolean): Boolean {\n   for (element in this) if (!predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if all elements match the given [predicate].\n * \n * Note that if the array contains no elements, the function returns `true`\n * because there are no elements in it that _do not_ match the predicate.\n * See a more detailed explanation of this logic concept in [\"Vacuous truth\"](https://en.wikipedia.org/wiki/Vacuous_truth) article.\n * \n * @sample samples.collections.Collections.Aggregates.all\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.all(predicate: (UShort) -> Boolean): Boolean {\n   for (element in this) if (!predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.any(): Boolean {\n   return storage.any()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.any(): Boolean {\n   return storage.any()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.any(): Boolean {\n   return storage.any()\n}\n\n/**\n * Returns `true` if array has at least one element.\n * \n * @sample samples.collections.Collections.Aggregates.any\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.any(): Boolean {\n   return storage.any()\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.any(predicate: (UInt) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return true\n   return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.any(predicate: (ULong) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return true\n   return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.any(predicate: (UByte) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return true\n   return false\n}\n\n/**\n * Returns `true` if at least one element matches the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.anyWithPredicate\n

```
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.any(predicate: (UShort) -> Boolean): Boolean {\n    for (element in this) if (predicate(element)) return true\n    return false\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.count(predicate: (UInt) -> Boolean): Int {\n    var count = 0\n    for (element in this) if (predicate(element)) ++count\n    return count\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.count(predicate: (ULong) -> Boolean): Int {\n    var count = 0\n    for (element in this) if (predicate(element)) ++count\n    return count\n}\n\n/**\n * Returns the number of elements
 matching the given [predicate].\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.count(predicate: (UByte) -> Boolean): Int {\n    var count = 0\n    for (element in this) if (predicate(element)) ++count\n    return count\n}\n\n/**\n * Returns the number of elements matching the given [predicate].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.count(predicate: (UShort) -> Boolean): Int {\n    var count = 0\n    for (element in this) if (predicate(element)) ++count\n    return count\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
 inline fun <R> UIntArray.fold(initial: R, operation: (acc: R, UInt) -> R): R {\n    var accumulator = initial\n    for (element in this) accumulator = operation(accumulator, element)\n    return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.fold(initial: R, operation: (acc: R, ULong) -> R): R {\n    var accumulator = initial\n    for (element in this) accumulator = operation(accumulator, element)\n    return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value
 and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.fold(initial: R, operation: (acc: R, UByte) -> R): R {\n    var accumulator = initial\n    for (element in this) accumulator = operation(accumulator, element)\n    return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
 inline fun <R> UShortArray.fold(initial: R, operation: (acc: R, UShort) -> R): R {\n    var accumulator = initial\n    for (element in this) accumulator = operation(accumulator, element)\n    return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): R {\n    var index = 0\n    var accumulator = initial\n    for (element in this) accumulator = operation(index++,
```

accumulator, element)\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.foldIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): R {\n   var index = 0\n   var accumulator = initial\n   for (element in this) accumulator = operation(index++, accumulator, element)\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index
in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): R {\n   var index = 0\n   var accumulator = initial\n   for (element in this) accumulator = operation(index++, accumulator, element)\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself,
and calculates the next accumulator value.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.foldIndexed(initial: R, operation: (index: Int, acc: R, UShort) -> R): R {\n   var index = 0\n   var accumulator = initial\n   for (element in this) accumulator = operation(index++, accumulator, element)\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.foldRight(initial: R, operation: (UInt, acc: R) -> R): R {\n   var index = lastIndex\n   var accumulator = initial\n
while (index >= 0) {\n   accumulator = operation(get(index--), accumulator)\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.foldRight(initial: R, operation: (ULong, acc: R) -> R): R {\n   var index = lastIndex\n   var accumulator = initial\n   while (index >= 0) {\n   accumulator = operation(get(index--), accumulator)\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns
the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.foldRight(initial: R, operation: (UByte, acc: R) -> R): R {\n   var index = lastIndex\n   var accumulator = initial\n   while (index >= 0) {\n   accumulator = operation(get(index--), accumulator)\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Returns the specified [initial] value if the array is

empty.\n * \n * @param [operation] function that takes an element and current accumulator value, and calculates the next accumulator value.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.foldRight(initial: R, operation: (UShort, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.foldRightIndexed(initial: R, operation: (index: Int, UInt, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.foldRightIndexed(initial: R, operation: (index: Int, ULong, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation]
from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.foldRightIndexed(initial: R, operation: (index: Int, UByte, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with [initial] value and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns the specified [initial] value if the array is empty.\n
* \n * @param [operation] function that takes the index of an element, the element itself\n * and current accumulator value, and calculates the next accumulator value.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.foldRightIndexed(initial: R, operation: (index: Int, UShort, acc: R) -> R): R {\n var index = lastIndex\n var accumulator = initial\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Performs the given [action] on each element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.forEach(action: (UInt) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.forEach(action:
(ULong) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.forEach(action: (UByte) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n * Performs the given [action] on each element.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.forEach(action: (UShort) -> Unit): Unit {\n for (element in this) action(element)\n}\n\n/**\n *

Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.forEachIndexed(action: (index: Int, UInt) -> Unit): Unit {\n var index = 0\n for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.forEachIndexed(action: (index: Int, ULong) -> Unit): Unit {\n var index = 0\n for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.forEachIndexed(action: (index: Int, UByte) -> Unit): Unit {\n var index = 0\n for (item in this) action(index++, item)\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.forEachIndexed(action: (index: Int, UShort) -> Unit): Unit {\n var index = 0\n for (item in this) action(index++, item)\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow-U\")\n@ExperimentalUnsignedTypes\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun UIntArray.max(): UInt {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow-U\")\n@ExperimentalUnsignedTypes\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun ULongArray.max(): ULong {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow-U\")\n@ExperimentalUnsignedTypes\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun UByteArray.max(): UByte {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the largest element.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxOrThrow-U\")\n@ExperimentalUnsignedTypes\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun UShortArray.max(): UShort {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow-U\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <R : Comparable<R>> UIntArray.maxBy(selector: (UInt) -> R): UInt {\n if (isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n

for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow-U\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <R : Comparable<R>> ULongArray.maxBy(selector: (ULong) -> R): ULong {\n if (isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow-U\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <R : Comparable<R>> UByteArray.maxBy(selector: (UByte) -> R): UByte {\n if (isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.maxBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxByOrThrow-U\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <R : Comparable<R>> UShortArray.maxBy(selector: (UShort) -> R): UShort {\n if (isEmpty()) throw NoSuchElementException()\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UIntArray.maxByOrNull(selector: (UInt) -> R): UInt? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ULongArray.maxByOrNull(selector: (ULong) -> R): ULong? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue = selector(maxElem)\n for (i in 1..lastIndex) {\n val e = this[i]\n val v = selector(e)\n if (maxValue < v) {\n maxElem = e\n maxValue = v\n }\n }\n return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UByteArray.maxByOrNull(selector: (UByte) -> R): UByte? {\n if (isEmpty()) return null\n var maxElem = this[0]\n val lastIndex = this.lastIndex\n if (lastIndex == 0) return maxElem\n var maxValue =

selector(maxElem)\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   val v = selector(e)\n   if (maxValue < v) {\n   maxElem
 = e\n   maxValue = v\n   }\n   }\n   return maxElem\n}\n\n/**\n * Returns the first element yielding the largest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.maxByOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UShortArray.maxByOrNull(selector: (UShort) -> R): UShort? {\n   if (isEmpty()) return null\n   var maxElem = this[0]\n   val lastIndex = this.lastIndex\n   if (lastIndex == 0) return maxElem\n   var maxValue = selector(maxElem)\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   val v = selector(e)\n   if (maxValue < v) {\n   maxElem = e\n   maxValue = v\n   }\n   }\n   return maxElem\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector]
 function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.maxOf(selector: (UInt) -> Double): Double {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
 inline fun ULongArray.maxOf(selector: (ULong) -> Double): Double {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.maxOf(selector: (UByte) -> Double): Double {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue =
 selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.maxOf(selector: (UShort) -> Double): Double {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector]
 function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.maxOf(selector: (UInt) -> Float): Float {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue =

maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.maxOf(selector: (ULong) -> Float): Float {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.maxOf(selector: (UByte)
-> Float): Float {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.maxOf(selector: (UShort) -> Float): Float {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   maxValue = maxOf(maxValue, v)\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UIntArray.maxOf(selector: (UInt) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (maxValue < v) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ULongArray.maxOf(selector: (ULong) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (maxValue < v) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UByteArray.maxOf(selector: (UByte) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (maxValue < v) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution

ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UShortArray.maxOf(selector: (UShort) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.maxOfOrNull(selector: (UInt) -> Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.maxOfOrNull(selector: (ULong) -> Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.maxOfOrNull(selector: (UByte) -> Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.maxOfOrNull(selector: (UShort) -> Double): Double? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.maxOfOrNull(selector: (UInt) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.maxOfOrNull(selector: (ULong) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of

values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.maxOfOrNull(selector: (UByte) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n
}\n return
maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.maxOfOrNull(selector: (UShort) -> Float): Float? {\n if (isEmpty()) return null\n var maxValue =
selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n maxValue = maxOf(maxValue, v)\n
}\n return maxValue\n}\n\n/**\n * Returns the largest value among all values produced by [selector] function\n *
applied to each element in the array or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> UIntArray.maxOfOrNull(selector: (UInt) -> R): R? {\n if (isEmpty()) return
null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value
among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no
elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> ULongArray.maxOfOrNull(selector: (ULong) -> R): R? {\n if (isEmpty()) return null\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n
if (maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest
value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are
no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R :
Comparable<R>> UByteArray.maxOfOrNull(selector: (UByte) -> R): R? {\n if (isEmpty()) return null\n var
maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (maxValue < v) {\n
maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value among all values
produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> UShortArray.maxOfOrNull(selector: (UShort) -> R): R? {\n if (isEmpty())
return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(maxValue < v) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest value
according to the provided [comparator]\n * among all values produced by [selector] function applied to each
element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.maxOfWith(comparator: Comparator<in R>, selector: (UInt) -> R): R {\n if (isEmpty()) throw
NoSuchElementException()\n
var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if
(comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n *

Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.maxOfWith(comparator: Comparator<in R>, selector: (ULong) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.maxOfWith(comparator: Comparator<in R>, selector: (UByte) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.maxOfWith(comparator: Comparator<in R>, selector: (UShort) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (UInt) -> R): R? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (ULong) -> R): R? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (UByte) -> R): R? {\n   if (isEmpty()) return null\n   var maxValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(maxValue, v) < 0) {\n   maxValue = v\n   }\n   }\n   return maxValue\n}\n\n/**\n * Returns the largest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>

UShortArray.maxOfWithOrNull(comparator: Comparator<in R>, selector: (UShort) -> R): R? {\n if (isEmpty()) return null\n var maxValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(maxValue, v) < 0) {\n maxValue = v\n }\n }\n return maxValue\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.maxOrNull(): UInt? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns
the largest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.maxOrNull(): ULong? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.maxOrNull(): UByte? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return max\n}\n\n/**\n * Returns the largest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.maxOrNull(): UShort? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (max < e) max = e\n }\n return
max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow-U\")\n@ExperimentalUnsignedTypes\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun UIntArray.maxWith(comparator: Comparator<in UInt>): UInt {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow-U\")\n@ExperimentalUnsignedTypes\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun ULongArray.maxWith(comparator: Comparator<in ULong>): ULong {\n if (isEmpty()) throw NoSuchElementException()\n
var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow-U\")\n@ExperimentalUnsignedTypes\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun UByteArray.maxWith(comparator: Comparator<in UByte>): UByte {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"maxWithOrThrow-U\")\n@ExperimentalUnsignedTypes\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic
fun UShortArray.maxWith(comparator: Comparator<in UShort>): UShort {\n if (isEmpty()) throw NoSuchElementException()\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.maxWithOrNull(comparator: Comparator<in UInt>): UInt? {\n if (isEmpty()) return null\n var max = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(max, e) < 0) max = e\n }\n return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n

```
*/\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic
 fun ULongArray.maxWithOrNull(comparator: Comparator<in ULong>): ULong? {\n    if (isEmpty()) return null\n
 var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (comparator.compare(max, e) < 0) max =
e\n    }\n    return max\n}\n\n/**\n * Returns the first element having the largest value according to the provided
[comparator] or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.maxWithOrNull(comparator:
Comparator<in UByte>): UByte? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n
    val e = this[i]\n        if (comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the
first element having the largest value according to the provided [comparator] or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.maxWithOrNull(comparator:
Comparator<in UShort>):
 UShort? {\n    if (isEmpty()) return null\n    var max = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n
if (comparator.compare(max, e) < 0) max = e\n    }\n    return max\n}\n\n/**\n * Returns the smallest element.\n * \n
* @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow-
U\")\n@ExperimentalUnsignedTypes\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun
UIntArray.min(): UInt {\n    if (isEmpty()) throw NoSuchElementException()\n    var min = this[0]\n    for (i in
1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\n/**\n * Returns the smallest
element.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow-
U\")\n@ExperimentalUnsignedTypes\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun
ULongArray.min(): ULong {\n    if (isEmpty()) throw NoSuchElementException()\n    var
 min = this[0]\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return
min\n}\n\n/**\n * Returns the smallest element.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow-
U\")\n@ExperimentalUnsignedTypes\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun
UByteArray.min(): UByte {\n    if (isEmpty()) throw NoSuchElementException()\n    var min = this[0]\n    for (i in
1..lastIndex) {\n        val e = this[i]\n        if (min > e) min = e\n    }\n    return min\n}\n\n/**\n * Returns the smallest
element.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minOrThrow-
U\")\n@ExperimentalUnsignedTypes\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun
UShortArray.min(): UShort {\n    if (isEmpty()) throw NoSuchElementException()\n    var min = this[0]\n    for (i in
1..lastIndex) {\n        val e = this[i]\n
    if (min > e) min = e\n    }\n    return min\n}\n\n/**\n * Returns the first element yielding the smallest value of
the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow-
U\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress(\"CONFLICTING_OVERLOADS
\")\npublic inline fun <R : Comparable<R>> UIntArray.minBy(selector: (UInt) -> R): UInt {\n    if (isEmpty())
throw NoSuchElementException()\n    var minElem = this[0]\n    val lastIndex = this.lastIndex\n    if (lastIndex ==
0) return minElem\n    var minValue = selector(minElem)\n    for (i in 1..lastIndex) {\n        val e = this[i]\n        val
v = selector(e)\n        if (minValue > v) {\n            minElem = e\n            minValue = v\n        }\n    }\n    return
minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n
 * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample
samples.collections.Collections.Aggregates.minBy\n
*/\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow-
U\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress(\"CONFLICTING_OVERLOADS
\")\npublic inline fun <R : Comparable<R>> ULongArray.minBy(selector: (ULong) -> R): ULong {\n    if
```

(isEmpty()) throw NoSuchElementException()\n   var minElem = this[0]\n   val lastIndex = this.lastIndex\n   if (lastIndex == 0) return minElem\n   var minValue = selector(minElem)\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   val v = selector(e)\n   if (minValue > v) {\n   minElem = e\n   minValue = v\n   }\n   }\n   return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.minBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow-U\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <R : Comparable<R>> UByteArray.minBy(selector: (UByte) -> R): UByte {\n   if (isEmpty()) throw NoSuchElementException()\n   var minElem = this[0]\n   val lastIndex = this.lastIndex\n   if (lastIndex == 0) return minElem\n   var minValue = selector(minElem)\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   val v = selector(e)\n   if (minValue > v) {\n   minElem = e\n   minValue = v\n   }\n   }\n   return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function.\n * \n * @throws NoSuchElementException if the array is empty.\n * \n * @sample samples.collections.Collections.Aggregates.minBy\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minByOrThrow-U\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic inline fun <R : Comparable<R>> UShortArray.minBy(selector: (UShort) -> R): UShort {\n   if (isEmpty()) throw NoSuchElementException()\n   var minElem = this[0]\n   val lastIndex = this.lastIndex\n   if (lastIndex == 0) return minElem\n   var minValue = selector(minElem)\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   val v = selector(e)\n   if (minValue > v) {\n   minElem = e\n   minValue = v\n   }\n   }\n   return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UIntArray.minByOrNull(selector: (UInt) -> R): UInt? {\n   if (isEmpty()) return null\n   var minElem = this[0]\n   val lastIndex = this.lastIndex\n   if (lastIndex == 0) return minElem\n   var minValue = selector(minElem)\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   val v = selector(e)\n   if (minValue > v) {\n   minElem = e\n   minValue = v\n   }\n   }\n   return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ULongArray.minByOrNull(selector: (ULong) -> R): ULong? {\n   if (isEmpty()) return null\n   var minElem = this[0]\n   val lastIndex = this.lastIndex\n   if (lastIndex == 0) return minElem\n   var minValue = selector(minElem)\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   val v = selector(e)\n   if (minValue > v) {\n   minElem = e\n   minValue = v\n   }\n   }\n   return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UByteArray.minByOrNull(selector: (UByte) -> R): UByte? {\n   if (isEmpty()) return null\n   var minElem = this[0]\n   val lastIndex = this.lastIndex\n   if (lastIndex == 0) return minElem\n   var minValue = selector(minElem)\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   val v = selector(e)\n   if (minValue > v) {\n   minElem = e\n   minValue = v\n   }\n   }\n   return minElem\n}\n\n/**\n * Returns the first element yielding the smallest value of the given function or `null` if there are no elements.\n * \n * @sample samples.collections.Collections.Aggregates.minByOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UShortArray.minByOrNull(selector: (UShort) -> R): UShort? {\n   if (isEmpty()) return null\n   var minElem = this[0]\n   val lastIndex = this.lastIndex\n   if (lastIndex == 0) return minElem\n

var minValue = selector(minElem)\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   val v = selector(e)\n   if (minValue > v) {\n   minElem = e\n   minValue = v\n   }\n   }\n   return minElem\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.minOf(selector: (UInt) -> Double): Double {\n   if (isEmpty()) throw NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   minValue = minOf(minValue, v)\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.minOf(selector: (ULong) -> Double): Double {\n   if (isEmpty()) throw NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   minValue = minOf(minValue, v)\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.minOf(selector: (UByte) -> Double): Double {\n   if (isEmpty()) throw NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   minValue = minOf(minValue, v)\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.minOf(selector: (UShort) -> Double): Double {\n   if (isEmpty()) throw NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   minValue = minOf(minValue, v)\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.minOf(selector: (UInt) -> Float): Float {\n   if (isEmpty()) throw NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   minValue = minOf(minValue, v)\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.minOf(selector: (ULong) -> Float): Float {\n   if (isEmpty()) throw NoSuchElementException()\n

var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   minValue = minOf(minValue, v)\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.minOf(selector: (UByte) -> Float): Float {\n   if (isEmpty()) throw NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   minValue = minOf(minValue, v)\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.minOf(selector: (UShort) -> Float): Float {\n   if (isEmpty()) throw NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   minValue = minOf(minValue, v)\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UIntArray.minOf(selector: (UInt) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (minValue > v) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ULongArray.minOf(selector: (ULong) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (minValue > v) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UByteArray.minOf(selector: (UByte) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (minValue > v) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UShortArray.minOf(selector: (UShort) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (minValue > v) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n

```
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
 inline fun UIntArray.minOfOrNull(selector: (UInt) -> Double): Double? {\n    if (isEmpty()) return null\n    var
minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue =
minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.minOfOrNull(selector: (ULong) -> Double): Double? {\n    if (isEmpty()) return null\n    var minValue
= selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n
        minValue = minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all
values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n *
\n * If any of values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.minOfOrNull(selector: (UByte) -> Double): Double? {\n    if (isEmpty()) return null\n    var minValue
= selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n
  }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n
* applied to each element in the array or `null` if there are no elements.\n * \n * If any of values
 produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UShortArray.minOfOrNull(selector: (UShort) -> Double): Double? {\n    if (isEmpty()) return null\n    var minValue
= selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n
  }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n
* applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by
[selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
 inline fun UIntArray.minOfOrNull(selector: (UInt) -> Float): Float? {\n    if (isEmpty()) return null\n    var
minValue = selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue =
minOf(minValue, v)\n    }\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced
by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n * \n * If any of
values produced by [selector] function is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
ULongArray.minOfOrNull(selector: (ULong) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue =
selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n
}\n    return
 minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to
each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function
is `NaN`, the returned result is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
UByteArray.minOfOrNull(selector: (UByte) -> Float): Float? {\n    if (isEmpty()) return null\n    var minValue =
selector(this[0])\n    for (i in 1..lastIndex) {\n        val v = selector(this[i])\n        minValue = minOf(minValue, v)\n
}\n    return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n *
```

applied to each element in the array or `null` if there are no elements.\n * \n * If any of values produced by [selector] function is `NaN`, the returned result
is `NaN`.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.minOfOrNull(selector: (UShort) -> Float): Float? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n minValue = minOf(minValue, v)\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UIntArray.minOfOrNull(selector: (UInt) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for
(i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> ULongArray.minOfOrNull(selector: (ULong) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R : Comparable<R>> UByteArray.minOfOrNull(selector: (UByte) -> R): R? {\n if (isEmpty()) return null\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value among all values produced by [selector] function\n * applied to each element in the array or `null` if there are no elements.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R : Comparable<R>> UShortArray.minOfOrNull(selector: (UShort) -> R): R? {\n if (isEmpty()) return null\n var minValue
= selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (minValue > v) {\n minValue = v\n }\n }\n return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.minOfWith(comparator: Comparator<in R>, selector: (UInt) -> R): R {\n if (isEmpty()) throw NoSuchElementException()\n var minValue = selector(this[0])\n for (i in 1..lastIndex) {\n val v = selector(this[i])\n if (comparator.compare(minValue, v) > 0) {\n minValue = v\n }\n }\n return
minValue\n}\n\n/**\n * Returns the
smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.minOfWith(comparator: Comparator<in R>, selector: (ULong) -> R): R {\n if (isEmpty()) throw

NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(minValue, v) > 0) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.minOfWith(comparator: Comparator<in R>, selector: (UByte) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(minValue, v) > 0) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array.\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.minOfWith(comparator: Comparator<in R>, selector: (UShort) -> R): R {\n   if (isEmpty()) throw NoSuchElementException()\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(minValue, v) > 0) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UInt) -> R): R? {\n   if (isEmpty()) return null\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(minValue, v) > 0) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (ULong) -> R): R? {\n   if (isEmpty()) return null\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(minValue, v) > 0) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function\n applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UByte) -> R): R? {\n   if (isEmpty()) return null\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(minValue, v) > 0) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest value according to the provided [comparator]\n * among all values produced by [selector] function applied to each element in the array or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic\n inline fun <R> UShortArray.minOfWithOrNull(comparator: Comparator<in R>, selector: (UShort) -> R): R? {\n   if (isEmpty()) return null\n   var minValue = selector(this[0])\n   for (i in 1..lastIndex) {\n   val v = selector(this[i])\n   if (comparator.compare(minValue, v) > 0) {\n   minValue = v\n   }\n   }\n   return minValue\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n

*/\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.minOrNull(): UInt? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.minOrNull(): ULong? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.minOrNull(): UByte? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the smallest element or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.minOrNull(): UShort? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (min > e) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow-U\")\n@ExperimentalUnsignedTypes\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun UIntArray.minWith(comparator: Comparator<in UInt>): UInt {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow-U\")\n@ExperimentalUnsignedTypes\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun ULongArray.minWith(comparator: Comparator<in ULong>): ULong {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow-U\")\n@ExperimentalUnsignedTypes\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun UByteArray.minWith(comparator: Comparator<in UByte>): UByte {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator].\n * \n * @throws NoSuchElementException if the array is empty.\n */\n@SinceKotlin(\"1.7\")\n@kotlin.jvm.JvmName(\"minWithOrThrow-U\")\n@ExperimentalUnsignedTypes\n@Suppress(\"CONFLICTING_OVERLOADS\")\npublic fun UShortArray.minWith(comparator: Comparator<in UShort>): UShort {\n if (isEmpty()) throw NoSuchElementException()\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UIntArray.minWithOrNull(comparator: Comparator<in UInt>): UInt? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun ULongArray.minWithOrNull(comparator: Comparator<in ULong>): ULong? {\n if (isEmpty()) return null\n var min = this[0]\n for (i in 1..lastIndex) {\n val e = this[i]\n if (comparator.compare(min, e) > 0) min = e\n }\n return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UByteArray.minWithOrNull(comparator:

Comparator<in UByte>): UByte? {\n   if (isEmpty()) return null\n   var min = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (comparator.compare(min, e) > 0) min = e\n   }\n   return min\n}\n\n/**\n * Returns the first element having the smallest value according to the provided [comparator] or `null` if there are no elements.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun UShortArray.minWithOrNull(comparator: Comparator<in UShort>): UShort? {\n   if (isEmpty()) return null\n   var min = this[0]\n   for (i in 1..lastIndex) {\n   val e = this[i]\n   if (comparator.compare(min, e) > 0) min = e\n   }\n   return min\n}\n\n/**\n * Returns `true` if the array
has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.none(): Boolean {\n   return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.none(): Boolean {\n   return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.none(): Boolean {\n   return isEmpty()\n}\n\n/**\n * Returns `true` if the array has no elements.\n * \n * @sample samples.collections.Collections.Aggregates.none\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun UShortArray.none(): Boolean {\n   return isEmpty()\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.none(predicate: (UInt) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.none(predicate: (ULong) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.none(predicate: (UByte) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return false\n   return true\n}\n\n/**\n * Returns `true` if no elements match the given [predicate].\n * \n * @sample samples.collections.Collections.Aggregates.noneWithPredicate\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.none(predicate: (UShort) -> Boolean): Boolean {\n   for (element in this) if (predicate(element)) return false\n   return true\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.onEach(action: (UInt) -> Unit): UIntArray {\n   return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given
[action] on each element and returns the array itself afterwards.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.onEach(action: (ULong) -> Unit): ULongArray {\n   return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.onEach(action: (UByte) -> Unit): UByteArray {\n   return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each element and returns the array itself afterwards.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.onEach(action: (UShort) -> Unit): UShortArray {\n   return apply { for (element in this) action(element) }\n}\n\n/**\n * Performs the given [action] on each element, providing

sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.onEachIndexed(action: (index: Int, UInt) -> Unit): UIntArray {\n   return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.onEachIndexed(action: (index: Int, ULong) -> Unit): ULongArray {\n   return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.onEachIndexed(action: (index: Int, UByte) -> Unit): UByteArray {\n   return apply { forEachIndexed(action) }\n}\n\n/**\n * Performs the given [action] on each element, providing sequential index with the element,\n * and returns the array itself afterwards.\n * @param [action] function that takes the index of an element and the element itself\n * and performs the action on the element.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.onEachIndexed(action: (index: Int, UShort) -> Unit): UShortArray {\n   return apply { forEachIndexed(action) }\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reduce(operation: (acc: UInt, UInt) -> UInt): UInt {\n   if (isEmpty())\n   throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n   var accumulator = this[0]\n   for (index in 1..lastIndex) {\n   accumulator = operation(accumulator, this[index])\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reduce(operation: (acc: ULong, ULong) -> ULong): ULong {\n   if (isEmpty())\n   throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n   var accumulator = this[0]\n   for (index in 1..lastIndex) {\n   accumulator = operation(accumulator, this[index])\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reduce(operation: (acc: UByte, UByte) -> UByte): UByte {\n   if (isEmpty())\n   throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n   var accumulator = this[0]\n   for (index in 1..lastIndex) {\n   accumulator = operation(accumulator, this[index])\n   }\n   return accumulator\n}\n\n/**\n *

Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator
value and each element.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reduce(operation: (acc: UShort, UShort) -> UShort): UShort {\n   if (isEmpty())\n   throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n   var accumulator = this[0]\n   for (index in 1..lastIndex) {\n   accumulator = operation(accumulator, this[index])\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each
element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reduceIndexed(operation: (index: Int, acc: UInt, UInt) -> UInt): UInt {\n   if (isEmpty())\n   throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n   var accumulator = this[0]\n   for (index in 1..lastIndex) {\n   accumulator = operation(index, accumulator, this[index])\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying
[operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reduceIndexed(operation: (index: Int, acc: ULong, ULong) -> ULong): ULong {\n   if (isEmpty())\n
throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n   var accumulator = this[0]\n   for (index in 1..lastIndex) {\n   accumulator = operation(index, accumulator, this[index])\n   }\n   return accumulator\n}\n\n/**\n
* Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reduceIndexed(operation: (index: Int, acc: UByte, UByte) -> UByte): UByte {\n   if (isEmpty())\n
throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n   var accumulator = this[0]\n   for (index in 1..lastIndex) {\n   accumulator
= operation(index, accumulator, this[index])\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduce\n

*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reduceIndexed(operation: (index: Int, acc: UShort, UShort) -> UShort): UShort {\n if (isEmpty())\n throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reduceIndexedOrNull(operation: (index: Int, acc: UInt, UInt) -> UInt): UInt? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator,
this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reduceIndexedOrNull(operation: (index: Int, acc: ULong, ULong) -> ULong): ULong? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element
and applying [operation] from left to right\n * to current accumulator value and each element with its index in the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reduceIndexedOrNull(operation: (index: Int, acc: UByte, UByte) -> UByte): UByte? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element with its index in
the original array.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, current accumulator value and the element itself,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reduceIndexedOrNull(operation: (index: Int, acc: UShort, UShort) -> UShort): UShort? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n accumulator = operation(index, accumulator, this[index])\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and
an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reduceOrNull(operation: (acc: UInt, UInt) -> UInt): UInt? {\n if (isEmpty())\n return null\n var accumulator = this[0]\n for (index in 1..lastIndex) {\n

accumulator = operation(accumulator, this[index])\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
 inline fun ULongArray.reduceOrNull(operation: (acc: ULong, ULong) -> ULong): ULong? {\n   if (isEmpty())\n return null\n   var accumulator = this[0]\n   for (index in 1..lastIndex) {\n   accumulator = operation(accumulator, this[index])\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reduceOrNull(operation: (acc:
 UByte, UByte) -> UByte): UByte? {\n   if (isEmpty())\n   return null\n   var accumulator = this[0]\n   for (index in 1..lastIndex) {\n   accumulator = operation(accumulator, this[index])\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the first element and applying [operation] from left to right\n * to current accumulator value and each element.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes current accumulator value and an element,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reduceOrNull(operation: (acc: UShort, UShort) ->
 UShort): UShort? {\n   if (isEmpty())\n   return null\n   var accumulator = this[0]\n   for (index in 1..lastIndex) {\n   accumulator
 = operation(accumulator, this[index])\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reduceRight(operation: (UInt, acc: UInt) -> UInt): UInt {\n   var index = lastIndex\n   if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n   var accumulator = get(index--)\n   while (index >= 0) {\n   accumulator
 = operation(get(index--), accumulator)\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reduceRight(operation: (ULong, acc: ULong) -> ULong): ULong {\n   var index = lastIndex\n   if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n   var accumulator = get(index--)\n   while (index >= 0) {\n   accumulator
 = operation(get(index--), accumulator)\n   }\n   return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n

* Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reduceRight(operation: (UByte, acc: UByte) -> UByte): UByte {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reduceRight(operation: (UShort, acc: UShort) -> UShort): UShort {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(get(index--), accumulator)\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reduceRightIndexed(operation: (index: Int, UInt, acc: UInt) -> UInt): UInt {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reduceRightIndexed(operation: (index: Int, ULong, acc: ULong) -> ULong): ULong {\n var index = lastIndex\n if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n var accumulator = get(index--)\n while (index >= 0) {\n accumulator = operation(index, get(index), accumulator)\n --index\n }\n return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic

```
 inline fun UByteArray.reduceRightIndexed(operation: (index: Int, UByte, acc: UByte) -> UByte): UByte {\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Throws an exception if this array is empty. If the array can be empty in an expected way,\n * please use [reduceRightIndexedOrNull] instead. It returns `null` when its receiver is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator
 value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRight\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reduceRightIndexed(operation: (index: Int, UShort, acc: UShort) -> UShort): UShort {\n    var index = lastIndex\n    if (index < 0) throw UnsupportedOperationException(\"Empty array can't be reduced.\")\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n
 * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reduceRightIndexedOrNull(operation: (index: Int, UInt, acc: UInt) -> UInt): UInt? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
 inline fun ULongArray.reduceRightIndexedOrNull(operation: (index: Int, ULong, acc: ULong) -> ULong): ULong? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
 inline fun UByteArray.reduceRightIndexedOrNull(operation: (index: Int, UByte, acc: UByte) -> UByte): UByte? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index), accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to left\n * to each element with its index in the original array and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function that takes the index of an element, the element itself and current accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun
```

```
UShortArray.reduceRightIndexedOrNull(operation: (index:
 Int, UShort, acc: UShort) -> UShort): UShort? {\n    var index = lastIndex\n    if (index < 0) return null\n    var
accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(index, get(index),
accumulator)\n        --index\n    }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last
element and applying [operation] from right to left\n * to each element and current accumulator value.\n * \n *
Returns `null` if the array is empty.\n * \n * @param [operation] function that takes an element and current
accumulator value,\n * and calculates the next accumulator value.\n * \n * @sample
samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UIntArray.reduceRightOrNull(operation: (UInt, acc: UInt) -> UInt):
UInt? {\n    var index = lastIndex\n    if (index < 0) return null\n
 var accumulator = get(index--)\n    while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n
  }\n    return accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation]
from right to left\n * to each element and current accumulator value.\n * \n * Returns `null` if the array is empty.\n *
\n * @param [operation] function that takes an element and current accumulator value,\n * and calculates the next
accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun ULongArray.reduceRightOrNull(operation: (ULong, acc: ULong) ->
ULong): ULong? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n
while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return
 accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param
[operation] function that takes an element and current accumulator value,\n * and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UByteArray.reduceRightOrNull(operation: (UByte, acc: UByte) ->
UByte): UByte? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n
while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return
accumulator\n}\n\n/**\n * Accumulates value starting with the last element and applying [operation] from right to
left\n * to each element
 and current accumulator value.\n * \n * Returns `null` if the array is empty.\n * \n * @param [operation] function
that takes an element and current accumulator value,\n * and calculates the next accumulator value.\n * \n *
@sample samples.collections.Collections.Aggregates.reduceRightOrNull\n
*/\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n
@kotlin.internal.InlineOnly\npublic inline fun UShortArray.reduceRightOrNull(operation: (UShort, acc: UShort) ->
UShort): UShort? {\n    var index = lastIndex\n    if (index < 0) return null\n    var accumulator = get(index--)\n
while (index >= 0) {\n        accumulator = operation(get(index--), accumulator)\n    }\n    return
accumulator\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
[operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n
* Note that `acc` value passed to [operation]
 function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param
[operation] function that takes current accumulator value and an element, and calculates the next accumulator
value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n
*/\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.runningFold(initial: R, operation: (acc: R, UInt) -> R): List<R> {\n    if (isEmpty()) return
listOf(initial)\n    val result = ArrayList<R>(size + 1).apply { add(initial) }\n    var accumulator = initial\n    for
(element in this) {\n        accumulator = operation(accumulator, element)\n        result.add(accumulator)\n    }\n
return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying
```

[operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that
`acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.runningFold(initial: R, operation: (acc: R, ULong) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with
[initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.runningFold(initial: R, operation: (acc: R, UByte) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current
accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.runningFold(initial: R, operation: (acc: R, UShort) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (element in this) {\n accumulator = operation(accumulator, element)\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n
* to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n
return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R>

ULongArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningFold\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.runningFoldIndexed(initial: R, operation: (index: Int, acc: R, UShort) -> R): List<R> {\n if (isEmpty()) return listOf(initial)\n val result = ArrayList<R>(size + 1).apply { add(initial) }\n var accumulator = initial\n for (index in indices) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.runningReduce(operation: (acc: UInt, UInt) -> UInt): List<UInt> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<UInt>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.runningReduce(operation: (acc: ULong, ULong) -> ULong): List<ULong> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<ULong>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would

affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element,
and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.runningReduce(operation: (acc: UByte, UByte) -> UByte): List<UByte> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<UByte>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation]
function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.runningReduce(operation: (acc: UShort, UShort) -> UShort): List<UShort> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<UShort>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n
* otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.runningReduceIndexed(operation: (index: Int, acc: UInt, UInt) -> UInt): List<UInt> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<UInt>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current
accumulator value that starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.runningReduceIndexed(operation: (index: Int, acc: ULong, ULong) -> ULong): List<ULong> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<ULong>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing
successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n

*/\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.runningReduceIndexed(operation: (index: Int, acc: UByte, UByte) -> UByte): List<UByte> {\n if (isEmpty()) return emptyList()\n var accumulator = this[0]\n val result = ArrayList<UByte>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with the first element of this array.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.runningReduce\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.runningReduceIndexed(operation: (index: Int, acc: UShort, UShort) -> UShort): List<UShort> {\n if (isEmpty())
return emptyList()\n var accumulator = this[0]\n val result = ArrayList<UShort>(size).apply { add(accumulator) }\n for (index in 1 until size) {\n accumulator = operation(index, accumulator, this[index])\n result.add(accumulator)\n }\n return result\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R>
UIntArray.scan(initial: R, operation: (acc: R, UInt) -> R): List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.scan(initial: R, operation: (acc: R, ULong) -> R): List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation
values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.scan(initial: R, operation: (acc: R, UByte) -> R): List<R> {\n return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element and current accumulator value that starts with [initial] value.\n * \n * Note that
`acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes current accumulator value and an element, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.scan(initial: R, operation: (acc: R, UShort) -> R):

List<R> {\n   return runningFold(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> UIntArray.scanIndexed(initial: R, operation: (index: Int, acc: R, UInt) -> R): List<R> {\n   return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element,
 current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> ULongArray.scanIndexed(initial: R, operation: (index: Int, acc: R, ULong) -> R): List<R> {\n   return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the
 next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> UByteArray.scanIndexed(initial: R, operation: (index: Int, acc: R, UByte) -> R): List<R> {\n   return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns a list containing successive accumulation values generated by applying [operation] from left to right\n * to each element, its index in the original array and current accumulator value that starts with [initial] value.\n * \n * Note that `acc` value passed to [operation] function should not be mutated;\n * otherwise it would affect the previous value in resulting list.\n * \n * @param [operation] function that takes the index of an element, current accumulator value\n * and the element itself, and calculates the next accumulator value.\n * \n * @sample samples.collections.Collections.Aggregates.scan\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun <R> UShortArray.scanIndexed(initial: R, operation: (index: Int, acc: R, UShort) -> R): List<R> {\n   return runningFoldIndexed(initial, operation)\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.sumBy(selector: (UInt) -> UInt): UInt {\n   var sum: UInt = 0u\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\",
 ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince
 = \"1.5\")\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.sumBy(selector: (ULong) -> UInt): UInt {\n   var sum: UInt = 0u\n   for (element in this) {\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\",

ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.sumBy(selector: (UByte) -> UInt): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.sumBy(selector: (UShort) -> UInt): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.sumByDouble(selector: (UInt) -> Double): Double {\n var sum: Double = 0.0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.sumByDouble(selector: (ULong) -> Double): Double {\n var sum: Double = 0.0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.sumByDouble(selector: (UByte) -> Double): Double {\n var sum: Double = 0.0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@Deprecated(\"Use sumOf instead.\", ReplaceWith(\"this.sumOf(selector)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.sumByDouble(selector: (UShort) -> Double): Double {\n var sum: Double = 0.0\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfDouble\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.sumOf(selector: (UInt) -> Double): Double {\n var sum: Double = 0.toDouble()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfDouble\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.sumOf(selector: (ULong) -> Double): Double {\n var sum: Double = 0.toDouble()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfDouble\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.sumOf(selector: (UByte) -> Double): Double {\n var sum: Double = 0.toDouble()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the

array.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfDouble\
")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.sumOf(selector:
(UShort) -> Double): Double {\n   var sum: Double = 0.toDouble()\n   for (element in this) {\n   sum +=
selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
applied to each element in the array.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfInt\")\n
@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
 inline fun UIntArray.sumOf(selector: (UInt) -> Int): Int {\n   var sum: Int = 0.toInt()\n   for (element in this) {\n
 sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfInt\")\n
@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.sumOf(selector:
(ULong) -> Int): Int {\n   var sum: Int = 0.toInt()\n   for (element in this) {\n   sum += selector(element)\n   }\n
return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in
the array.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfInt\")\n
@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
 inline fun UByteArray.sumOf(selector: (UByte) -> Int): Int {\n   var sum: Int = 0.toInt()\n   for (element in this)
{\n   sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfInt\")\n
@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.sumOf(selector:
(UShort) -> Int): Int {\n   var sum: Int = 0.toInt()\n   for (element in this) {\n   sum += selector(element)\n   }\n
return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function
 applied to each element in the array.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfLong\")
\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.sumOf(selector: (UInt)
-> Long): Long {\n   var sum: Long = 0.toLong()\n   for (element in this) {\n   sum += selector(element)\n   }\n
 return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in
the array.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfLong\")
\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.sumOf(selector:
(ULong) -> Long): Long {\n   var sum: Long = 0.toLong()\n   for (element in this) {\n
  sum += selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector]
function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfLong\")
\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.sumOf(selector:
(UByte) -> Long): Long {\n   var sum: Long = 0.toLong()\n   for (element in this) {\n   sum +=
selector(element)\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function

applied to each element in the array.\n
*/\n@SinceKotlin(\"1.4\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfLong\")
\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline
fun UShortArray.sumOf(selector: (UShort) -> Long): Long {\n var sum: Long = 0.toLong()\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfUInt\")\
n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline
Only\npublic inline fun UIntArray.sumOf(selector: (UInt) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for
(element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfUInt\")\
n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline
Only\npublic
inline fun ULongArray.sumOf(selector: (ULong) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n for (element in
this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by
[selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfUInt\")\
n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline
Only\npublic inline fun UByteArray.sumOf(selector: (UByte) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n
for (element in this) {\n sum += selector(element)\n
}\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each
element in the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfUInt\")\
n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inline
Only\npublic inline fun UShortArray.sumOf(selector: (UShort) -> UInt): UInt {\n var sum: UInt = 0.toUInt()\n
for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all
values produced by [selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfULong\
")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inli
neOnly\npublic
inline fun UIntArray.sumOf(selector: (UInt) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for
(element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values
produced by [selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfULong\
")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.Inli
neOnly\npublic inline fun ULongArray.sumOf(selector: (ULong) -> ULong): ULong {\n var sum: ULong =
0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns
the sum of all values produced by [selector] function applied to each element in the array.\n
*/\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolution
ByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfULong\

")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic
inline fun UByteArray.sumOf(selector: (UByte) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns the sum of all values produced by [selector] function applied to each element in the array.\n */\n@SinceKotlin(\"1.5\")\n@OptIn(kotlin.experimental.ExperimentalTypeInference::class)\n@OverloadResolutionByLambdaReturnType\n@Suppress(\"INAPPLICABLE_JVM_NAME\")\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@ExperimentalUnsignedTypes\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.sumOf(selector: (UShort) -> ULong): ULong {\n var sum: ULong = 0.toULong()\n for (element in this) {\n sum += selector(element)\n }\n return sum\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UIntArray.zip(other: Array<out R>): List<Pair<UInt, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic infix fun <R> ULongArray.zip(other: Array<out R>): List<Pair<ULong, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UByteArray.zip(other: Array<out R>): List<Pair<UByte, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UShortArray.zip(other: Array<out R>): List<Pair<UShort, R>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V> UIntArray.zip(other: Array<out R>, transform: (a: UInt, b: R) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic
inline fun <R, V> ULongArray.zip(other: Array<out R>, transform: (a: ULong, b: R) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V> UByteArray.zip(other: Array<out R>, transform: (a: UByte, b: R) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n

}\n   return list\n}\n\n/**\n
 * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V> UShortArray.zip(other: Array<out R>, transform: (a: UShort, b: R) -> V): List<V> {\n   val size = minOf(size, other.size)\n   val list = ArrayList<V>(size)\n   for (i in 0 until size) {\n   list.add(transform(this[i], other[i]))\n   }\n   return list\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic
 infix fun <R> UIntArray.zip(other: Iterable<R>): List<Pair<UInt, R>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic infix fun <R> ULongArray.zip(other: Iterable<R>): List<Pair<ULong, R>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UByteArray.zip(other: Iterable<R>): List<Pair<UByte, R>> {\n   return zip(other) { t1,
 t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` collection and [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic infix fun <R> UShortArray.zip(other: Iterable<R>): List<Pair<UShort, R>> {\n   return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V> UIntArray.zip(other: Iterable<R>, transform: (a: UInt, b: R) -> V): List<V> {\n   val
 arraySize = size\n   val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n   var i = 0\n   for (element in other) {\n   if (i >= arraySize) break\n   list.add(transform(this[i++], element))\n   }\n   return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V> ULongArray.zip(other: Iterable<R>, transform: (a: ULong, b: R) -> V): List<V> {\n   val arraySize = size\n   val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n   var i = 0\n   for (element in other) {\n   if (i >= arraySize) break\n   list.add(transform(this[i++],
 element))\n   }\n   return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V> UByteArray.zip(other: Iterable<R>, transform: (a: UByte, b: R) -> V): List<V> {\n   val arraySize = size\n   val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n   var i = 0\n   for (element in other) {\n

if (i >= arraySize) break\n list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] collection with the same index\n * using the provided
[transform] function applied to each pair of elements.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <R, V> UShortArray.zip(other: Iterable<R>, transform: (a: UShort, b: R) -> V): List<V> {\n val arraySize = size\n val list = ArrayList<V>(minOf(other.collectionSizeOrDefault(10), arraySize))\n var i = 0\n for (element in other) {\n if (i >= arraySize) break\n list.add(transform(this[i++], element))\n }\n return list\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic infix fun UIntArray.zip(other:
UIntArray): List<Pair<UInt, UInt>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic infix fun ULongArray.zip(other: ULongArray): List<Pair<ULong, ULong>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic infix fun UByteArray.zip(other: UByteArray): List<Pair<UByte, UByte>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of pairs built from
the elements of `this` array and the [other] array with the same index.\n * The returned list has length of the shortest collection.\n * \n * @sample samples.collections.Iterables.Operations.zipIterable\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic infix fun UShortArray.zip(other: UShortArray): List<Pair<UShort, UShort>> {\n return zip(other) { t1, t2 -> t1 to t2 }\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V> UIntArray.zip(other: UIntArray, transform: (a: UInt, b: UInt) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for
(i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V> ULongArray.zip(other: ULongArray, transform: (a: ULong, b: ULong) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array with the same index\n * using the provided [transform] function applied to each pair of elements.\n
* The returned list has length of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V> UByteArray.zip(other: UByteArray, transform: (a: UByte, b: UByte) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns a list of values built from the elements of `this` array and the [other] array

with the same index\n * using the provided [transform] function applied to each pair of elements.\n * The returned list has length of the shortest array.\n * \n * @sample samples.collections.Iterables.Operations.zipIterableWithTransform\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun <V> UShortArray.zip(other: UShortArray, transform: (a: UShort, b: UShort) -> V): List<V> {\n val size = minOf(size, other.size)\n val list = ArrayList<V>(size)\n for (i in 0 until size) {\n list.add(transform(this[i], other[i]))\n }\n return list\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Array<out UInt>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Array<out ULong>.sum(): ULong {\n var sum: ULong = 0uL\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\n@kotlin.jvm.JvmName(\"sumOfUByte\")\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Array<out UByte>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\n@kotlin.jvm.JvmName(\"sumOfUShort\")\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Array<out UShort>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UIntArray.sum(): UInt {\n return storage.sum().toUInt()\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun ULongArray.sum(): ULong {\n return storage.sum().toULong()\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UByteArray.sum(): UInt {\n return sumOf { it.toUInt() }\n}\n\n/**\n * Returns the sum of all elements in the array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\n@kotlin.internal.InlineOnly\npublic inline fun UShortArray.sum(): UInt {\n return sumOf { it.toUInt() }\n}\n\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"UCollectionsKt\")\n\npackage kotlin.collections\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.random.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns an array of UByte containing all of the elements of this collection.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun Collection<UByte>.toUByteArray(): UByteArray {\n val result = UByteArray(size)\n var index = 0\n for (element in this)\n result[index++] = element\n return result\n}\n\n/**\n * Returns an array of UInt containing all of the elements of this collection.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun Collection<UInt>.toUIntArray(): UIntArray {\n val result = UIntArray(size)\n var index = 0\n for (element in this)\n result[index++] = element\n return result\n}\n\n/**\n * Returns an array of ULong containing all of the elements of this collection.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun Collection<ULong>.toULongArray(): ULongArray {\n val result = ULongArray(size)\n var index = 0\n for (element in this)\n result[index++] = element\n return result\n}\n\n/**\n * Returns an array of UShort containing all of the elements of this collection.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun Collection<UShort>.toUShortArray(): UShortArray {\n val result = UShortArray(size)\n var index = 0\n for (element in this)\n result[index++] =

element\n   return result\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Iterable<UInt>.sum(): UInt {\n   var sum: UInt = 0u\n   for (element in this) {\n   sum += element\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Iterable<ULong>.sum(): ULong {\n   var sum: ULong = 0uL\n   for (element in this) {\n   sum += element\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName(\"sumOfUByte\")\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Iterable<UByte>.sum(): UInt {\n   var sum: UInt = 0u\n   for (element in this) {\n   sum += element\n   }\n   return sum\n}\n\n/**\n * Returns the sum of all elements in the collection.\n */\n@kotlin.jvm.JvmName(\"sumOfUShort\")\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Iterable<UShort>.sum(): UInt {\n   var sum: UInt = 0u\n   for (element in this) {\n   sum += element\n   }\n   return sum\n}\n\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"UComparisonsKt\")\n\npackage kotlin.comparisons\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED\n// by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.random.*\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun maxOf(a: UInt, b: UInt): UInt {\n   return if (a >= b) a else b\n}\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun maxOf(a: ULong, b: ULong): ULong {\n   return if (a >= b) a else b\n}\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun maxOf(a: UByte, b: UByte): UByte {\n   return if (a >= b) a else b\n}\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun maxOf(a: UShort, b: UShort): UShort {\n   return if (a >= b) a else b\n}\n\n/**\n * Returns the greater of three values.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun maxOf(a: UInt, b: UInt, c: UInt): UInt {\n   return maxOf(a, maxOf(b, c))\n}\n\n/**\n * Returns the greater of three values.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun maxOf(a: ULong, b: ULong, c: ULong): ULong {\n   return maxOf(a, maxOf(b, c))\n}\n\n/**\n * Returns the greater of three values.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun maxOf(a: UByte, b: UByte, c: UByte): UByte {\n   return maxOf(a, maxOf(b, c))\n}\n\n/**\n * Returns the greater of three values.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun maxOf(a: UShort, b: UShort, c: UShort): UShort {\n   return maxOf(a, maxOf(b, c))\n}\n\n/**\n * Returns the greater of the given values.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun maxOf(a: UInt, vararg other: UInt): UInt {\n   var max = a\n   for (e in other) max = maxOf(max, e)\n   return max\n}\n\n/**\n * Returns the greater of the given values.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun maxOf(a: ULong, vararg other: ULong): ULong {\n   var max = a\n   for (e in other) max = maxOf(max, e)\n   return max\n}\n\n/**\n * Returns the greater of the given values.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun maxOf(a: UByte, vararg other: UByte): UByte {\n   var max = a\n   for (e in other) max = maxOf(max, e)\n   return max\n}\n\n/**\n * Returns the greater of the given values.\n

*/\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun maxOf(a: UShort, vararg other: UShort): UShort {\n   var max = a\n   for (e in other) max = maxOf(max, e)\n   return max\n}\n\n/**\n
 * Returns the smaller of two values.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun minOf(a: UInt, b: UInt): UInt {\n   return if (a <= b) a else b\n}\n\n/**\n * Returns the smaller of two values.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun minOf(a: ULong, b: ULong): ULong {\n   return if (a <= b) a else b\n}\n\n/**\n * Returns the smaller of two values.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun minOf(a: UByte, b: UByte): UByte {\n   return if (a <= b) a else b\n}\n\n/**\n * Returns the smaller of two values.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun minOf(a: UShort, b: UShort): UShort {\n   return if (a <= b) a else b\n}\n\n/**\n * Returns the smaller of three values.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline
 fun minOf(a: UInt, b: UInt, c: UInt): UInt {\n   return minOf(a, minOf(b, c))\n}\n\n/**\n * Returns the smaller of three values.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun minOf(a: ULong, b: ULong, c: ULong): ULong {\n   return minOf(a, minOf(b, c))\n}\n\n/**\n * Returns the smaller of three values.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun minOf(a: UByte, b: UByte, c: UByte): UByte {\n   return minOf(a, minOf(b, c))\n}\n\n/**\n * Returns the smaller of three values.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun minOf(a: UShort, b: UShort, c: UShort): UShort {\n   return minOf(a, minOf(b, c))\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun minOf(a: UInt, vararg
 other: UInt): UInt {\n   var min = a\n   for (e in other) min = minOf(min, e)\n   return min\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun minOf(a: ULong, vararg other: ULong): ULong {\n   var min = a\n   for (e in other) min = minOf(min, e)\n   return min\n}\n\n/**\n * Returns the smaller of the given values.\n
*/\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun minOf(a: UByte, vararg other: UByte): UByte {\n   var min = a\n   for (e in other) min = minOf(min, e)\n   return min\n}\n\n/**\n * Returns the smaller of the given values.\n */\n@SinceKotlin(\"1.4\")\n@ExperimentalUnsignedTypes\npublic fun minOf(a: UShort, vararg other: UShort): UShort {\n   var min = a\n   for (e in other) min = minOf(min, e)\n   return min\n}\n\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that
 can be found in the license/LICENSE.txt file.\n
/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"URangesKt\")\n\npackage kotlin.ranges\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.random.\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the progression is empty.\n
*/\n@SinceKotlin(\"1.7\")\npublic fun UIntProgression.first(): UInt {\n   if (isEmpty())\n   throw NoSuchElementException(\"Progression $this is empty.\")\n   return this.first\n}\n\n/**\n * Returns the first element.\n * \n * @throws NoSuchElementException if the progression is empty.\n
*/\n@SinceKotlin(\"1.7\")\npublic fun ULongProgression.first(): ULong {\n   if (isEmpty())\n   throw NoSuchElementException(\"Progression $this is empty.\")\n   return this.first\n}\n\n/**\n * Returns the first element, or `null` if the progression is
 empty.\n */\n@SinceKotlin(\"1.7\")\npublic fun UIntProgression.firstOrNull(): UInt? {\n   return if (isEmpty()) null else this.first\n}\n\n/**\n * Returns the first element, or `null` if the progression is empty.\n

*/\n@SinceKotlin(\"1.7\")\npublic fun ULongProgression.firstOrNull(): ULong? {\n   return if (isEmpty()) null else this.first\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the progression is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.7\")\npublic fun UIntProgression.last(): UInt {\n   if (isEmpty())\n   throw NoSuchElementException(\"Progression $this is empty.\")\n   return this.last\n}\n\n/**\n * Returns the last element.\n * \n * @throws NoSuchElementException if the progression is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.7\")\npublic fun ULongProgression.last(): ULong {\n   if (isEmpty())\n   throw NoSuchElementException(\"Progression
 $this is empty.\")\n   return this.last\n}\n\n/**\n * Returns the last element, or `null` if the progression is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.7\")\npublic fun UIntProgression.lastOrNull(): UInt? {\n   return if (isEmpty()) null else this.last\n}\n\n/**\n * Returns the last element, or `null` if the progression is empty.\n * \n * @sample samples.collections.Collections.Elements.last\n */\n@SinceKotlin(\"1.7\")\npublic fun ULongProgression.lastOrNull(): ULong? {\n   return if (isEmpty()) null else this.last\n}\n\n/**\n * Returns a random element from this range.\n * \n * @throws IllegalArgumentException if this range is empty.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UIntRange.random(): UInt {\n   return random(Random)\n}\n\n/**\n * Returns a random element from this range.\n * \n * @throws IllegalArgumentException if this
 range is empty.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULongRange.random(): ULong {\n   return random(Random)\n}\n\n/**\n * Returns a random element from this range using the specified source of randomness.\n * \n * @throws IllegalArgumentException if this range is empty.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun UIntRange.random(random: Random): UInt {\n   try {\n   return random.nextUInt(this)\n   } catch(e: IllegalArgumentException) {\n   throw NoSuchElementException(e.message)\n   }\n}\n\n/**\n * Returns a random element from this range using the specified source of randomness.\n * \n * @throws IllegalArgumentException if this range is empty.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun ULongRange.random(random: Random): ULong {\n   try {\n   return random.nextULong(this)\n
 } catch(e: IllegalArgumentException) {\n   throw NoSuchElementException(e.message)\n   }\n}\n\n/**\n * Returns a random element from this range, or `null` if this range is empty.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UIntRange.randomOrNull(): UInt? {\n   return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this range, or `null` if this range is empty.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULongRange.randomOrNull(): ULong? {\n   return randomOrNull(Random)\n}\n\n/**\n * Returns a random element from this range using the specified source of randomness, or `null` if this range is empty.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\npublic fun UIntRange.randomOrNull(random:
 Random): UInt? {\n   if (isEmpty())\n   return null\n   return random.nextUInt(this)\n}\n\n/**\n * Returns a random element from this range using the specified source of randomness, or `null` if this range is empty.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class,
ExperimentalUnsignedTypes::class)\npublic fun ULongRange.randomOrNull(random: Random): ULong? {\n   if (isEmpty())\n   return null\n   return random.nextULong(this)\n}\n\n/**\n * Returns `true` if this range contains the specified [element].\n * \n * Always returns `false` if the [element] is `null`.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline operator fun UIntRange.contains(element: UInt?): Boolean {\n   return element != null &&

contains(element)\n}\n\n/**\n * Returns `true` if this range contains the specified [element].\n * \n * Always returns `false` if the [element] is `null`.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic
inline operator fun ULongRange.contains(element: ULong?): Boolean {\n return element != null && contains(element)\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun UIntRange.contains(value: UByte): Boolean {\n return contains(value.toUInt())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun ULongRange.contains(value: UByte): Boolean {\n return contains(value.toULong())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun ULongRange.contains(value: UInt): Boolean {\n return contains(value.toULong())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun UIntRange.contains(value: ULong): Boolean {\n return (value shr UInt.SIZE_BITS) == 0uL && contains(value.toUInt())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun UIntRange.contains(value: UShort): Boolean {\n return contains(value.toUInt())\n}\n\n/**\n * Checks if the specified [value] belongs to this range.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic operator fun ULongRange.contains(value: UShort): Boolean {\n return contains(value.toULong())\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the
returned progression is empty.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun UByte.downTo(to: UByte): UIntProgression {\n return UIntProgression.fromClosedRange(this.toUInt(), to.toUInt(), -1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun UInt.downTo(to: UInt): UIntProgression {\n return UIntProgression.fromClosedRange(this, to, -1)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic
infix fun ULong.downTo(to: ULong): ULongProgression {\n return ULongProgression.fromClosedRange(this, to, -1L)\n}\n\n/**\n * Returns a progression from this value down to the specified [to] value with the step -1.\n * \n * The [to] value should be less than or equal to `this` value.\n * If the [to] value is greater than `this` value the returned progression is empty.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun UShort.downTo(to: UShort): UIntProgression {\n return UIntProgression.fromClosedRange(this.toUInt(), to.toUInt(), -1)\n}\n\n/**\n * Returns a progression that goes over the same range in the opposite direction with the same step.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun UIntProgression.reversed(): UIntProgression {\n return UIntProgression.fromClosedRange(last, first, -step)\n}\n\n/**\n * Returns a progression that goes
over the same range in the opposite direction with the same step.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun

ULongProgression.reversed(): ULongProgression {\n   return ULongProgression.fromClosedRange(last, first, -step)\n}\n\n/**\n * Returns a progression that goes over the same range with the given step.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun UIntProgression.step(step: Int): UIntProgression {\n   checkStepIsPositive(step > 0, step)\n   return UIntProgression.fromClosedRange(first, last, if (this.step > 0) step else -step)\n}\n\n/**\n * Returns a progression that goes over the same range with the given step.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun ULongProgression.step(step: Long): ULongProgression {\n   checkStepIsPositive(step > 0, step)\n   return ULongProgression.fromClosedRange(first, last, if (this.step > 0)
 step else -step)\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun UByte.until(to: UByte): UIntRange {\n   if (to <= UByte.MIN_VALUE) return UIntRange.EMPTY\n   return this.toUInt() .. (to - 1u).toUInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun UInt.until(to: UInt): UIntRange {\n   if (to <= UInt.MIN_VALUE) return UIntRange.EMPTY\n   return this .. (to - 1u).toUInt()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or
 equal to `this` value, then the returned range is empty.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun ULong.until(to: ULong): ULongRange {\n   if (to <= ULong.MIN_VALUE) return ULongRange.EMPTY\n   return this .. (to - 1u).toULong()\n}\n\n/**\n * Returns a range from this value up to but excluding the specified [to] value.\n * \n * If the [to] value is less than or equal to `this` value, then the returned range is empty.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic infix fun UShort.until(to: UShort): UIntRange {\n   if (to <= UShort.MIN_VALUE) return UIntRange.EMPTY\n   return this.toUInt() .. (to - 1u).toUInt()\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic
 fun UInt.coerceAtLeast(minimumValue: UInt): UInt {\n   return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun ULong.coerceAtLeast(minimumValue: ULong): ULong {\n   return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic
 fun UByte.coerceAtLeast(minimumValue: UByte): UByte {\n   return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not less than the specified [minimumValue].\n * \n * @return this value if it's greater than or equal to the [minimumValue] or the [minimumValue] otherwise.\n * \n * @sample samples.comparisons.ComparableOps.coerceAtLeastUnsigned\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun UShort.coerceAtLeast(minimumValue: UShort): UShort {\n   return if (this < minimumValue) minimumValue else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample

samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UInt.coerceAtMost(maximumValue:
UInt): UInt {\n   return if (this > maximumValue) maximumValue else this\n}\n\n/**\n * Ensures that this value is
not greater than the specified [maximumValue].\n * \n * @return this value if it's less than or equal to the
[maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULong.coerceAtMost(maximumValue: ULong): ULong {\n   return if (this > maximumValue) maximumValue else
this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return this
value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UByte.coerceAtMost(maximumValue: UByte): UByte {\n   return if (this > maximumValue) maximumValue
else this\n}\n\n/**\n * Ensures that this value is not greater than the specified [maximumValue].\n * \n * @return
this value if it's less than or equal to the [maximumValue] or the [maximumValue] otherwise.\n * \n * @sample
samples.comparisons.ComparableOps.coerceAtMostUnsigned\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UShort.coerceAtMost(maximumValue: UShort): UShort {\n   return if (this > maximumValue) maximumValue
else this\n}\n\n/**\n * Ensures that this value lies in the specified range [minimumValue]..[maximumValue].\n * \n
* @return this value if it's in the range, or [minimumValue] if this value is less than [minimumValue], or
[maximumValue] if this value is greater than [maximumValue].\n * \n * @sample
samples.comparisons.ComparableOps.coerceInUnsigned\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UInt.coerceIn(minimumValue: UInt, maximumValue: UInt): UInt {\n   if (minimumValue
> maximumValue) throw IllegalArgumentException(\"Cannot coerce value to an empty range: maximum
$maximumValue is less than minimum $minimumValue.\")\n   if (this < minimumValue) return minimumValue\n
if (this > maximumValue) return maximumValue\n   return this\n}\n\n/**\n * Ensures that this value lies in the
specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or
[minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than
[maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceInUnsigned\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
ULong.coerceIn(minimumValue: ULong, maximumValue: ULong): ULong {\n   if (minimumValue >
maximumValue) throw IllegalArgumentException(\"Cannot coerce value to an empty range: maximum
$maximumValue is less than minimum $minimumValue.\")\n   if (this < minimumValue) return minimumValue\n
    if (this > maximumValue) return maximumValue\n   return this\n}\n\n/**\n * Ensures that this value lies in the
specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range, or
[minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than
[maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceInUnsigned\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun
UByte.coerceIn(minimumValue: UByte, maximumValue: UByte): UByte {\n   if (minimumValue >
maximumValue) throw IllegalArgumentException(\"Cannot coerce value to an empty range: maximum
$maximumValue is less than minimum $minimumValue.\")\n   if (this < minimumValue) return minimumValue\n
if (this > maximumValue) return maximumValue\n   return this\n}\n\n/**\n * Ensures that this value lies in the
specified range [minimumValue]..[maximumValue].\n * \n * @return this value if it's in the range,
 or [minimumValue] if this value is less than [minimumValue], or [maximumValue] if this value is greater than
[maximumValue].\n * \n * @sample samples.comparisons.ComparableOps.coerceInUnsigned\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun

UShort.coerceIn(minimumValue: UShort, maximumValue: UShort): UShort {\n if (minimumValue > maximumValue) throw IllegalArgumentException(\"Cannot coerce value to an empty range: maximum $maximumValue is less than minimum $minimumValue.\")\n if (this < minimumValue) return minimumValue\n if (this > maximumValue) return maximumValue\n return this\n}\n\n/**\n * Ensures that this value lies in the specified [range].\n * \n * @return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.\n * \n * @sample samples.comparisons.ComparableOps.coerceInUnsigned\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun UInt.coerceIn(range: ClosedRange<UInt>): UInt {\n if (range is ClosedFloatingPointRange) {\n return this.coerceIn<UInt>(range)\n }\n if (range.isEmpty()) throw IllegalArgumentException(\"Cannot coerce value to an empty range: $range.\")\n return when {\n this < range.start -> range.start\n this > range.endInclusive -> range.endInclusive\n else -> this\n }\n}\n\n/**\n * Ensures that this value lies in the specified [range].\n * \n * @return this value if it's in the [range], or `range.start` if this value is less than `range.start`, or `range.endInclusive` if this value is greater than `range.endInclusive`.\n * \n * @sample samples.comparisons.ComparableOps.coerceInUnsigned\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun ULong.coerceIn(range: ClosedRange<ULong>): ULong {\n if (range is ClosedFloatingPointRange) {\n return this.coerceIn<ULong>(range)\n }\n if (range.isEmpty()) throw IllegalArgumentException(\"Cannot coerce value to an empty range: $range.\")\n return when {\n this < range.start -> range.start\n this > range.endInclusive -> range.endInclusive\n else -> this\n }\n}\n\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"USequencesKt\")\n\npackage kotlin.sequences\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.random.*\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName(\"sumOfUInt\")\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Sequence<UInt>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName(\"sumOfULong\")\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Sequence<ULong>.sum(): ULong {\n var sum: ULong = 0uL\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName(\"sumOfUByte\")\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Sequence<UByte>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n/**\n * Returns the sum of all elements in the sequence.\n *\n * The operation is _terminal_.\n */\n@kotlin.jvm.JvmName(\"sumOfUShort\")\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Sequence<UShort>.sum(): UInt {\n var sum: UInt = 0u\n for (element in this) {\n sum += element\n }\n return sum\n}\n\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\n\npublic expect open class Error : Throwable {\n constructor()\n constructor(message: String?)\n constructor(message: String?, cause: Throwable?)\n constructor(cause: Throwable?)\n}\n\npublic expect open class Exception : Throwable {\n constructor()\n constructor(message: String?)\n constructor(message: String?, cause: Throwable?)\n constructor(cause:

```
Throwable?)\n}\n\npublic expect open class RuntimeException : Exception {\n    constructor()\n
constructor(message: String?)\n    constructor(message:
 String?, cause: Throwable?)\n    constructor(cause: Throwable?)\n}\n\npublic expect open class
IllegalArgumentException : RuntimeException {\n    constructor()\n    constructor(message: String?)\n
constructor(message: String?, cause: Throwable?)\n    constructor(cause: Throwable?)\n}\n\npublic expect open
class IllegalStateException : RuntimeException {\n    constructor()\n    constructor(message: String?)\n
constructor(message: String?, cause: Throwable?)\n    constructor(cause: Throwable?)\n}\n\npublic expect open
class IndexOutOfBoundsException : RuntimeException {\n    constructor()\n    constructor(message:
String?)\n}\n\npublic expect open class ConcurrentModificationException : RuntimeException {\n    constructor()\n
 constructor(message: String?)\n    constructor(message: String?, cause: Throwable?)\n    constructor(cause:
Throwable?)\n}\n\npublic expect open class UnsupportedOperationException : RuntimeException {\n
constructor()\n    constructor(message: String?)\n
    constructor(message: String?, cause: Throwable?)\n    constructor(cause: Throwable?)\n}\n\npublic expect open
class NumberFormatException : IllegalArgumentException {\n    constructor()\n    constructor(message:
String?)\n}\n\npublic expect open class NullPointerException : RuntimeException {\n    constructor()\n
constructor(message: String?)\n}\n\npublic expect open class ClassCastException : RuntimeException {\n
constructor()\n    constructor(message: String?)\n}\n\npublic expect open class AssertionError : Error {\n
constructor()\n    constructor(message: Any?)\n}\n\npublic expect open class NoSuchElementException :
RuntimeException {\n    constructor()\n    constructor(message: String?)\n}\n\n@SinceKotlin(\"1.3\")\npublic
expect open class ArithmeticException : RuntimeException {\n    constructor()\n    constructor(message:
String?)\n}\n\n@Deprecated(\"This exception type is not supposed to be thrown or caught in common code and will
be removed from kotlin-stdlib-common
 soon.\", level = DeprecationLevel.ERROR)\npublic expect open class NoWhenBranchMatchedException :
RuntimeException {\n    constructor()\n    constructor(message: String?)\n    constructor(message: String?, cause:
Throwable?)\n    constructor(cause: Throwable?)\n}\n\n@Deprecated(\"This exception type is not supposed to be
thrown or caught in common code and will be removed from kotlin-stdlib-common soon.\", level =
DeprecationLevel.ERROR)\npublic expect class UninitializedPropertyAccessException : RuntimeException {\n
constructor()\n    constructor(message: String?)\n    constructor(message: String?, cause: Throwable?)\n
constructor(cause: Throwable?)\n}\n\n/**\n * Thrown after invocation of a function or property that was expected to
return `Nothing`, but returned something instead.\n */\n@SinceKotlin(\"1.4\")\n@PublishedApi\ninternal class
KotlinNothingValueException : RuntimeException {\n    constructor() : super()\n    constructor(message: String?) :
super(message)\n    constructor(message:
 String?, cause: Throwable?) : super(message, cause)\n    constructor(cause: Throwable?) :
super(cause)\n}\n\n\n/**\n * Returns the detailed description of this throwable with its stack trace.\n *\n * The
detailed description includes:\n * - the short description (see [Throwable.toString]) of this throwable;\n * - the
complete stack trace;\n * - detailed descriptions of the exceptions that were [suppressed][suppressedExceptions] in
order to deliver this exception;\n * - the detailed description of each throwable in the [Throwable.cause] chain.\n
*/\n@SinceKotlin(\"1.4\")\npublic expect fun Throwable.stackTraceToString(): String\n\n/**\n * Prints the [detailed
description][Throwable.stackTraceToString] of this throwable to the standard output or standard error output.\n
*/\n@SinceKotlin(\"1.4\")\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\")\npublic expect fun
Throwable.printStackTrace(): Unit\n\n/**\n * When supported by the platform, adds the specified exception to the
 list of exceptions that were\n * suppressed in order to deliver this exception.\n
*/\n@SinceKotlin(\"1.4\")\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\")\npublic expect fun
Throwable.addSuppressed(exception: Throwable)\n\n/**\n * Returns a list of all exceptions that were suppressed in
order to deliver this exception.\n *\n * The list can be empty:\n * - if no exceptions were suppressed;\n * - if the
platform doesn't support suppressed exceptions;\n * - if this [Throwable] instance has disabled the suppression.\n
*/\n@SinceKotlin(\"1.4\")\npublic expect val Throwable.suppressedExceptions: List<Throwable>\n","/*\n *
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is
```

governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\nimport kotlin.annotation.AnnotationTarget.*\n\n/**\n * Gives a declaration (a function, a property or a class) specific name in JavaScript.\n */\n@Target(CLASS,
FUNCTION, PROPERTY, CONSTRUCTOR, PROPERTY_GETTER,
PROPERTY_SETTER)\n@OptionalExpectation\npublic expect annotation class JsName(val name: String)\n\n/**\n * Marks experimental JS export annotations.\n *\n * Note that behavior of these annotations will likely be changed in the future.\n *\n * Usages of such annotations will be reported as warnings unless an explicit opt-in with\n * the [OptIn] annotation, e.g. `@OptIn(ExperimentalJsExport::class)`,\n * or with the `-opt-in=kotlin.js.ExperimentalJsExport` compiler option is given.\n */\n@RequiresOptIn(level = RequiresOptIn.Level.WARNING)\n@MustBeDocumented\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin(\"1.4\")\npublic annotation class ExperimentalJsExport\n\n/**\n * Exports top-level declaration on JS platform.\n *\n * Compiled module exposes declarations that are marked with this annotation without name mangling.\n *\n * This annotation can be applied to either files or top-level declarations.\n *\n * It is currently prohibited to export the following kinds of declarations:\n *\n * * `expect` declarations\n * * inline functions with reified type parameters\n * * suspend functions\n * * secondary constructors without `@JsName`\n * * extension properties\n * * enum classes\n * * annotation classes\n *\n * Signatures of exported declarations must only contain \"exportable\" types:\n *\n * * `dynamic`, `Any`, `String`, `Boolean`, `Byte`, `Short`, `Int`, `Float`, `Double`\n * * `BooleanArray`, `ByteArray`, `ShortArray`, `IntArray`, `FloatArray`, `DoubleArray`\n * * `Array<exportable-type>`\n * * Function types with exportable parameters and return types\n * * `external` or `@JsExport` classes and interfaces\n * * Nullable counterparts of types above\n * * Unit return type. Must not be nullable\n *\n * This annotation is experimental, meaning that restrictions mentioned above are subject to change.\n */\n@ExperimentalJsExport\n@Retention(AnnotationRetention.BINARY)\n@Target(CLASS,
PROPERTY, FUNCTION, FILE)\n@SinceKotlin(\"1.4\")\n@OptionalExpectation\npublic expect annotation class JsExport()","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.io\n\n\n/** Prints the line separator to the standard output stream. */\npublic expect fun println()\n\n/** Prints the given [message] and the line separator to the standard output stream. */\npublic expect fun println(message: Any?)\n\n/** Prints the given [message] to the standard output stream. */\npublic expect fun print(message: Any?)\n\n/**\n * Reads a line of input from the standard input stream and returns it,\n * or throws a [RuntimeException] if EOF has already been reached when [readln] is called.\n *\n * LF or CRLF is treated as the line terminator. Line terminator is not included in the returned string.\n *\n * Currently this function
is not supported in Kotlin/JS and throws [UnsupportedOperationException].\n */\n@SinceKotlin(\"1.6\")\npublic expect fun readln(): String\n\n/**\n * Reads a line of input from the standard input stream and returns it,\n * or return `null` if EOF has already been reached when [readlnOrNull] is called.\n *\n * LF or CRLF is treated as the line terminator. Line terminator is not included in the returned string.\n *\n * Currently this function is not supported in Kotlin/JS and throws [UnsupportedOperationException].\n */\n@SinceKotlin(\"1.6\")\npublic expect fun readlnOrNull(): String?\n\ninternal class ReadAfterEOFException(message: String?) : RuntimeException(message)\n\n\ninternal expect interface Serializable\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\nimport kotlin.internal.PlatformDependent\n\n/**\n
 * Classes that inherit from this interface can be represented as a sequence of elements that can\n * be iterated over.\n * @param T the type of element being iterated over. The iterator is covariant in its element type.\n */\npublic interface Iterable<out T> {\n   /**\n   * Returns an iterator over the elements of this object.\n   */\n   public operator fun iterator(): Iterator<T>\n}\n\n/**\n * Classes that inherit from this interface can be represented as a sequence of elements that can\n * be iterated over and that supports removing elements during iteration.\n * @param T the type of element being iterated over. The mutable iterator is invariant in its element type.\n */\npublic interface MutableIterable<out T> : Iterable<T> {\n   /**\n   * Returns an iterator over the elements of this

sequence that supports removing elements during iteration.\n */\n override fun iterator(): MutableIterator<T>\n}\n\n/**\n * A generic collection of elements. Methods in this interface support only read-only access to the collection;\n * read/write access is supported through the [MutableCollection] interface.\n * @param E the type of elements contained in the collection. The collection is covariant in its element type.\n */\npublic interface Collection<out E> : Iterable<E> {\n // Query Operations\n /**\n * Returns the size of the collection.\n */\n public val size: Int\n\n /**\n * Returns `true` if the collection is empty (contains no elements), `false` otherwise.\n */\n public fun isEmpty(): Boolean\n\n /**\n * Checks if the specified element is contained in this collection.\n */\n public operator fun contains(element: @UnsafeVariance E): Boolean\n\n override fun iterator(): Iterator<E>\n\n // Bulk Operations\n /**\n * Checks if all elements in the specified collection are contained in this collection.\n */\n public fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean\n}\n\n/**\n * A generic collection of elements that supports adding and removing elements.\n *\n * @param E the type of elements contained in the collection. The mutable collection is invariant in its element type.\n */\npublic interface MutableCollection<E> : Collection<E>, MutableIterable<E> {\n // Query Operations\n override fun iterator(): MutableIterator<E>\n\n // Modification Operations\n /**\n * Adds the specified element to the collection.\n *\n * @return `true` if the element has been added, `false` if the collection does not support duplicates\n * and the element is already contained in the collection.\n */\n public fun add(element: E): Boolean\n\n /**\n * Removes a single instance of the specified element from this\n * collection, if it is present.\n *\n * @return `true` if the element has been successfully removed; `false` if it was not present in the collection.\n */\n public fun remove(element: E): Boolean\n\n // Bulk Modification Operations\n /**\n * Adds all of the elements of the specified collection to this collection.\n *\n * @return `true` if any of the specified elements was added to the collection, `false` if the collection was not modified.\n */\n public fun addAll(elements: Collection<E>): Boolean\n\n /**\n * Removes all of this collection's elements that are also contained in the specified collection.\n *\n * @return `true` if any of the specified elements was removed from the collection, `false` if the collection was not modified.\n */\n public fun removeAll(elements: Collection<E>): Boolean\n\n /**\n * Retains only the elements in this collection that are contained in the specified collection.\n *\n * @return `true` if any element was removed from the collection, `false` if the collection was not modified.\n */\n public fun retainAll(elements: Collection<E>): Boolean\n\n /**\n * Removes all elements from this collection.\n */\n public fun clear(): Unit\n}\n\n/**\n * A generic ordered collection of elements. Methods in this interface support only read-only access to the list;\n * read/write access is supported through the [MutableList] interface.\n * @param E the type of elements contained in the list. The list is covariant in its element type.\n */\npublic interface List<out E> : Collection<E> {\n // Query Operations\n\n override val size: Int\n override fun isEmpty(): Boolean\n override fun contains(element: @UnsafeVariance E): Boolean\n override fun iterator(): Iterator<E>\n\n // Bulk Operations\n override fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean\n\n // Positional Access Operations\n /**\n * Returns the element at the specified index in the list.\n */\n public operator fun get(index: Int): E\n\n // Search Operations\n /**\n * Returns the index of the first occurrence of the specified element in the list, or -1 if the specified\n * element is not contained in the list.\n */\n public fun indexOf(element: @UnsafeVariance E): Int\n\n /**\n * Returns the index of the last occurrence of the specified element in the list, or -1 if the specified\n * element is not contained in the list.\n */\n public fun lastIndexOf(element: @UnsafeVariance E): Int\n\n // List Iterators\n /**\n * Returns a list iterator over the elements in this list (in proper sequence).\n */\n public fun listIterator(): ListIterator<E>\n\n /**\n * Returns a list iterator over the elements in this list (in proper sequence), starting at the specified [index].\n */\n public fun listIterator(index: Int): ListIterator<E>\n\n // View\n /**\n * Returns a view of the portion of this list between the specified [fromIndex] (inclusive) and [toIndex] (exclusive).\n * The returned list is backed by this list, so non-structural changes in the returned list are

reflected in this list, and vice-versa.\n *\n * Structural changes in the base list make the behavior of the view undefined.\n */\n public fun subList(fromIndex: Int, toIndex: Int): List<E>\n}\n\n/**\n * A generic ordered collection of elements that supports adding and removing elements.\n * @param E the type of elements contained in the list. The mutable list is invariant in its element type.\n */\npublic interface MutableList<E> : List<E>, MutableCollection<E> {\n // Modification Operations\n /**\n * Adds the specified element to the end of this list.\n *\n * @return `true` because the list is always modified as the result of this operation.\n */\n override fun add(element: E): Boolean\n\n override fun remove(element: E): Boolean\n\n // Bulk Modification Operations\n /**\n * Adds all of the elements of the specified collection to the end of this list.\n *\n * The elements are appended in the order they appear in the [elements] collection.\n *\n * @return `true` if the list was changed as the result of the operation.\n */\n override fun addAll(elements: Collection<E>): Boolean\n\n /**\n * Inserts all of the elements of the specified collection [elements] into this list at the specified [index].\n *\n * @return `true` if the list was changed as the result of the operation.\n */\n public fun addAll(index: Int, elements: Collection<E>): Boolean\n\n override fun removeAll(elements: Collection<E>): Boolean\n override fun retainAll(elements: Collection<E>): Boolean\n override fun clear(): Unit\n\n // Positional Access Operations\n /**\n * Replaces the element at the specified position in this list with the specified element.\n *\n * @return the element previously at the specified position.\n */\n public operator fun set(index: Int, element: E): E\n\n /**\n * Inserts an element into the list at the specified [index].\n */\n public fun add(index: Int, element: E): Unit\n\n /**\n * Removes an element at the specified [index] from the list.\n *\n * @return the element that has been removed.\n */\n public fun removeAt(index: Int): E\n\n // List Iterators\n override fun listIterator(): MutableListIterator<E>\n\n override fun listIterator(index: Int): MutableListIterator<E>\n\n // View\n override fun subList(fromIndex: Int, toIndex: Int): MutableList<E>\n}\n\n/**\n * A generic unordered collection of elements that does not support duplicate elements.\n * Methods in this interface support only read-only access to the set;\n * read/write access is supported through the [MutableSet] interface.\n * @param E the type of elements contained in the set. The set is covariant in its element type.\n */\npublic interface Set<out E> : Collection<E> {\n // Query Operations\n\n override val size: Int\n override fun isEmpty(): Boolean\n override fun contains(element: @UnsafeVariance E): Boolean\n override fun iterator(): Iterator<E>\n\n // Bulk Operations\n override fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean\n}\n\n/**\n * A generic unordered collection of elements that does not support duplicate elements, and supports\n * adding and removing elements.\n * @param E the type of elements contained in the set. The mutable set is invariant in its element type.\n */\npublic interface MutableSet<E> : Set<E>, MutableCollection<E> {\n // Query Operations\n override fun iterator(): MutableIterator<E>\n\n // Modification Operations\n\n /**\n * Adds the specified element to the set.\n *\n * @return `true` if the element has been added, `false` if the element is already contained in the set.\n */\n override fun add(element: E): Boolean\n\n override fun remove(element: E): Boolean\n\n // Bulk Modification Operations\n\n override fun addAll(elements: Collection<E>): Boolean\n override fun removeAll(elements: Collection<E>): Boolean\n override fun retainAll(elements: Collection<E>): Boolean\n override fun clear(): Unit\n}\n\n/**\n * A collection that holds pairs of objects (keys and values) and supports efficiently retrieving\n * the value corresponding to each key. Map keys are unique; the map holds only one value for each key.\n * Methods in this interface support only read-only access to the map; read-write access is supported through\n * the [MutableMap] interface.\n * @param K the type of map keys. The map is invariant in its key type, as it\n * can accept key as a parameter (of [containsKey] for example) and return it in [keys] set.\n * @param V the type of map values. The map is covariant in its value type.\n */\npublic interface Map<K, out V> {\n // Query Operations\n /**\n * Returns the number of key/value pairs in the map.\n */\n public val size: Int\n\n /**\n * Returns `true` if the map is empty (contains no elements), `false` otherwise.\n */\n public fun isEmpty(): Boolean\n\n /**\n * Returns `true` if the map contains the specified [key].\n */\n public fun containsKey(key: K): Boolean\n\n /**\n * Returns `true` if the map maps one or more keys to the specified [value].\n */\n public fun containsValue(value: @UnsafeVariance V): Boolean\n\n /**\n *

Returns the value corresponding to the given [key], or `null` if such a key is not present in the map.\n */\n public operator fun get(key: K): V?\n\n /**\n * Returns the value corresponding to the given [key], or [defaultValue] if such a key is not present in the map.\n *\n * @since JDK 1.8\n */\n @SinceKotlin(\"1.1\")\n @PlatformDependent\n public fun getOrDefault(key: K, defaultValue: @UnsafeVariance V): V {\n // See default implementation in JDK sources\n throw NotImplementedError()\n }\n\n // Views\n /**\n * Returns a read-only [Set] of all keys in this map.\n */\n public val keys: Set<K>\n\n /**\n * Returns a read-only [Collection] of all values in this map. Note that this collection may contain duplicate values.\n */\n public val values: Collection<V>\n\n /**\n * Returns a read-only [Set] of all key/value pairs in this map.\n */\n public val entries: Set<Map.Entry<K, V>>\n\n /**\n * Represents a key/value pair held by a [Map].\n */\n public interface Entry<out K, out V> {\n /**\n * Returns the key of this key/value pair.\n */\n public val key: K\n\n /**\n * Returns the value of this key/value pair.\n */\n public val value: V\n }\n}\n\n/**\n * A modifiable collection that holds pairs of objects (keys and values) and supports efficiently retrieving\n * the value corresponding to each key. Map keys are unique; the map holds only one value for each key.\n * @param K the type of map keys. The map is invariant in its key type.\n * @param V the
type of map values. The mutable map is invariant in its value type.\n */\npublic interface MutableMap<K, V> : Map<K, V> {\n // Modification Operations\n /**\n * Associates the specified [value] with the specified [key] in the map.\n *\n * @return the previous value associated with the key, or `null` if the key was not present in the map.\n */\n public fun put(key: K, value: V): V?\n\n /**\n * Removes the specified key and its corresponding value from this map.\n *\n * @return the previous value associated with the key, or `null` if the key was not present in the map.\n */\n public fun remove(key: K): V?\n\n /**\n * Removes the entry for the specified key only if it is mapped to the specified value.\n *\n * @return true if entry was removed\n */\n @SinceKotlin(\"1.1\")\n @PlatformDependent\n public fun remove(key: K, value: V): Boolean {\n // See default implementation in JDK sources\n return true\n }\n\n // Bulk Modification Operations\n /**\n * Updates this map with key/value pairs from the specified map [from].\n */\n public fun putAll(from: Map<out K, V>): Unit\n\n /**\n * Removes all elements from this map.\n */\n public fun clear(): Unit\n\n // Views\n /**\n * Returns a [MutableSet] of all keys in this map.\n */\n override val keys: MutableSet<K>\n\n /**\n * Returns a [MutableCollection] of all values in this map. Note that this collection may contain duplicate values.\n */\n override val values: MutableCollection<V>\n\n /**\n * Returns a [MutableSet] of all key/value pairs in this map.\n */\n override val entries: MutableSet<MutableMap.MutableEntry<K, V>>\n\n /**\n * Represents a key/value pair held by a [MutableMap].\n */\n public interface MutableEntry<K, V> : Map.Entry<K, V> {\n /**\n * Changes the value associated with the key of this entry.\n *\n
* @return the previous value corresponding to the key.\n */\n public fun setValue(newValue: V): V\n }\n}\n","/*\n * Copyright 2010-2015 JetBrains s.r.o.\n *\n * Licensed under the Apache License, Version 2.0 (the \"License\");\n * you may not use this file except in compliance with the License.\n * You may obtain a copy of the License at\n *\n * http://www.apache.org/licenses/LICENSE-2.0\n *\n * Unless required by applicable law or agreed to in writing, software\n * distributed under the License is distributed on an \"AS IS\" BASIS,\n * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.\n * See the License for the specific language governing permissions and\n * limitations under the License.\n */\n\npackage kotlin\n\n/**\n * The type with only one value: the `Unit` object. This type corresponds to the `void` type in Java.\n */\npublic object Unit {\n override fun toString() = \"kotlin.Unit\"\n}\n","/*\n * Copyright 2010-2015 JetBrains s.r.o.\n *\n * Licensed under the Apache License, Version 2.0 (the \"License\");\n * you may not use this file except in compliance with the License.\n * You may obtain a copy of the License at\n *\n * http://www.apache.org/licenses/LICENSE-2.0\n *\n * Unless required by applicable law or agreed to in writing, software\n * distributed under the License is distributed on an \"AS IS\" BASIS,\n * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.\n * See the License for the specific language governing permissions and\n * limitations under the License.\n */\n\npackage kotlin.annotation\n\nimport

```
kotlin.annotation.AnnotationTarget.*\n\n/**\n * Contains the list of code elements which are the possible annotation targets\n */\npublic enum class AnnotationTarget {\n    /** Class, interface or object, annotation class is also included */\n    CLASS,\n    /** Annotation class only */\n    ANNOTATION_CLASS,\n    /** Generic type parameter */\n    TYPE_PARAMETER,\n    /** Property
 */\n    PROPERTY,\n    /** Field, including property's backing field */\n    FIELD,\n    /** Local variable */\n    LOCAL_VARIABLE,\n    /** Value parameter of a function or a constructor */\n    VALUE_PARAMETER,\n    /** Constructor only (primary or secondary) */\n    CONSTRUCTOR,\n    /** Function (constructors are not included) */\n    FUNCTION,\n    /** Property getter only */\n    PROPERTY_GETTER,\n    /** Property setter only */\n    PROPERTY_SETTER,\n    /** Type usage */\n    TYPE,\n    /** Any expression */\n    EXPRESSION,\n    /** File */\n    FILE,\n    /** Type alias */\n    @SinceKotlin(\"1.1\")\n    TYPEALIAS\n}\n\n/**\n * Contains the list of possible annotation's retentions.\n *\n * Determines how an annotation is stored in binary output.\n */\npublic enum class AnnotationRetention {\n    /** Annotation isn't stored in binary output */\n    SOURCE,\n    /** Annotation is stored in binary output, but invisible for reflection */\n    BINARY,\n    /** Annotation is stored
 in binary output and visible for reflection (default retention) */\n    RUNTIME\n}\n\n/**\n * This meta-annotation indicates the kinds of code elements which are possible targets of an annotation.\n *\n * If the target meta-annotation is not present on an annotation declaration, the annotation is applicable to the following elements:\n * [CLASS], [PROPERTY], [FIELD], [LOCAL_VARIABLE], [VALUE_PARAMETER], [CONSTRUCTOR], [FUNCTION], [PROPERTY_GETTER], [PROPERTY_SETTER].\n *\n * @property allowedTargets list of allowed annotation targets\n */\n@Target(AnnotationTarget.ANNOTATION_CLASS)\n@MustBeDocumented\npublic annotation class Target(vararg val allowedTargets: AnnotationTarget)\n\n/**\n * This meta-annotation determines whether an annotation is stored in binary output and visible for reflection. By default, both are true.\n *\n * @property value necessary annotation retention (RUNTIME, BINARY or SOURCE)\n
*/\n@Target(AnnotationTarget.ANNOTATION_CLASS)\npublic annotation class Retention(val
 value: AnnotationRetention = AnnotationRetention.RUNTIME)\n\n/**\n * This meta-annotation determines that an annotation is applicable twice or more on a single code element\n
*/\n@Target(AnnotationTarget.ANNOTATION_CLASS)\npublic annotation class Repeatable\n\n/**\n * This meta-annotation determines that an annotation is a part of public API and therefore should be included in the generated\n * documentation for the element to which the annotation is applied.\n
*/\n@Target(AnnotationTarget.ANNOTATION_CLASS)\npublic annotation class MustBeDocumented\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@JsName(\"arrayIterator\")\ninternal fun arrayIterator(array: dynamic, type: String?) = when (type) {\n    null -> {\n        val arr: Array<dynamic> = array\n        object : Iterator<dynamic> {\n            var index =
 0\n            override fun hasNext() = index < arr.size\n            override fun next() = if (index < arr.size) arr[index++] else throw NoSuchElementException(\"$index\")\n        }\n    }\n    \"BooleanArray\" -> booleanArrayIterator(array)\n    \"ByteArray\" -> byteArrayIterator(array)\n    \"ShortArray\" -> shortArrayIterator(array)\n    \"CharArray\" -> charArrayIterator(array)\n    \"IntArray\" -> intArrayIterator(array)\n    \"LongArray\" -> longArrayIterator(array)\n    \"FloatArray\" -> floatArrayIterator(array)\n    \"DoubleArray\" -> doubleArrayIterator(array)\n    else -> throw IllegalStateException(\"Unsupported type argument for arrayIterator: $type\")\n}\n\n@JsName(\"booleanArrayIterator\")\ninternal fun booleanArrayIterator(array: BooleanArray) = object : BooleanIterator() {\n    var index = 0\n    override fun hasNext() = index < array.size\n    override fun nextBoolean() = if (index < array.size) array[index++] else throw NoSuchElementException(\"$index\")\n}\n\n@JsName(\"byteArrayIterator\")\ninternal
 fun byteArrayIterator(array: ByteArray) = object : ByteIterator() {\n    var index = 0\n    override fun hasNext() = index < array.size\n    override fun nextByte() = if (index < array.size) array[index++] else throw NoSuchElementException(\"$index\")\n}\n\n@JsName(\"shortArrayIterator\")\ninternal fun shortArrayIterator(array: ShortArray) = object : ShortIterator() {\n    var index = 0\n    override fun hasNext() =
```

index < array.size\n   override fun nextShort() = if (index < array.size) array[index++] else throw NoSuchElementException(\"$index\")\n}\n\n@JsName(\"charArrayIterator\")\ninternal fun charArrayIterator(array: CharArray) = object : CharIterator() {\n   var index = 0\n   override fun hasNext() = index < array.size\n   override fun nextChar() = if (index < array.size) array[index++] else throw NoSuchElementException(\"$index\")\n}\n\n@JsName(\"intArrayIterator\")\ninternal fun intArrayIterator(array: IntArray) = object : IntIterator()
 {\n   var index = 0\n   override fun hasNext() = index < array.size\n   override fun nextInt() = if (index < array.size) array[index++] else throw NoSuchElementException(\"$index\")\n}\n\n@JsName(\"floatArrayIterator\")\ninternal fun floatArrayIterator(array: FloatArray) = object : FloatIterator() {\n   var index = 0\n   override fun hasNext() = index < array.size\n   override fun nextFloat() = if (index < array.size) array[index++] else throw NoSuchElementException(\"$index\")\n}\n\n@JsName(\"doubleArrayIterator\")\ninternal fun doubleArrayIterator(array: DoubleArray) = object : DoubleIterator() {\n   var index = 0\n   override fun hasNext() = index < array.size\n   override fun nextDouble() = if (index < array.size) array[index++] else throw NoSuchElementException(\"$index\")\n}\n\n@JsName(\"longArrayIterator\")\ninternal fun longArrayIterator(array: LongArray) = object : LongIterator() {\n   var index = 0\n   override fun hasNext() = index < array.size\n override fun
 nextLong() = if (index < array.size) array[index++] else throw NoSuchElementException(\"$index\")\n}\n\n@JsName(\"PropertyMetadata\")\ninternal class PropertyMetadata(@JsName(\"callableName\") val name: String)\n\n@JsName(\"noWhenBranchMatched\")\ninternal fun noWhenBranchMatched(): Nothing = throw NoWhenBranchMatchedException()\n\n@JsName(\"subSequence\")\ninternal fun subSequence(c: CharSequence, startIndex: Int, endIndex: Int): CharSequence {\n   if (c is String) {\n   return c.substring(startIndex, endIndex)\n   } else {\n   return c.asDynamic().`subSequence_vux9f0$`(startIndex, endIndex)\n   }\n}\n\n@JsName(\"captureStack\")\ninternal fun captureStack(@Suppress(\"UNUSED_PARAMETER\") baseClass: JsClass<in Throwable>, instance: Throwable) {\n   if (js(\"Error\").captureStackTrace) {\n   // Using uncropped stack traces due to KT-37563.\n   // Precise stack traces are implemented in JS IR compiler and stdlib\n   js(\"Error\").captureStackTrace(instance);\n   } else {\n   instance.asDynamic().stack = js(\"new Error()\").stack;\n   }\n}\n\n@JsName(\"newThrowable\")\ninternal fun newThrowable(message: String?, cause: Throwable?): Throwable {\n   val throwable = js(\"new Error()\")\n   throwable.message = if (jsTypeOf(message) == \"undefined\") {\n   if (cause != null) cause.toString() else null\n   } else {\n   message\n   }\n   throwable.cause = cause\n   throwable.name = \"Throwable\"\n   return throwable\n}\n\n@JsName(\"BoxedChar\")\ninternal class BoxedChar(val c: Int) : Comparable<Int> {\n   override fun equals(other: Any?): Boolean {\n   return other is BoxedChar && c == other.c\n   }\n\n   override fun hashCode(): Int {\n   return c\n   }\n\n   override fun toString(): String {\n   return js(\"this.c\").unsafeCast<Char>().toString()\n   }\n\n   override fun compareTo(other: Int): Int {\n   return js(\"this.c - other\").unsafeCast<Int>()\n   }\n\n   @JsName(\"valueOf\")\n   public fun valueOf(): Int {\n   return c\n   }\n}\n\n@kotlin.internal.InlineOnly\ninternal inline fun <T> concat(args: Array<T>): T {\n   val typed = js(\"Array\")(args.size)\n   for (i in args.indices) {\n   val arr = args[i]\n   if (arr !is Array<*>) {\n   typed[i] = js(\"[]\").slice.call(arr)\n   } else {\n   typed[i] = arr\n   }\n   }\n   return js(\"[]\").concat.apply(js(\"[]\"), typed);\n}\n\n/** Concat regular Array's and TypedArray's into an Array.\n */\n@PublishedApi\n@JsName(\"arrayConcat\")\n@Suppress(\"UNUSED_PARAMETER\")\ninternal fun <T> arrayConcat(a: T, b: T): T {\n   return concat(js(\"arguments\"))\n}\n\n/** Concat primitive arrays. Main use: prepare vararg arguments.\n * For compatibility with 1.1.0 the arguments may be a mixture of Array's and TypedArray's.\n *\n * If the first argument is TypedArray (Byte-, Short-, Char-, Int-, Float-, and DoubleArray) returns a TypedArray, otherwise an Array.\n * If the first argument

has the $type$ property (Boolean-, Char-, and LongArray) copy its value to result.$type$.\n * If the first argument is a regular Array without the $type$ property default to arrayConcat.\n */\n@PublishedApi\n@JsName(\"primitiveArrayConcat\")\n@Suppress(\"UNUSED_PARAMETER\")\ninternal fun <T> primitiveArrayConcat(a: T, b: T): T {\n   val args: Array<T> = js(\"arguments\")\n   if (a is Array<*> && a.asDynamic().`$type$` === undefined) {\n   return concat(args)\n   } else {\n   var size = 0\n   for (i in args.indices) {\n   size += args[i].asDynamic().length as Int\n   }\n   val result = js(\"new a.constructor(size)\")\n   kotlin.copyArrayType(a, result)\n   size = 0\n   for (i in args.indices) {\n   val arr = args[i].asDynamic()\n   for (j in 0 until arr.length) {\n   result[size++] = arr[j]\n   }\n   }\n   return result\n   }\n}\n\n@JsName(\"booleanArrayOf\")\ninternal fun booleanArrayOf() = withType(\"BooleanArray\", js(\"[].slice.call(arguments)\"))\n\n@JsName(\"charArrayOf\")\ninternal fun charArrayOf() = withType(\"CharArray\", js(\"new Uint16Array(arguments)\"))\n\n@JsName(\"longArrayOf\")\ninternal fun longArrayOf() = withType(\"LongArray\", js(\"[].slice.call(arguments)\"))\n\n@JsName(\"withType\")\n@kotlin.internal.InlineOnly\ninternal inline fun withType(type: String, array: dynamic): dynamic {\n   array.`$type$` = type\n   return array\n}","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n/**\n * Function corresponding to JavaScript's `typeof` operator\n */\n@kotlin.internal.InlineOnly\n@Suppress(\"UNUSED_PARAMETER\")\npublic inline fun jsTypeOf(a: Any?): String = js(\"typeof a\")\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:Suppress(\"UNUSED_PARAMETER\", \"NOTHING_TO_INLINE\")\n\npackage kotlin\n\n/**\n * Returns an empty array of the specified type [T].\n */\npublic inline fun <T> emptyArray(): Array<T> = js(\"[]\")\n\n@library\npublic fun <T> arrayOf(vararg elements: T): Array<T> = definedExternally\n\n@library\npublic fun doubleArrayOf(vararg elements: Double): DoubleArray = definedExternally\n\n@library\npublic fun floatArrayOf(vararg elements: Float): FloatArray = definedExternally\n\n@library\npublic fun longArrayOf(vararg elements: Long): LongArray = definedExternally\n\n@library\npublic fun intArrayOf(vararg elements: Int): IntArray = definedExternally\n\n@library\npublic fun charArrayOf(vararg elements: Char): CharArray = definedExternally\n\n@library\npublic fun shortArrayOf(vararg elements: Short): ShortArray = definedExternally\n\n@library\npublic fun byteArrayOf(vararg elements: Byte): ByteArray = definedExternally\n\n@library\npublic fun booleanArrayOf(vararg elements: Boolean): BooleanArray = definedExternally\n\n/**\n * Creates a new instance of the [Lazy] that uses the specified initialization function [initializer].\n */\npublic actual fun <T> lazy(initializer: () -> T): Lazy<T> = UnsafeLazyImpl(initializer)\n\n/**\n * Creates a new instance of the [Lazy] that uses the specified initialization function [initializer].\n *\n * The [mode] parameter is ignored. */\npublic actual fun <T> lazy(mode: LazyThreadSafetyMode, initializer: () -> T): Lazy<T> = UnsafeLazyImpl(initializer)\n\n/**\n * Creates a new instance of the [Lazy] that uses the specified initialization function [initializer].\n *\n * The [lock] parameter is ignored.\n */\npublic actual fun <T> lazy(lock: Any?, initializer: () -> T): Lazy<T> = UnsafeLazyImpl(initializer)\n\n\ninternal fun fillFrom(src: dynamic, dst: dynamic): dynamic {\n   val srcLen: Int = src.length\n   val dstLen: Int = dst.length\n   var index: Int = 0\n   while (index < srcLen && index < dstLen) dst[index] = src[index++]\n   return dst\n}\n\n\ninternal fun arrayCopyResize(source: dynamic, newSize: Int, defaultValue: Any?): dynamic {\n   val result = source.slice(0, newSize)\n   copyArrayType(source, result)\n   var index: Int = source.length\n   if (newSize > index) {\n   result.length = newSize\n   while (index < newSize) result[index++] = defaultValue\n   }\n   return result\n}\n\ninternal fun <T> arrayPlusCollection(array: dynamic, collection: Collection<T>): dynamic {\n   val result = array.slice()\n   result.length += collection.size\n

copyArrayType(array, result)\n   var index: Int = array.length\n   for (element in collection) result[index++] = element\n   return result\n}\n\ninternal fun <T> fillFromCollection(dst: dynamic, startIndex: Int, collection: Collection<T>): dynamic {\n   var index = startIndex\n   for (element in collection) dst[index++] = element\n   return dst\n}\n\ninternal inline fun copyArrayType(from: dynamic, to: dynamic) {\n   if (from.`$type$` !== undefined) {\n   to.`$type$` = from.`$type$`\n   }\n}\n\ninternal inline fun jsIsType(obj: dynamic, jsClass: dynamic) = js(\"Kotlin\").isType(obj, jsClass)","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\n/**\n * Creates a Char with the specified [code].\n *\n * @sample samples.text.Chars.charFromCode\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic actual inline fun Char(code: UShort): Char {\n   return code.toInt().toChar()\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines\n\nimport kotlin.coroutines.intrinsics.COROUTINE_SUSPENDED\n\n@SinceKotlin(\"1.3\")\n@JsName(\"CoroutineImpl\")\ninternal abstract class CoroutineImpl(private val resultContinuation: Continuation<Any?>) : Continuation<Any?> {\n   protected var state = 0\n   protected var exceptionState = 0\n   protected var result: Any? = null\n   protected var exception: Throwable? = null\n   protected var finallyPath: Array<Int>? = null\n\n   public override val context: CoroutineContext = resultContinuation.context\n\n   private var intercepted_: Continuation<Any?>? = null\n\n   public fun intercepted(): Continuation<Any?> =\n   intercepted_\n   ?: (context[ContinuationInterceptor]?.interceptContinuation(this) ?: this)\n   .also { intercepted_ = it }\n\n   override fun resumeWith(result: Result<Any?>) {\n   var current = this\n   var currentResult: Any? = result.getOrNull()\n   var currentException: Throwable? = result.exceptionOrNull()\n\n   // This loop unrolls recursion in current.resumeWith(param) to make saner and shorter stack traces on resume\n   while (true) {\n   with(current) {\n   val completion = resultContinuation\n\n   // Set result and exception fields in the current continuation\n   if (currentException == null) {\n   this.result = currentResult\n   } else {\n   state = exceptionState\n   exception = currentException\n   }\n\n   try {\n   val outcome = doResume()\n   if (outcome === COROUTINE_SUSPENDED) return\n   currentResult = outcome\n   currentException = null\n   } catch (exception: dynamic) { // Catch all exceptions\n   currentResult = null\n   currentException = exception.unsafeCast<Throwable>()\n   }\n\n   releaseIntercepted() // this state machine instance is terminating\n\n   if (completion is CoroutineImpl) {\n   // unrolling recursion via loop\n   current = completion\n   } else {\n   // top-level completion reached -- invoke and return\n   currentException?.let {\n   completion.resumeWithException(it)\n   } ?: completion.resume(currentResult)\n   return\n   }\n   }\n   }\n   }\n\n   private fun releaseIntercepted() {\n   val intercepted = intercepted_\n   if (intercepted != null && intercepted !== this) {\n   context[ContinuationInterceptor]!!.releaseInterceptedContinuation(intercepted)\n   }\n   this.intercepted_ = CompletedContinuation // just in case\n   }\n\n   protected abstract fun doResume(): Any?\n}\n\ninternal object CompletedContinuation : Continuation<Any?> {\n   override val context: CoroutineContext\n   get() = error(\"This continuation is already complete\")\n\n   override fun resumeWith(result: Result<Any?>) {\n   error(\"This continuation is already complete\")\n   }\n\n   override fun toString(): String = \"This continuation is already complete\"\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:Suppress(\"UNCHECKED_CAST\", \"RedundantVisibilityModifier\")\n\npackage kotlin\n\nimport kotlin.contracts.*\nimport kotlin.internal.InlineOnly\nimport kotlin.jvm.JvmField\nimport

kotlin.jvm.JvmInline\nimport kotlin.jvm.JvmName\n\n/**\n * A discriminated union that encapsulates a successful outcome with a value of type [T]\n * or a failure with an arbitrary [Throwable] exception.\n */\n@SinceKotlin(\"1.3\")\n@JvmInline\npublic
value class Result<out T> @PublishedApi internal constructor(\n @PublishedApi\n internal val value: Any?\n) : Serializable {\n // discovery\n\n /**\n * Returns `true` if this instance represents a successful outcome.\n * In this case [isFailure] returns `false`.\n */\n public val isSuccess: Boolean get() = value !is Failure\n\n /**\n * Returns `true` if this instance represents a failed outcome.\n * In this case [isSuccess] returns `false`.\n */\n public val isFailure: Boolean get() = value is Failure\n\n // value & exception retrieval\n\n /**\n * Returns the encapsulated value if this instance represents [success][Result.isSuccess] or `null`\n * if it is [failure][Result.isFailure].\n *\n * This function is a shorthand for `getOrElse { null }` (see [getOrElse]) or\n * `fold(onSuccess = { it }, onFailure = { null })` (see [fold]).\n */\n @InlineOnly\n public inline fun getOrNull(): T? =\n when
{\n isFailure -> null\n else -> value as T\n }\n\n /**\n * Returns the encapsulated [Throwable] exception if this instance represents [failure][isFailure] or `null`\n * if it is [success][isSuccess].\n *\n * This function is a shorthand for `fold(onSuccess = { null }, onFailure = { it })` (see [fold]).\n */\n public fun exceptionOrNull(): Throwable? =\n when (value) {\n is Failure -> value.exception\n else -> null\n }\n\n /**\n * Returns a string `Success(v)` if this instance represents [success][Result.isSuccess]\n * where `v` is a string representation of the value or a string `Failure(x)` if\n * it is [failure][isFailure] where `x` is a string representation of the exception.\n */\n public override fun toString(): String =\n when (value) {\n is Failure -> value.toString() // \"Failure($exception)\"\n else -> \"Success($value)\"\n }\n\n // companion with constructors\n\n /**\n * Companion object for [Result] class that contains its constructor functions\n * [success] and [failure].\n */\n public companion object {\n /**\n * Returns an instance that encapsulates the given [value] as successful value.\n */\n @Suppress(\"INAPPLICABLE_JVM_NAME\")\n @InlineOnly\n @JvmName(\"success\")\n public inline fun <T> success(value: T): Result<T> =\n Result(value)\n\n /**\n * Returns an instance that encapsulates the given [Throwable] [exception] as failure.\n */\n @Suppress(\"INAPPLICABLE_JVM_NAME\")\n @InlineOnly\n @JvmName(\"failure\")\n public inline fun <T> failure(exception: Throwable): Result<T> =\n Result(createFailure(exception))\n }\n\n internal class Failure(\n @JvmField\n val exception: Throwable\n) : Serializable {\n override fun equals(other:
Any?): Boolean = other is Failure && exception == other.exception\n override fun hashCode(): Int = exception.hashCode()\n override fun toString(): String = \"Failure($exception)\"\n }\n}\n\n/**\n * Creates an instance of internal marker [Result.Failure] class to\n * make sure that this class is not exposed in ABI.\n */\n@PublishedApi\n@SinceKotlin(\"1.3\")\ninternal fun createFailure(exception: Throwable): Any =\n Result.Failure(exception)\n\n/**\n * Throws exception if the result is failure. This internal function minimizes\n * inlined bytecode for [getOrThrow] and makes sure that in the future we can\n * add some exception-augmenting logic here (if needed).\n */\n@PublishedApi\n@SinceKotlin(\"1.3\")\ninternal fun Result<*>.throwOnFailure() {\n if (value is Result.Failure) throw value.exception\n}\n\n/**\n * Calls the specified function [block] and returns its encapsulated result if invocation was successful,\n * catching any [Throwable] exception that was thrown
from the [block] function execution and encapsulating it as a failure.\n */\n@InlineOnly\n@SinceKotlin(\"1.3\")\npublic inline fun <R> runCatching(block: () -> R): Result<R> {\n return try {\n Result.success(block())\n } catch (e: Throwable) {\n Result.failure(e)\n }\n}\n\n/**\n * Calls the specified function [block] with `this` value as its receiver and returns its encapsulated result if invocation was successful,\n * catching any [Throwable] exception that was thrown from the [block] function execution and encapsulating it as a failure.\n */\n@InlineOnly\n@SinceKotlin(\"1.3\")\npublic inline fun <T, R> T.runCatching(block: T.() -> R): Result<R> {\n return try {\n Result.success(block())\n } catch (e: Throwable) {\n Result.failure(e)\n }\n}\n\n// -- extensions ---\n\n/**\n * Returns the encapsulated value if this instance represents [success][Result.isSuccess] or throws the encapsulated [Throwable] exception\n * if it is

[failure][Result.isFailure].\n
 *\n * This function is a shorthand for `getOrElse { throw it }` (see [getOrElse]).\n
*/\n@InlineOnly\n@SinceKotlin(\"1.3\")\npublic inline fun <T> Result<T>.getOrThrow(): T {\n
throwOnFailure()\n   return value as T\n}\n\n/**\n * Returns the encapsulated value if this instance represents
[success][Result.isSuccess] or the\n * result of [onFailure] function for the encapsulated [Throwable] exception if it
is [failure][Result.isFailure].\n *\n * Note, that this function rethrows any [Throwable] exception thrown by
[onFailure] function.\n *\n * This function is a shorthand for `fold(onSuccess = { it }, onFailure = onFailure)` (see
[fold]).\n */\n@InlineOnly\n@SinceKotlin(\"1.3\")\npublic inline fun <R, T : R> Result<T>.getOrElse(onFailure:
(exception: Throwable) -> R): R {\n   contract {\n   callsInPlace(onFailure,
InvocationKind.AT_MOST_ONCE)\n   }\n   return when (val exception = exceptionOrNull()) {\n   null ->
value as T\n   else -> onFailure(exception)\n
    }\n}\n\n/**\n * Returns the encapsulated value if this instance represents [success][Result.isSuccess] or the\n *
[defaultValue] if it is [failure][Result.isFailure].\n *\n * This function is a shorthand for `getOrElse { defaultValue
}` (see [getOrElse]).\n */\n@InlineOnly\n@SinceKotlin(\"1.3\")\npublic inline fun <R, T : R>
Result<T>.getOrDefault(defaultValue: R): R {\n   if (isFailure) return defaultValue\n   return value as
T\n}\n\n/**\n * Returns the result of [onSuccess] for the encapsulated value if this instance represents
[success][Result.isSuccess]\n * or the result of [onFailure] function for the encapsulated [Throwable] exception if it
is [failure][Result.isFailure].\n *\n * Note, that this function rethrows any [Throwable] exception thrown by
[onSuccess] or by [onFailure] function.\n */\n@InlineOnly\n@SinceKotlin(\"1.3\")\npublic inline fun <R, T>
Result<T>.fold(\n   onSuccess: (value: T) -> R,\n   onFailure: (exception: Throwable) -> R\n): R {\n   contract
 {\n   callsInPlace(onSuccess, InvocationKind.AT_MOST_ONCE)\n   callsInPlace(onFailure,
InvocationKind.AT_MOST_ONCE)\n   }\n   return when (val exception = exceptionOrNull()) {\n   null ->
onSuccess(value as T)\n   else -> onFailure(exception)\n   }\n}\n\n// transformation\n\n/**\n * Returns the
encapsulated result of the given [transform] function applied to the encapsulated value\n * if this instance represents
[success][Result.isSuccess] or the\n * original encapsulated [Throwable] exception if it is
[failure][Result.isFailure].\n *\n * Note, that this function rethrows any [Throwable] exception thrown by
[transform] function.\n * See [mapCatching] for an alternative that encapsulates exceptions.\n
*/\n@InlineOnly\n@SinceKotlin(\"1.3\")\npublic inline fun <R, T> Result<T>.map(transform: (value: T) -> R):
Result<R> {\n   contract {\n   callsInPlace(transform, InvocationKind.AT_MOST_ONCE)\n   }\n   return when
{\n   isSuccess -> Result.success(transform(value
 as T))\n   else -> Result(value)\n   }\n}\n\n/**\n * Returns the encapsulated result of the given [transform]
function applied to the encapsulated value\n * if this instance represents [success][Result.isSuccess] or the\n *
original encapsulated [Throwable] exception if it is [failure][Result.isFailure].\n *\n * This function catches any
[Throwable] exception thrown by [transform] function and encapsulates it as a failure.\n * See [map] for an
alternative that rethrows exceptions from `transform` function.\n */\n@InlineOnly\n@SinceKotlin(\"1.3\")\npublic
inline fun <R, T> Result<T>.mapCatching(transform: (value: T) -> R): Result<R> {\n   return when {\n
isSuccess -> runCatching { transform(value as T) }\n   else -> Result(value)\n   }\n}\n\n/**\n * Returns the
encapsulated result of the given [transform] function applied to the encapsulated [Throwable] exception\n * if this
instance represents [failure][Result.isFailure] or the\n * original encapsulated value
 if it is [success][Result.isSuccess].\n *\n * Note, that this function rethrows any [Throwable] exception thrown by
[transform] function.\n * See [recoverCatching] for an alternative that encapsulates exceptions.\n
*/\n@InlineOnly\n@SinceKotlin(\"1.3\")\npublic inline fun <R, T : R> Result<T>.recover(transform: (exception:
Throwable) -> R): Result<R> {\n   contract {\n   callsInPlace(transform, InvocationKind.AT_MOST_ONCE)\n
}\n   return when (val exception = exceptionOrNull()) {\n   null -> this\n   else ->
Result.success(transform(exception))\n   }\n}\n\n/**\n * Returns the encapsulated result of the given [transform]
function applied to the encapsulated [Throwable] exception\n * if this instance represents [failure][Result.isFailure]
or the\n * original encapsulated value if it is [success][Result.isSuccess].\n *\n * This function catches any
[Throwable] exception thrown by [transform] function and encapsulates it as a failure.\n * See [recover] for an

alternative
that rethrows exceptions.\n */\n@InlineOnly\n@SinceKotlin(\"1.3\")\npublic inline fun <R, T : R> Result<T>.recoverCatching(transform: (exception: Throwable) -> R): Result<R> {\n return when (val exception = exceptionOrNull()) {\n null -> this\n else -> runCatching { transform(exception) }\n }\n}\n\n// \"peek\" onto value/exception and pipe\n\n/**\n * Performs the given [action] on the encapsulated [Throwable] exception if this instance represents [failure][Result.isFailure].\n * Returns the original `Result` unchanged.\n */\n@InlineOnly\n@SinceKotlin(\"1.3\")\npublic inline fun <T> Result<T>.onFailure(action: (exception: Throwable) -> Unit): Result<T> {\n contract {\n callsInPlace(action, InvocationKind.AT_MOST_ONCE)\n }\n exceptionOrNull()?.let { action(it) }\n return this\n}\n\n/**\n * Performs the given [action] on the encapsulated value if this instance represents [success][Result.isSuccess].\n * Returns the original `Result` unchanged.\n
/\n@InlineOnly\n@SinceKotlin(\"1.3\")\npublic inline fun <T> Result<T>.onSuccess(action: (value: T) -> Unit): Result<T> {\n contract {\n callsInPlace(action, InvocationKind.AT_MOST_ONCE)\n }\n if (isSuccess) action(value as T)\n return this\n}\n\n// ----------\n","/\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines\n\nimport kotlin.contracts.*\nimport kotlin.coroutines.intrinsics.*\nimport kotlin.internal.InlineOnly\n\n/**\n * Interface representing a continuation after a suspension point that returns a value of type `T`.\n */\n@SinceKotlin(\"1.3\")\npublic interface Continuation<in T> {\n /**\n * The context of the coroutine that corresponds to this continuation.\n */\n public val context: CoroutineContext\n\n /**\n * Resumes the execution of the
corresponding coroutine passing a successful or failed [result] as the\n * return value of the last suspension point.\n */\n public fun resumeWith(result: Result<T>)\n}\n\n/**\n * Classes and interfaces marked with this annotation are restricted when used as receivers for extension\n * `suspend` functions. These `suspend` extensions can only invoke other member or extension `suspend` functions on this particular\n * receiver and are restricted from calling arbitrary suspension functions.\n
*/\n@SinceKotlin(\"1.3\")\n@Target(AnnotationTarget.CLASS)\n@Retention(AnnotationRetention.BINARY)\npublic annotation class RestrictsSuspension\n\n/**\n * Resumes the execution of the corresponding coroutine passing [value] as the return value of the last suspension point.\n */\n@SinceKotlin(\"1.3\")\n@InlineOnly\npublic inline fun <T> Continuation<T>.resume(value: T): Unit =\n resumeWith(Result.success(value))\n\n/**\n * Resumes the execution of the corresponding coroutine so that the
[exception] is re-thrown right after the\n * last suspension point.\n
*/\n@SinceKotlin(\"1.3\")\n@InlineOnly\npublic inline fun <T> Continuation<T>.resumeWithException(exception: Throwable): Unit =\n resumeWith(Result.failure(exception))\n\n/**\n * Creates a [Continuation] instance with the given [context] and implementation of [resumeWith] method.\n */\n@SinceKotlin(\"1.3\")\n@InlineOnly\npublic inline fun <T> Continuation(\n context: CoroutineContext,\n crossinline resumeWith: (Result<T>) -> Unit\n): Continuation<T> =\n object : Continuation<T> {\n override val context: CoroutineContext\n get() = context\n\n override fun resumeWith(result: Result<T>) =\n resumeWith(result)\n }\n\n/**\n * Creates a coroutine without a receiver and with result type [T].\n * This function creates a new, fresh instance of suspendable computation every time it is invoked.\n *\n * To start executing the created coroutine, invoke `resume(Unit)` on the returned
[Continuation] instance.\n * The [completion] continuation is invoked when the coroutine completes with a result or an exception.\n * Subsequent invocation of any resume function on the resulting continuation will produce an [IllegalStateException].\n */\n@SinceKotlin(\"1.3\")\n@Suppress(\"UNCHECKED_CAST\")\npublic fun <T> (suspend () -> T).createCoroutine(\n completion: Continuation<T>\n): Continuation<Unit> =\n SafeContinuation(createCoroutineUnintercepted(completion).intercepted(), COROUTINE_SUSPENDED)\n\n/**\n * Creates a coroutine with receiver type [R] and result type [T].\n * This function creates a new, fresh instance of suspendable computation every time it is invoked.\n *\n * To start executing the created coroutine, invoke

`resume(Unit)` on the returned [Continuation] instance.\n * The [completion] continuation is invoked when the coroutine completes with a result or an exception.\n * Subsequent invocation of any resume function on the resulting continuation will
produce an [IllegalStateException].\n */\n@SinceKotlin(\"1.3\")\n@Suppress(\"UNCHECKED_CAST\")\npublic fun <R, T> (suspend R.() -> T).createCoroutine(\n receiver: R,\n completion: Continuation<T>\n): Continuation<Unit> =\n SafeContinuation(createCoroutineUnintercepted(receiver, completion).intercepted(), COROUTINE_SUSPENDED)\n\n/**\n * Starts a coroutine without a receiver and with result type [T].\n * This function creates and starts a new, fresh instance of suspendable computation every time it is invoked.\n * The [completion] continuation is invoked when the coroutine completes with a result or an exception.\n */\n@SinceKotlin(\"1.3\")\n@Suppress(\"UNCHECKED_CAST\")\npublic fun <T> (suspend () -> T).startCoroutine(\n completion: Continuation<T>\n) {\n createCoroutineUnintercepted(completion).intercepted().resume(Unit)\n}\n\n/**\n * Starts a coroutine with receiver type [R] and result type [T].\n * This function creates and starts a new, fresh instance of suspendable
computation every time it is invoked.\n * The [completion] continuation is invoked when the coroutine completes with a result or an exception.\n */\n@SinceKotlin(\"1.3\")\n@Suppress(\"UNCHECKED_CAST\")\npublic fun <R, T> (suspend R.() -> T).startCoroutine(\n receiver: R,\n completion: Continuation<T>\n) {\n createCoroutineUnintercepted(receiver, completion).intercepted().resume(Unit)\n}\n\n/**\n * Obtains the current continuation instance inside suspend functions and suspends\n * the currently running coroutine.\n *\n * In this function both [Continuation.resume] and [Continuation.resumeWithException] can be used either synchronously in\n * the same stack-frame where the suspension function is run or asynchronously later in the same thread or\n * from a different thread of execution. Subsequent invocation of any resume function will produce an [IllegalStateException].\n */\n@SinceKotlin(\"1.3\")\n@InlineOnly\npublic suspend inline fun <T> suspendCoroutine(crossinline block: (Continuation<T>)
-> Unit): T {\n contract { callsInPlace(block, InvocationKind.EXACTLY_ONCE) }\n return suspendCoroutineUninterceptedOrReturn { c: Continuation<T> ->\n val safe = SafeContinuation(c.intercepted())\n block(safe)\n safe.getOrThrow()\n }\n}\n\n/**\n * Returns the context of the current coroutine.\n */\n@SinceKotlin(\"1.3\")\n@Suppress(\"WRONG_MODIFIER_TARGET\")\n@InlineOnly\npublic suspend inline val coroutineContext: CoroutineContext\n get() {\n throw NotImplementedError(\"Implemented as intrinsic\")\n }\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines.intrinsics\n\nimport kotlin.coroutines.*\nimport kotlin.internal.InlineOnly\n\n/**\n * Starts an unintercepted coroutine without a receiver and with result type [T] and executes it until its
first suspension.\n * Returns the result of the coroutine or throws its exception if it does not suspend or [COROUTINE_SUSPENDED] if it suspends.\n * In the latter case, the [completion] continuation is invoked when the coroutine completes with a result or an exception.\n *\n * The coroutine is started directly in the invoker's thread without going through the [ContinuationInterceptor] that might\n * be present in the completion's [CoroutineContext]. It is the invoker's responsibility to ensure that a proper invocation\n * context is established.\n *\n * This function is designed to be used from inside of [suspendCoroutineUninterceptedOrReturn] to resume the execution of the suspended\n * coroutine using a reference to the suspending function.\n */\n@SinceKotlin(\"1.3\")\n@InlineOnly\npublic actual inline fun <T> (suspend () -> T).startCoroutineUninterceptedOrReturn(\n completion: Continuation<T>\n): Any? = this.asDynamic()(completion, false)\n\n/**\n * Starts an unintercepted coroutine
with receiver type [R] and result type [T] and executes it until its first suspension.\n * Returns the result of the coroutine or throws its exception if it does not suspend or [COROUTINE_SUSPENDED] if it suspends.\n * In the latter case, the [completion] continuation is invoked when the coroutine completes with a result or an exception.\n *\n * The coroutine is started directly in the invoker's thread without going through the [ContinuationInterceptor] that might\n * be present in the completion's [CoroutineContext]. It is the invoker's responsibility to ensure that a

proper invocation\n * context is established.\n *\n * This function is designed to be used from inside of [suspendCoroutineUninterceptedOrReturn] to resume the execution of the suspended\n * coroutine using a reference to the suspending function.\n */\n@SinceKotlin(\"1.3\")\n@InlineOnly\npublic actual inline fun <R, T> (suspend R.() -> T).startCoroutineUninterceptedOrReturn(\n receiver: R,\n completion: Continuation<T>\n): Any? = this.asDynamic()(receiver, completion, false)\n\n@InlineOnly\ninternal actual inline fun <R, P, T> (suspend R.(P) -> T).startCoroutineUninterceptedOrReturn(\n receiver: R,\n param: P,\n completion: Continuation<T>\n): Any? = this.asDynamic()(receiver, param, completion, false)\n\n/**\n * Creates unintercepted coroutine without receiver and with result type [T].\n * This function creates a new, fresh instance of suspendable computation every time it is invoked.\n *\n * To start executing the created coroutine, invoke `resume(Unit)` on the returned [Continuation] instance.\n * The [completion] continuation is invoked when coroutine completes with result or exception.\n *\n * This function returns unintercepted continuation.\n * Invocation of `resume(Unit)` starts coroutine immediately in the invoker's call stack without going through the\n * [ContinuationInterceptor] that might be present in the completion's [CoroutineContext].\n * It is the invoker's responsibility to ensure that a proper invocation context is established.\n * Note that [completion] of this function may get invoked in an arbitrary context.\n *\n * [Continuation.intercepted] can be used to acquire the intercepted continuation.\n * Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of\n * both the coroutine and [completion] happens in the invocation context established by\n * [ContinuationInterceptor].\n *\n * Repeated invocation of any resume function on the resulting continuation corrupts the\n * state machine of the coroutine and may result in arbitrary behaviour or exception.\n */\n@SinceKotlin(\"1.3\")\npublic actual fun <T> (suspend () -> T).createCoroutineUnintercepted(\n completion: Continuation<T>\n): Continuation<Unit> =\n // Kotlin/JS suspend lambdas have an extra parameter `suspended`\n if (this.asDynamic().length == 2) {\n // When `suspended` is true the continuation is created, but not executed\n this.asDynamic()(completion, true)\n } else {\n createCoroutineFromSuspendFunction(completion) {\n this.asDynamic()(completion)\n }\n }\n\n/**\n * Creates unintercepted coroutine with receiver type [R] and result type [T].\n * This function creates a new, fresh instance of suspendable computation every time it is invoked.\n *\n * To start executing the created coroutine, invoke `resume(Unit)` on the returned [Continuation] instance.\n * The [completion] continuation is invoked when coroutine completes with result or exception.\n *\n * This function returns unintercepted continuation.\n * Invocation of `resume(Unit)` starts coroutine immediately in the invoker's call stack without going through the\n * [ContinuationInterceptor] that might be present in the completion's [CoroutineContext].\n * It is the invoker's responsibility to ensure that a proper invocation context is established.\n * Note that [completion] of this function may get invoked in an arbitrary context.\n *\n * [Continuation.intercepted] can be used to acquire the intercepted continuation.\n * Invocation of `resume(Unit)` on intercepted continuation guarantees that execution of\n * both the coroutine and [completion] happens in the invocation context established by\n * [ContinuationInterceptor].\n *\n * Repeated invocation of any resume function on the resulting continuation corrupts the\n * state machine of the coroutine and may result in arbitrary behaviour or exception.\n */\n@SinceKotlin(\"1.3\")\npublic actual fun <R, T> (suspend R.() -> T).createCoroutineUnintercepted(\n receiver: R,\n completion: Continuation<T>\n): Continuation<Unit> =\n // Kotlin/JS suspend lambdas have an extra parameter `suspended`\n if (this.asDynamic().length == 3) {\n // When `suspended` is true the continuation is created, but not executed\n this.asDynamic()(receiver, completion, true)\n } else {\n createCoroutineFromSuspendFunction(completion) {\n this.asDynamic()(receiver, completion)\n }\n }\n\n/**\n * Intercepts this continuation with [ContinuationInterceptor].\n *\n * This function shall be used on the immediate result of [createCoroutineUnintercepted] or [suspendCoroutineUninterceptedOrReturn],\n * in which case it checks for [ContinuationInterceptor] in the continuation's [context][Continuation.context],\n * invokes [ContinuationInterceptor.interceptContinuation], caches and returns the result.\n *\n * If this function is invoked on other [Continuation] instances it returns `this` continuation unchanged.\n */\n@SinceKotlin(\"1.3\")\npublic actual fun <T> Continuation<T>.intercepted(): Continuation<T> =\n (this as? CoroutineImpl)?.intercepted() ?: this\n\n\nprivate inline fun <T> createCoroutineFromSuspendFunction(\n completion: Continuation<T>,\n

crossinline block: () -> Any?\n): Continuation<Unit> {\n   @Suppress(\"UNCHECKED_CAST\")\n   return object : CoroutineImpl(completion as Continuation<Any?>) {\n
        override fun doResume(): Any? {\n   exception?.let { throw it }\n   return block()\n   }\n   }\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n// Mirrors signature from JS IR BE\n// Used for js.translator/testData/box/number/mulInt32.kt\n@library\n@JsName(\"imulEmulated\")\n@Suppress(\"UNUSED_PARAMETER\")\ninternal fun imul(x: Int, y: Int): Int = definedExternally\n\n@Suppress(\"NOTHING_TO_INLINE\")\ninternal inline fun isArrayish(o: dynamic) = js(\"Kotlin\").isArrayish(o)\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\n// NOTE: Do not author your exceptions as they are written in this file, instead use this template:\n/*\npublic open class MyException : Exception {\n   constructor() : super()\n   constructor(message: String?) : super(message)\n   constructor(message: String?, cause: Throwable?) : super(message, cause)\n   constructor(cause: Throwable?) : super(cause)\n}\n*/\n\n\n// TODO: remove primary constructors, make all secondary KT-22055\n\n@Suppress(\"USELESS_ELVIS_RIGHT_IS_NULL\")\npublic actual open class Error actual constructor(message: String?, cause: Throwable?) : Throwable(message, cause ?: null) {\n   actual constructor() : this(null, null)\n   actual constructor(message: String?) : this(message, null)\n   actual constructor(cause: Throwable?) : this(undefined, cause)\n}\n\n@Suppress(\"USELESS_ELVIS_RIGHT_IS_NULL\")\npublic actual open class Exception actual constructor(message: String?, cause: Throwable?) : Throwable(message, cause ?: null) {\n   actual constructor() : this(null, null)\n   actual constructor(message: String?) : this(message, null)\n   actual constructor(cause: Throwable?) : this(undefined, cause)\n}\n\npublic actual open class RuntimeException actual constructor(message: String?, cause: Throwable?) : Exception(message, cause) {\n   actual constructor() : this(null, null)\n   actual constructor(message: String?) : this(message, null)\n   actual constructor(cause: Throwable?) : this(undefined, cause)\n}\n\npublic actual open class IllegalArgumentException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {\n   actual constructor() : this(null, null)\n   actual constructor(message: String?) : this(message, null)\n   actual constructor(cause: Throwable?) : this(undefined, cause)\n}\n\npublic actual open class IllegalStateException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {\n   actual constructor() : this(null, null)\n   actual constructor(message: String?) : this(message, null)\n   actual constructor(cause: Throwable?) : this(undefined, cause)\n}\n\npublic actual open class IndexOutOfBoundsException actual constructor(message: String?) : RuntimeException(message) {\n   actual constructor() : this(null)\n}\n\npublic actual open class ConcurrentModificationException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {\n   actual constructor() : this(null, null)\n   actual constructor(message: String?) : this(message, null)\n   actual constructor(cause: Throwable?) : this(undefined, cause)\n}\n\npublic actual open class UnsupportedOperationException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {\n   actual constructor() : this(null, null)\n   actual constructor(message: String?) : this(message, null)\n   actual constructor(cause: Throwable?) : this(undefined, cause)\n}\n\n\npublic actual open class NumberFormatException actual constructor(message: String?) : IllegalArgumentException(message) {\n   actual constructor() : this(null)\n}\n\n\npublic actual open class NullPointerException actual constructor(message: String?) : RuntimeException(message) {\n   actual constructor() : this(null)\n}\n\npublic actual open class ClassCastException actual constructor(message: String?) : RuntimeException(message) {\n   actual constructor() : this(null)\n}\n\npublic actual open class AssertionError\n@SinceKotlin(\"1.4\")\nconstructor(message: String?, cause: Throwable?) : Error(message, cause) {\n   actual constructor() : this(null)\n   constructor(message: String?) : this(message, null)\n   actual constructor(message: Any?) : this(message.toString(), message as? Throwable)\n}\n\npublic actual open class

NoSuchElementException actual constructor(message: String?) : RuntimeException(message) {\n   actual constructor() : this(null)\n}\n\n@SinceKotlin(\"1.3\")\npublic actual open class ArithmeticException actual constructor(message: String?) : RuntimeException(message) {\n   actual constructor() : this(null)\n}\n\npublic actual open class NoWhenBranchMatchedException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {\n   actual constructor() : this(null, null)\n   actual constructor(message: String?) : this(message, null)\n   actual constructor(cause: Throwable?) : this(undefined, cause)\n}\n\npublic actual open class UninitializedPropertyAccessException actual constructor(message: String?, cause: Throwable?) : RuntimeException(message, cause) {\n   actual constructor() : this(null, null)\n   actual constructor(message: String?) : this(message, null)\n   actual constructor(cause: Throwable?) : this(undefined, cause)\n}\n","/*\n * Copyright 2010-2019 JetBrains s.r.o. Use of this source code is governed by the Apache 2.0 license\n * that can be found in the license/LICENSE.txt file.\n */\n\n\n@file:Suppress(\"UNUSED_PARAMETER\")\n\npackage kotlin.js\n\n@kotlin.internal.InlineOnly\ninternal inline fun jsDeleteProperty(obj: Any, property: Any) {\n   js(\"delete obj[property]\")\n}\n\n@kotlin.internal.InlineOnly\ninternal inline fun jsBitwiseOr(lhs: Any?, rhs: Any?): Int =\n   js(\"lhs | rhs\").unsafeCast<Int>()","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin.math\n\n/**\n * Returns this value with the sign bit same as of the [sign] value.\n *\n * If [sign] is `NaN` the sign of the result is undefined.\n */\n@SinceKotlin(\"1.2\")\npublic actual fun Double.withSign(sign: Double): Double {\n   val thisSignBit = js(\"Kotlin\").doubleSignBit(this).unsafeCast<Int>()\n   val newSignBit = js(\"Kotlin\").doubleSignBit(sign).unsafeCast<Int>()\n   return if (thisSignBit == newSignBit) this else -this\n}","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\n\n/**\n * Returns a bit representation of the specified floating-point value as [Long]\n * according to the IEEE 754 floating-point \"double format\" bit layout.\n */\n@SinceKotlin(\"1.2\")\n@library(\"doubleToBits\")\npublic actual fun Double.toBits(): Long = definedExternally\n\n/**\n * Returns a bit representation of the specified floating-point value as [Long]\n * according to the IEEE 754 floating-point \"double format\" bit layout,\n * preserving `NaN` values exact layout.\n */\n@SinceKotlin(\"1.2\")\n@library(\"doubleToRawBits\")\npublic actual fun Double.toRawBits(): Long = definedExternally\n\n/**\n * Returns the [Double] value corresponding to a given bit representation.\n */\n@SinceKotlin(\"1.2\")\n@kotlin.internal.InlineOnly\npublic actual inline fun Double.Companion.fromBits(bits: Long): Double = js(\"Kotlin\").doubleFromBits(bits).unsafeCast<Double>()\n\n/**\n * Returns a bit representation of the specified floating-point value as [Int]\n * according to the IEEE 754 floating-point \"single format\" bit layout.\n *\n * Note that in Kotlin/JS [Float] range is wider than \"single format\" bit layout can represent,\n * so some [Float] values may overflow, underflow or loose their accuracy after conversion to bits and back.\n */\n@SinceKotlin(\"1.2\")\n@library(\"floatToBits\")\npublic actual fun Float.toBits(): Int = definedExternally\n\n/**\n * Returns a bit representation of the specified floating-point value as [Int]\n * according to the IEEE 754 floating-point \"single format\" bit layout,\n * preserving `NaN` values exact layout.\n *\n * Note that in Kotlin/JS [Float] range is wider than \"single format\" bit layout can represent,\n * so some [Float] values may overflow, underflow or loose their accuracy after conversion to bits and back.\n */\n@SinceKotlin(\"1.2\")\n@library(\"floatToRawBits\")\npublic actual fun Float.toRawBits(): Int = definedExternally\n\n/**\n * Returns the [Float] value corresponding to a given bit representation.\n */\n@SinceKotlin(\"1.2\")\n@kotlin.internal.InlineOnly\npublic actual inline fun Float.Companion.fromBits(bits: Int): Float = js(\"Kotlin\").floatFromBits(bits).unsafeCast<Float>()\n\n\n@Suppress(\"NOTHING_TO_INLINE\")\ninternal inline fun Long(low: Int, high: Int) = js(\"Kotlin\").Long.fromBits(low, high).unsafeCast<Long>()\ninternal inline

val Long.low: Int get() = this.asDynamic().getLowBits().unsafeCast<Int>()\ninternal inline val Long.high: Int get() = this.asDynamic().getHighBits().unsafeCast<Int>()\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\nimport kotlin.reflect.KClass\n\n@PublishedApi\ninternal fun <T : Annotation> KClass<*>.findAssociatedObject(@Suppress(\"UNUSED_PARAMETER\") annotationClass: KClass<T>): Any?
{\n // This API is not supported in js-v1. Return `null` to be source-compatible with js-ir.\n return null\n}\n","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/**\n * Returns a string representation of this [Long] value in the specified [radix].\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.\n */\n@SinceKotlin(\"1.2\")\npublic actual fun Long.toString(radix: Int): String = asDynamic().toString(checkRadix(radix))","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n@PublishedApi\n@Suppress(\"NOTHING_TO_INLINE\")\n@JsPolyfill(\"\"\"\nif (typeof Array.prototype.fill
=== \"undefined\") {\n // Polyfill from https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/fill#Polyfill\n Object.defineProperty(Array.prototype, 'fill', {\n value: function (value) {\n // Steps 1-2.\n if (this == null) {\n throw new TypeError('this is null or not defined');\n }\n\n var O = Object(this);\n\n // Steps 3-5.\n var len = O.length >>> 0;\n\n // Steps 6-7.\n var start = arguments[1];\n var relativeStart = start >> 0;\n\n // Step 8.\n var k = relativeStart < 0 ?\n Math.max(len + relativeStart, 0) :\n Math.min(relativeStart, len);\n\n // Steps 9-10.\n var end = arguments[2];\n var relativeEnd = end === undefined ?\n len : end >> 0;\n\n // Step 11.\n var finalValue = relativeEnd < 0 ?\n Math.max(len + relativeEnd, 0) :\n Math.min(relativeEnd, len);\n\n // Step 12.\n while (k < finalValue) {\n O[k] = value;\n k++;\n }\n\n // Step 13.\n return O;\n }\n });\n}\n\n[Int8Array, Int16Array, Uint16Array, Int32Array, Float32Array, Float64Array].forEach(function (TypedArray) {\n if (typeof TypedArray.prototype.fill === \"undefined\") {\n Object.defineProperty(TypedArray.prototype, 'fill', {\n value: Array.prototype.fill\n });\n }\n})\n\"\"\")\ninternal inline fun Any.nativeFill(element: Any?, fromIndex: Int, toIndex: Int): Unit {\n asDynamic().fill(element, fromIndex, toIndex)\n}\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt
file.\n */\n\npackage kotlin.js\n\n@PublishedApi\n@Suppress(\"NOTHING_TO_INLINE\")\n@JsPolyfill(\"\"\"\n[Int8Array, Int16Array, Uint16Array, Int32Array, Float32Array, Float64Array].forEach(function (TypedArray) {\n if (typeof TypedArray.prototype.sort === \"undefined\") {\n Object.defineProperty(TypedArray.prototype, 'sort', {\n value: function(compareFunction) {\n compareFunction = compareFunction || function (a, b) {\n if (a < b) return -1;\n if (a > b) return 1;\n if (a === b) {\n if (a !== 0) return 0;\n var ia = 1 / a;\n return ia === 1 / b ? 0 : (ia < 0 ? -1 : 1);\n }\n return a !== a ? (b !== b ? 0 : 1) : -1\n }\n return Array.prototype.sort.call(this, compareFunction || totalOrderComparator);\n }\n });\n }\n})\n\"\"\")\ninternal
inline fun Any.nativeSort(noinline comparison: (a: dynamic, b: dynamic) -> Int = js(\"undefined\")): Unit {\n asDynamic().sort(comparison)\n}","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\n// 1343 ranges

totally\nprivate object Category {\n val decodedRangeStart: IntArray\n val decodedRangeCategory: IntArray\n \n init {\n val toBase64 =
\"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/\"\n val fromBase64 = IntArray(128)\n for (i in toBase64.indices) {\n fromBase64[toBase64[i].code] = i\n }\n \n // rangeStartDiff.length = 1482\n val rangeStartDiff
=
\"gBCFEDCKCDCaDDaDBhBCEEDDDDDEDXBHYBH5BRwBGDCHDCIDFHDCHFDCDEIRTEE7BGHDDJlCBbSEMOFGERwDEDDDDECEFCRBJhBFDCYFFCCzBvBjBBFC3BOhDBmBDGpBDDCtBBJIbEECLGDFCLDCgBBKVKEDiDDHCFECECKCEODBebC5CLBOKhBJDDDDWEBHFCFCPBZDEL1BVBSLPBgBB2BDBDICFBHKCCKCPDBHEDWBHEDDDDEDEDIBDGDCKCCGDDDCGECCWBFMDDCDEDDCHDDHKDDBKDBHFCWBFGFDBDDFEDBPDDKCHBGDCHEDWBFGFDCEDEDBHDDGDCKCGJEGDBFDDFDDDDDMEFDBFDCGBOKDFDFDCGFCXBQDDDDDBEGEDFDDKHBHDDGFCXBKBFCEFCFCHCHECCKDNCCHFCoBEDECFDDDDHDCCKJBGDCSDYBJEHBFDDEBIGKDCMuBFHEBGBIBKCkBFBFBXEIFJDFDGCKCEgBBDPEDGKKGECIBkBEOBDFFLBkBBIBEFFEClBrBCEBEGDBKGGDDDDDCHDENDCFEKDDlBDDFrBCDpKBECGEECpBBEChBBECGEECPB5BBECjCCDJUDQKG2CCGDsTCRBaCDrCDDIHNBEDLSDCJSCMLFCCM0BDHGFLBFDDKGKGEFDDBKGjBB1BHFChBDFmCKfDDDDDDCGDCFDKeCFLsBEaGKBDiBXDDD1BDGDEIGJEKGKGHBGCMF/BEBvBCEDDFHEKHKJJDDeDDGDKsBFEDCIEkBIICCDFKDDKeGCJHrBCDIIDBNBHEBEFDBFsB/BNBiBlB6BBF1EIiDJIGCGCIIIIGCGCIIIIOCIIIIIIDFEDDBFEDDDDEBDIFDDFEDBLFGCEEICFBJCDEDCLDKBFBKCCGDDKDDNDgBQNEBDMPFFDEDEBFFHECEBEEDFBEDDQjBCEDEFFCCJHBeEEfsIIEUCHCxCBeZoBGlCZLV8BuCW3FBJB2BIvDB4HOesBFCfKQgIjEW/BEgBCiIwBVCGnBCgBBpDvBBuBEDBHEFGCCjDCGEDCFCFlBDDF4BHCOBXJHBHBHBHBHBHBHBHBgBCECGHGEDIFBKCEDMEtBaB5CM2GaMEDDCKCGFCJEDFDDDC2CDDDB6CDCFrBB+CDEKgBkBMQfBKeIBPgBKnBPgKguGgC9vUDVB3jBD3BJoBGCsIBDQKCUuBDDKCcCCmCKCGIXJCNC/BBHGKDECEVFBEMCEEBqBDDGDFDXDCEBDGEG0BEICyBQCICKGSGDEBKcICXLCLBdDDBvBDECCDNCKECFCJKFBpBFEDCJDBICCKCEQBGDDByBEDCEFBYDCLEDDCKGCGCGJHBHBrBBEJDEwCjBIDCKGk9KMXExBEggCgoGuLCqDmBHMFFCKBNBFBIsDQRrLCQgCC2BoBMCCQGEGQDCQDDDDFDGDECEEFBnEEBFEDCKCDCaDDaDBFCKBtBCfDGCGCFEDDDDCECKDC\"\n
val diff = decodeVarLenBase64(rangeStartDiff, fromBase64, 1342)\n val start = IntArray(diff.size + 1)\n for (i in diff.indices) {\n start[i + 1] = start[i] + diff[i]\n }\n decodedRangeStart = start\n \n // rangeCategory.length = 2033\n val rangeCategory =
\"PsY44a41W54UYJYZYB14W7XC15WZPsYa84bl9Zw8b85Lr7C44brlerrYBZBCZCiBiBiBhCiiBhChiBhiCBhhChiCihBhChCChiBhChiClBCFhjCiBiBihDhiBhCCihBiBBhCCFCEbEbEb7EbGhCk7BixRkiCi4BRbh4BhRhCBRBCiiBBCiBChiZBCBCiBcGHhChCiBRBxxEYC40Rx8c6RGUm4GRFRFYRQZ44acG4wRYFEFGJYllGFlYGwcGmkEmcGFJFl8cYxwFGFGRFGFRJFGkkcYkxRm6aFGEGmmEmEGRYRFGxxYFRFRFRGQGIFmIFIGIooGFGFGYJ4EFmoIRFlxRlxRFRFxlRxlFllRxmFIGxxIoxRomFRIRxlFlmGRJFaL86F4mRxmGoRFRFRFRFllRxGIGRxmGxmGmxRxGRFlRRJmmFllGYRmmIRFllRlRFRFllRFxxGFIGmmRoxImxRFRllGmxRJ4aRFGxmIoRFlxRlxRFRFllRFxxGlImoGmmRxoIxoIGRmmIRxlFlmGRJ8FLRxmFFRFllRllRxxFlRlxRxlFRFRFRooGRIooRomRxFRIRJLc8aRmoIoGFllRlRFRFRlmGmoIooRGRGRxmGFRllGmxRJRYL8lGooYFllRlRFRFRFRmlIIxGooRGRIRlxFGRJxlFRGIFllRlRFlmGIGxIooRomF8xRxxFlllLFGRJLcFxmIoRFRFRFxlRFRxxGxxIooGmmRRIRJxxIoYRFllGGRaFEGYJYRxlFRFRFlRFllGGlxRFxEGRJRFRFcY84c8mGcJL8G1WIFRFRGIGmmYFGRGRcGc88RYcYRFIGIGmmIomGFJYFooGmlFllGmmFIFIFGFmoIGIomFJIm8cBhRRxxBC4ECFRFRFlRFRFRFRFRFRFlRFRFRFRFRFRFRGYLRFcRBRCxxUF8YFMF1WRFYKFRFRFGRFGYRFGRFllRlRGRFmmIGIooGGY44E46FmxRJRLRY44U44GmmQRJRFEFRFGFlGRFRFxmGmoIooGmoIoxRxxIoGIGRxxcx4YJFRFRFRFRJLRcFmmIomRx4YFoGGmRomIGIGmxRJRJRYEYRGmmHRGIFmIGmIIooGFRJYcGcRmmIFomGmmIomGmlFJFmoGooGGIRYFIGIGRYJRFJFEYCRBRBYRGYGIGFGFllGomGFRCECECEGRGhCCiBCBCRBRCBCBCRBRCxBCBCRCDCDCDCiiRBj7CbCiiRBj7b7iCiiRxiCBRbCBbxxCiiRBj7bRMQUY9+V9+VYtOQMY9eY43X44Z1WY54XYMQRQrERLZ12ELZ12RERaRGHGHGR88B88BihBhiChhC8hcZBc8BB8CBCFi8cihBZBC8Z8CLKhCKr8cRZcZc88ZcZc85Z8ZcZc1WcZc1WcZcZcZcRcRLcLcZcZcZcZc1WLcZ1WZ1WZcZ1WZ1WZ1WZcZcZcRcRcBRCixBBCiBBihC

CEBhCCchCGhCRY44LCiRRxxCFRkYRGFRFRFRFRFRFRFRFRFRGY9eY49eY44U49e49e1WYEYUY04VY48cRcRcRcRcRs4Y48ElK1Wc1W12U2cKGooUE88KqqEl4c8RFxxGm7bkkFUF4kEkFRFRFx8cLcFcRFcRLcLcLcLcLcFcFRFEFRcRFEYFEYFJFRhClmHnnYG4EhCEGFKGYRbEbhCCiBECiBhCk7bhClBihCiBBCBhCRhiBhhCCRhiFkkCFlGllGllGFooGmIcGRL88aRFYRIFIGRYJRGFYl4FGJFGYFGIRYFRGIFmoIGIGIYxEJRYFmEFJFRFGmoImoIGRFGFmIRJRYFEFcloGIFmlGmlFGFlmGFRllEYFomGo4YlkEoGRFRFRFRFRFRCbECk7bRCFooG4oGRJRFRFRFRTSFRFRCRCRlGFZFRFRlxFFbRF2VRFRFRF6cRGY41WRG40UX1W44V24Y44X33Y44R44U1WY50Z5R46YRFRFxxQY44a41W54UYJYZYB14W7XC15WZ12YYFEFEFRFRFRFlxRllRxxa65b86axcZcRQcR\"\n
decodedRangeCategory = decodeVarLenBase64(rangeCategory, fromBase64, 1343)\n }\n}\n\nprivate fun categoryValueFrom(code: Int, ch: Int):
Int {\n return when {\n code < 0x20 -> code\n code < 0x400 -> if ((ch and 1) == 1) code shr 5 else code and 0x1f\n else ->\n when (ch % 3) {\n 2 -> code shr 10\n 1 -> (code shr 5) and 0x1f\n else -> code and 0x1f\n }\n }\n}\n\n/**\n * Returns the Unicode general category of this character as an Int.\n */\ninternal fun Char.getCategoryValue(): Int {\n val ch = this.code\n\n val index = binarySearchRange(Category.decodedRangeStart, ch)\n val start = Category.decodedRangeStart[index]\n val code = Category.decodedRangeCategory[index]\n val value = categoryValueFrom(code, ch - start)\n\n return if (value == 17) CharCategory.UNASSIGNED.value else value\n}\n\ninternal fun decodeVarLenBase64(base64: String, fromBase64: IntArray, resultLength: Int): IntArray {\n val result = IntArray(resultLength)\n var index = 0\n var int = 0\n var shift = 0\n for (char in base64) {\n
val sixBit = fromBase64[char.code]\n int = int or ((sixBit and 0x1f) shl shift)\n if (sixBit < 0x20) {\n result[index++] = int\n int = 0\n shift = 0\n } else {\n shift += 5\n }\n }\n return result\n}\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.js.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Reverses elements in the list in-place.\n */\npublic actual fun <T> MutableList<T>.reverse(): Unit {\n val midPoint = (size / 2) - 1\n if (midPoint < 0) return\n var reverseIndex = lastIndex\n for (index in 0..midPoint)
{\n val tmp = this[index]\n this[index] = this[reverseIndex]\n this[reverseIndex] = tmp\n reverseIndex--\n }\n}\n\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\n// 37 ranges totally\nprivate object Digit {\n internal val rangeStart = intArrayOf(\n 0x0030, 0x0660, 0x06f0, 0x07c0, 0x0966, 0x09e6, 0x0a66, 0x0ae6, 0x0b66, 0x0be6, 0x0c66, 0x0ce6, 0x0d66, 0x0de6, 0x0e50, 0x0ed0, 0x0f20, 0x1040, 0x1090, 0x17e0, \n 0x1810, 0x1946, 0x19d0, 0x1a80, 0x1a90, 0x1b50, 0x1bb0, 0x1c40, 0x1c50, 0xa620, 0xa8d0, 0xa900, 0xa9d0, 0xa9f0, 0xaa50, 0xabf0, 0xff10, \n)\n}\n\n/**\n * Returns the index of the largest element in [array] smaller or equal to the specified [needle],\n * or -1 if [needle] is smaller than the smallest element in [array].\n */\ninternal fun binarySearchRange(array: IntArray, needle: Int): Int {\n var bottom = 0\n var top = array.size - 1\n var middle = -1\n var value = 0\n while (bottom <= top) {\n middle = (bottom + top) / 2\n value = array[middle]\n if (needle > value)\n bottom = middle + 1\n else if (needle == value)\n return middle\n else\n top = middle - 1\n }\n return middle - (if (needle < value) 1 else 0)\n}\n\n/**\n * Returns an integer from 0..9 indicating the digit this character represents,\n * or -1 if this character is not a digit.\n */\ninternal fun Char.digitToIntImpl(): Int {\n val ch = this.code\n val index = binarySearchRange(Digit.rangeStart, ch)\n val diff = ch - Digit.rangeStart[index]\n return if (diff < 10) diff else -1\n}\n\n/**\n * Returns
`true` if this character is a digit.\n */\ninternal fun Char.isDigitImpl(): Boolean {\n return digitToIntImpl() >= 0\n}\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of

this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\n// 222 ranges totally\nprivate object Letter {\n val decodedRangeStart: IntArray\n val decodedRangeLength: IntArray\n val decodedRangeCategory: IntArray\n \n init {\n val toBase64 = \"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/\"\n val fromBase64 = IntArray(128)\n for (i in toBase64.indices) {\n fromBase64[toBase64[i].code] = i\n }\n \n // rangeStartDiff.length = 356\n val rangeStartDiff = \"hCgBpCQGYHZH5BRpBPPPPPPRMP5BPPlCPP6BkEPPPPcPXPzBvBrB3BOiDoBHwD+E3DauCnFmBmB2D6E1BlBTiBmBlBP5BhBiBrBvBjBqBnBPRtBiCmCtBlB0BmB5BiB7BmBgEmChBZgCoEoGVpBSfRhBPqKQ2BwBYoFgB4CJuTiEvBuCuDrF5DgEgFlJ1DgFmBQtBsBRGsB+BPiBlD1EIjDPRPPPQPPPPPGQSQS/DxENVNU+B9zCwBwBPPCkDPNnBPqDYY1R8B7FkFgTgwGgwUwmBgKwBuBScmEP/BPPPPPPrBP8B7F1B/ErBqC6B7BiBmBfQsBUwCw/KwqIwLwETPcPjQgJxFgBlBsD\"\n val diff = decodeVarLenBase64(rangeStartDiff, fromBase64, 222)\n val start = IntArray(diff.size)\n for (i in diff.indices) {\n if (i == 0) start[i] = diff[i]\n else start[i] = start[i - 1] + diff[i]\n }\n decodedRangeStart = start\n \n // rangeLength.length = 328\n val rangeLength = \"aaMBXHYH5BRpBPPPPPPRMP5BPPlCPPzBDOOPPcPXPzBvBjB3BOhDmBBpB7DoDYxB+EiBP1DoExBkBQhBekBPmBgBhBctBiBMWOOXhCsBpBkBUV3Ba4BkB0DlCgBXgBtD4FSdBfPhBPpKP0BvBXjEQ2CGsT8DhBtCqDpFvD1D3E0IrD2EkBJrBDOBsB+BPiBlB1EIjDPPPPPPPPPPPGPPMNLsBNPNPKCvBvBPPCkDPBmBPhDXXgD4B6FzEgDguG9vUtkB9JcuBSckEP/BPPPPPPBPf4FrBjEhBpC3B5BKaWPrBOwCk/KsCuLqDHPbPxPsFtEaaqDL\"\n decodedRangeLength = decodeVarLenBase64(rangeLength, fromBase64, 222)\n \n // rangeCategory.length = 959\n val rangeCategory = \"GFjgggUHGGFFZZZmzpz5qB6s6020B60ptltB6smt2sB60mz22B1+vv+8BZZ5s2850BW5q1ymtB506smzBF3q1q1qB1q1q1+Bgii4wDTm74g3KiggxqM60q1q1Bq1o1q1BF1qlrqrBZ2q5wprBGFZWWZGHFsjiooLowgmOowjkwCkgoiIk7ligGogiioBkwkiYkzj2oNoi+sbkwj04DghhkQ8wgiYkgoioDsgnkwC4gikQ//v+85BkwvoIsgoyI4yguI0whiwEowri4CoghsJowgqYowgm4DkwgsY/nwnzPowhmYkg6wI8yggZswikwHgxgmIoxgqYkwgk4DkxgmIkgoioBsgssoBgzgyI8g9gL8g9kI0wgwJoxgkoC0wgioFkw/wI0w53iF4gioYowjmgBHGq1qkgwBF1q1q8qBHwghuIwghyKk0goQkwgoQk3goQHGFHkyg0pBgxj6IoinkxDswno7Ikwhz9Bo0gioB8z48Rwli0xN0mpjoX8w78pDwltoqKHFGGwwgsIHFH3q1q16BFHWFZ1q10q1B2qlwq1B1q10q1B2q1yq1B6q1gq1Biq1qhxBir1qp1Bqt1q1qB1g1q1+B//3q16B///q1qBH/qlqq9Bholqq9B1i00a1q10qD1op1HkwmigEigiy6Cptogq1Bixo1kDq7/j00B2qgoBWGFm1lz50B6s5q1+BGWhggzhwBFFhgk4//Bo2jigE8wguI8wguI8wgugUog1qoB4qjmIwwi2KgkYHHH4lBgiFWkgIWoghssMmz5smrBZ3q1y50B5sm7gzBtz1smzB5smz50BqzqtmzB5sgzqzBF2/9//5BowgoIwmnkzPkwgk4C8ys65BkgoqI0wgy6FghquZo2giY0ghiIsgh24B4ghsQ8QF/v1q1OFs0O8iCHHF1qggz/B8wg6Iznv+//B08QgohsjK0QGFk7hsQ4gB\"\n decodedRangeCategory = decodeVarLenBase64(rangeCategory, fromBase64, 222)\n }\n}\n\n/**\n * Returns `true` if this character is a letter.\n */\ninternal fun Char.isLetterImpl(): Boolean {\n return getLetterType() != 0\n}\n\n/**\n * Returns `true` if this character is a lower case letter, or it has contributory property `Other_Lowercase`.\n */\ninternal fun Char.isLowerCaseImpl(): Boolean {\n return getLetterType() == 1 || code.isOtherLowercase()\n}\n\n/**\n * Returns `true` if this character is an upper case letter, or it has contributory property `Other_Uppercase`.\n */\ninternal fun Char.isUpperCaseImpl(): Boolean {\n return getLetterType() == 2 || code.isOtherUppercase()\n}\n\n/**\n * Returns\n * - `1` if the character is a lower case letter,\n * - `2` if the character is an
upper case letter,\n * - `3` if the character is a letter but not a lower or upper case letter,\n * - `0` otherwise.\n */\nprivate fun Char.getLetterType(): Int {\n val ch = this.code\n val index = binarySearchRange(Letter.decodedRangeStart, ch)\n\n val rangeStart = Letter.decodedRangeStart[index]\n val rangeEnd = rangeStart + Letter.decodedRangeLength[index] - 1\n val code = Letter.decodedRangeCategory[index]\n\n if (ch > rangeEnd) {\n return 0\n }\n\n val lastTwoBits = code

and 0x3\n\n   if (lastTwoBits == 0) { // gap pattern\n   var shift = 2\n   var threshold = rangeStart\n   for (i in 0..1) {\n   threshold += (code shr shift) and 0x7f\n   if (threshold > ch) {\n   return 3\n   }\n   shift += 7\n   threshold += (code shr shift) and 0x7f\n   if (threshold > ch) {\n   return 0\n   }\n   shift += 7\n   }\n   return 3\n   }\n\n   if (code <= 0x7) {\n   return lastTwoBits\n   }\n\n   val distance = (ch - rangeStart)\n   val shift = if (code <= 0x1F) distance % 2 else distance\n   return (code shr (2 * shift)) and 0x3\n}\n\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nprivate object OtherLowercase {\n   internal val otherLowerStart = intArrayOf(\n   0x00aa, 0x00ba, 0x02b0, 0x02c0, 0x02e0, 0x0345, 0x037a, 0x1d2c, 0x1d78, 0x1d9b, 0x2071, 0x207f, 0x2090, 0x2170, 0x24d0, 0x2c7c, 0xa69c, 0xa770, 0xa7f8, 0xab5c, \n   )\n   internal val otherLowerLength = intArrayOf(\n   1, 1, 9, 2, 5, 1, 1, 63, 1, 37, 1, 1, 13, 16, 26, 2, 2, 1, 2, 4, \n   )\n}\n\ninternal fun Int.isOtherLowercase(): Boolean {\n   val index = binarySearchRange(OtherLowercase.otherLowerStart, this)\n   return index >= 0 && this < OtherLowercase.otherLowerStart[index] + OtherLowercase.otherLowerLength[index]\n}\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\ninternal fun Int.isOtherUppercase(): Boolean {\n   return this in 0x2160..0x216f\n || this in 0x24b6..0x24cf\n}\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.js.*\n\n/**\n * Returns a character at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this char sequence.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\npublic actual fun CharSequence.elementAt(index: Int): Char {\n   return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, length: $length}\") }\n}\n\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\n// 4 ranges totally\ninternal fun Char.titlecaseCharImpl():
Char {\n   val code = this.code\n   // Letters repeating <Lu, Lt, Ll> sequence and code of the Lt is a multiple of 3, e.g. <\u01c4, \u01c5, \u01c6>\n   if (code in 0x01c4..0x01cc || code in 0x01f1..0x01f3) {\n   return (3 * ((code + 1) / 3)).toChar()\n   }\n   // Lower case letters whose title case mapping equivalent is equal to the original letter\n   if (code in 0x10d0..0x10fa || code in 0x10fd..0x10ff) {\n   return this\n   }\n   return uppercaseChar()\n}","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateStandardLib.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\nimport kotlin.js.*\nimport kotlin.ranges.contains\nimport kotlin.ranges.reversed\n\n/**\n * Returns an element
at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic actual fun UIntArray.elementAt(index: Int): UInt {\n   return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size}\")

```
}\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic actual fun ULongArray.elementAt(index: Int): ULong {\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size}\") }\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic actual fun UByteArray.elementAt(index: Int): UByte {\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size}\") }\n}\n\n/**\n * Returns an element at the given [index] or throws an [IndexOutOfBoundsException] if the [index] is out of bounds of this array.\n * \n * @sample samples.collections.Collections.Elements.elementAt\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic actual fun UShortArray.elementAt(index: Int): UShort {\n    return elementAtOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, size: $size}\") }\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic actual fun UIntArray.asList(): List<UInt> {\n    return object : AbstractList<UInt>(), RandomAccess {\n        override val size: Int get() = this@asList.size\n        override fun isEmpty(): Boolean = this@asList.isEmpty()\n        override fun contains(element: UInt): Boolean = this@asList.contains(element)\n        override fun get(index: Int): UInt {\n            AbstractList.checkElementIndex(index, size)\n            return this@asList[index]\n        }\n        override fun indexOf(element: UInt): Int {\n            @Suppress(\"USELESS_CAST\")\n            if ((element as Any?) !is UInt) return -1\n            return this@asList.indexOf(element)\n        }\n        override fun lastIndexOf(element: UInt): Int {\n            @Suppress(\"USELESS_CAST\")\n            if ((element as Any?) !is UInt) return -1\n            return this@asList.lastIndexOf(element)\n        }\n    }\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic actual fun ULongArray.asList(): List<ULong> {\n    return object : AbstractList<ULong>(), RandomAccess {\n        override val size: Int get() = this@asList.size\n        override fun isEmpty(): Boolean = this@asList.isEmpty()\n        override fun contains(element: ULong): Boolean = this@asList.contains(element)\n        override fun get(index: Int): ULong {\n            AbstractList.checkElementIndex(index, size)\n            return this@asList[index]\n        }\n        override fun indexOf(element: ULong): Int {\n            @Suppress(\"USELESS_CAST\")\n            if ((element as Any?) !is ULong) return -1\n            return this@asList.indexOf(element)\n        }\n        override fun lastIndexOf(element: ULong): Int {\n            @Suppress(\"USELESS_CAST\")\n            if ((element as Any?) !is ULong) return -1\n            return this@asList.lastIndexOf(element)\n        }\n    }\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic actual fun UByteArray.asList(): List<UByte> {\n    return object : AbstractList<UByte>(), RandomAccess {\n        override val size: Int get() = this@asList.size\n        override fun isEmpty(): Boolean = this@asList.isEmpty()\n        override fun contains(element: UByte): Boolean = this@asList.contains(element)\n        override fun get(index: Int): UByte {\n            AbstractList.checkElementIndex(index, size)\n            return this@asList[index]\n        }\n        override fun indexOf(element: UByte): Int {\n            @Suppress(\"USELESS_CAST\")\n            if ((element as Any?) !is UByte) return -1\n            return this@asList.indexOf(element)\n        }\n        override fun lastIndexOf(element: UByte): Int {\n            @Suppress(\"USELESS_CAST\")\n            if ((element as Any?) !is UByte) return -1\n            return this@asList.lastIndexOf(element)\n        }\n    }\n}\n\n/**\n * Returns a [List] that wraps the original array.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic actual fun UShortArray.asList(): List<UShort> {\n    return object : AbstractList<UShort>(), RandomAccess {\n        override val size: Int get() = this@asList.size\n        override fun isEmpty(): Boolean = this@asList.isEmpty()\n        override fun contains(element: UShort): Boolean = this@asList.contains(element)\n        override fun get(index: Int): UShort {\n            AbstractList.checkElementIndex(index, size)\n            return this@asList[index]\n        }\n        override fun indexOf(element: UShort): Int {\n            @Suppress(\"USELESS_CAST\")\n            if ((element as Any?) !is UShort) return -1\n            return
```

this@asList.indexOf(element)\n }\n override fun lastIndexOf(element: UShort): Int {\n @Suppress(\"USELESS_CAST\")\n if ((element as Any?) !is UShort) return -1\n return this@asList.lastIndexOf(element)\n }\n }\n}\n\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n//\n// NOTE: THIS FILE IS AUTO-GENERATED by the GenerateUnicodeData.kt\n// See: https://github.com/JetBrains/kotlin/tree/master/libraries/stdlib\n//\n\n// 9 ranges totally\n/**\n * Returns `true` if this character is a whitespace.\n */\ninternal fun Char.isWhitespaceImpl(): Boolean {\n val ch = this.code\n return ch in 0x0009..0x000d\n || ch in 0x001c..0x0020\n || ch == 0x00a0\n || ch > 0x1000 && (\n ch == 0x1680\n || ch in 0x2000..0x200a\n || ch == 0x2028\n || ch == 0x2029\n || ch == 0x202f\n || ch == 0x205f\n || ch == 0x3000\n)\n}\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\n\npublic
actual fun interface Comparator<T> {\n @JsName(\"compare\")\n public actual fun compare(a: T, b: T): Int\n}\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\nimport kotlin.annotation.AnnotationTarget.*\n\n@Target(FUNCTION)\n@Deprecated(\"Use inline extension function with body using dynamic\")\npublic annotation class nativeGetter\n\n@Target(FUNCTION)\n@Deprecated(\"Use inline extension function with body using dynamic\")\npublic annotation class nativeSetter\n\n@Target(FUNCTION)\n@Deprecated(\"Use inline extension function with body using dynamic\")\npublic annotation class nativeInvoke\n\n@Target(CLASS, FUNCTION, PROPERTY)\ninternal annotation class library(public val name: String = \"\")\n\n@Target(CLASS)\ninternal annotation class marker\n\n/**\n * Gives a declaration (a function, a property
or a class) specific name in JavaScript.\n *\n * This may be useful in the following cases:\n *\n * * There are two functions for which the compiler gives same name in JavaScript, you can\n * mark one with `@JsName(...)` to prevent the compiler from reporting error.\n * * You are writing a JavaScript library in Kotlin. The compiler produces mangled names\n * for functions with parameters, which is unnatural for usual JavaScript developer.\n * You can put `@JsName(...)` on functions you want to be available from JavaScript.\n * * For some reason you want to rename declaration, e.g. there's common term in JavaScript\n * for a concept provided by the declaration, which in uncommon in Kotlin.\n *\n * Example:\n *\n * ``` kotlin\n * class Person(val name: String) {\n * fun hello() {\n * println(\"Hello $name!\")\n * }\n *\n * @JsName(\"helloWithGreeting\")\n * fun hello(greeting: String) {\n * println(\"$greeting $name!\")\n * }\n
* }\n * ```\n *\n * @property name the name which compiler uses both for declaration itself and for all references to the declaration.\n * It's required to denote a valid JavaScript identifier.\n *\n */\n@Retention(AnnotationRetention.BINARY)\n@Target(CLASS, FUNCTION, PROPERTY, CONSTRUCTOR, PROPERTY_GETTER, PROPERTY_SETTER)\npublic actual annotation class JsName(actual val name: String)\n\n/**\n * Denotes an `external` declaration that must be imported from native JavaScript library.\n *\n * The compiler produces the code relevant for the target module system, for example, in case of CommonJS,\n * it will import the declaration via the `require(...)` function.\n *\n * The annotation can be used on top-level external declarations (classes, properties, functions) and files.\n * In case of file (which can't be `external`) the following rule applies: all the declarations in\n * the file must be `external`. By applying `@JsModule(...)` on a file you tell the compiler to import a
JavaScript object\n * that contain all the declarations from the file.\n *\n * Example:\n *\n * ``` kotlin\n * @JsModule(\"jquery\")\n * external abstract class JQuery() {\n * // some declarations here\n * }\n *\n * @JsModule(\"jquery\")\n * external fun JQuery(element: Element): JQuery\n * ```\n *\n * @property import name of a module to import declaration from.\n * It is not interpreted by the Kotlin compiler, it's passed as is

directly to the target module system.\n *\n * @see JsNonModule\n */\n@Retention(AnnotationRetention.BINARY)\n@Target(CLASS, PROPERTY, FUNCTION, FILE)\npublic annotation class JsModule(val import: String)\n\n/**\n * Denotes an `external` declaration that can be used without module system.\n *\n * By default, an `external` declaration is available regardless your target module system.\n * However, by applying [JsModule] annotation you can make a declaration unavailable to *plain* module system.\n * Some JavaScript libraries are distributed both as a standalone downloadable piece of JavaScript and as a module available\n * as an npm package.\n * To tell the Kotlin compiler to accept both cases, you can augment [JsModule] with the `@JsNonModule` annotation.\n *\n * For example:\n *\n * ``` kotlin\n * @JsModule(\"jquery\")\n * @JsNonModule\n * @JsName(\"$\")\n * external abstract class JQuery() {\n *   // some declarations here\n * }\n *\n * @JsModule(\"jquery\")\n * @JsNonModule\n * @JsName(\"$\")\n * external fun JQuery(element: Element): JQuery\n * ```\n *\n * @see JsModule\n */\n@Retention(AnnotationRetention.BINARY)\n@Target(CLASS, PROPERTY, FUNCTION, FILE)\npublic annotation class JsNonModule\n\n/**\n * Adds prefix to `external` declarations in a source file.\n *\n * JavaScript does not have concept of packages (namespaces). They are usually emulated by nested objects.\n * The compiler turns references to `external` declarations either to plain unprefixed names (in case of *plain* modules)\n * or to plain imports.\n * However, if a JavaScript library provides its declarations in packages, you won't be satisfied with this.\n * You can tell the compiler to generate additional prefix before references to `external` declarations using the `@JsQualifier(...)`\n * annotation.\n *\n * Note that a file marked with the `@JsQualifier(...)` annotation can't contain non-`external` declarations.\n *\n * Example:\n *\n * ```\n * @file:JsQualifier(\"my.jsPackageName\")\n * package some.kotlinPackage\n *\n * external fun foo(x: Int)\n *\n * external fun bar(): String\n * ```\n *\n * @property value the qualifier to add to the declarations in the generated code.\n *   It must be a sequence of valid JavaScript identifiers separated by the `.` character.\n *   Examples of valid qualifiers are: `foo`, `bar.Baz`, `_.$0.f`.\n *\n * @see JsModule\n */\n@Retention(AnnotationRetention.BINARY)\n@Target(AnnotationTarget.FILE)\npublic annotation class JsQualifier(val value: String)\n\n/**\n * Exports top-level declaration on JS platform.\n *\n * Compiled module exposes declarations that are marked with this annotation without name mangling.\n *\n * This annotation can be applied to either files or top-level declarations.\n *\n * It is currently prohibited to export the following kinds of declarations:\n *\n *   * `expect` declarations\n *   * inline functions with reified type parameters\n *   * suspend functions\n *   * secondary constructors without `@JsName`\n *   * extension properties\n *   * enum classes\n *   * annotation classes\n *\n * Signatures of exported declarations must only contain \"exportable\" types:\n *\n *   * `dynamic`, `Any`, `String`, `Boolean`, `Byte`, `Short`, `Int`, `Float`, `Double`\n *   * `BooleanArray`, `ByteArray`, `ShortArray`, `IntArray`, `FloatArray`, `DoubleArray`\n *   * `Array<exportable-type>`\n *   * Function types with exportable parameters and return types\n *   * `external` or `@JsExport` classes and interfaces\n *   * Nullable counterparts of types above\n *   * Unit return type. Must not be nullable\n *\n * This annotation is experimental, meaning that restrictions mentioned above are subject to change.\n */\n@ExperimentalJsExport\n@Retention(AnnotationRetention.BINARY)\n@Target(CLASS, PROPERTY, FUNCTION, FILE)\n@SinceKotlin(\"1.3\")\npublic actual annotation class JsExport\n\n/**\n * Forces a top-level property to be initialized eagerly, opposed to lazily on the first access to file and/or property.\n */\n@ExperimentalStdlibApi\n@Retention(AnnotationRetention.BINARY)\n@Target(AnnotationTarget.PROPERTY)\n@SinceKotlin(\"1.6\")\n@Deprecated(\"This annotation is a temporal migration assistance and may be removed in the future releases, please consider filing an issue about the case where it is needed\")\npublic annotation class EagerInitialization","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.jvm\n\n// these are used in common generated code in stdlib\n\n// TODO: find how to deprecate these ones\n\n@Target(AnnotationTarget.FIELD)\n@Retention(AnnotationRetention.SOURCE)\npublic actual annotation class Volatile\n\n@Target(AnnotationTarget.FUNCTION,

AnnotationTarget.PROPERTY_GETTER,
AnnotationTarget.PROPERTY_SETTER)\n@Retention(AnnotationRetention.SOURCE)\npublic actual annotation class Synchronized\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n/**\n * Provides a skeletal implementation of the [MutableCollection] interface.\n *\n * @param E the type of elements contained in the collection. The collection is invariant in its element type.\n */\npublic actual abstract class AbstractMutableCollection<E> protected actual constructor() : AbstractCollection<E>(),
MutableCollection<E> {\n\n actual abstract override fun add(element: E): Boolean\n\n actual override fun remove(element: E): Boolean {\n checkIsMutable()\n val iterator = iterator()\n while (iterator.hasNext()) {\n if (iterator.next() == element) {\n iterator.remove()\n return true\n }\n }\n return false\n }\n\n actual override fun addAll(elements: Collection<E>): Boolean {\n checkIsMutable()\n var modified = false\n for (element in elements) {\n if (add(element)) modified = true\n }\n return modified\n }\n\n actual override fun removeAll(elements: Collection<E>): Boolean {\n checkIsMutable()\n return (this as MutableIterable<E>).removeAll { it in elements }\n }\n\n actual override fun retainAll(elements: Collection<E>): Boolean {\n checkIsMutable()\n return (this as MutableIterable<E>).removeAll
{ it !in elements }\n }\n\n actual override fun clear(): Unit {\n checkIsMutable()\n val iterator = this.iterator()\n while (iterator.hasNext()) {\n iterator.next()\n iterator.remove()\n }\n }\n\n @Deprecated(\"Provided so that subclasses inherit this function\", level = DeprecationLevel.HIDDEN)\n @JsName(\"toJSON\")\n protected fun toJSON(): Any = this.toArray()\n\n\n /**\n * This method is called every time when a mutating method is called on this mutable collection.\n * Mutable collections that are built (frozen) must throw `UnsupportedOperationException`.\n */\n internal open fun checkIsMutable(): Unit { }\n}\n\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n/*\n * Based on GWT AbstractList\n * Copyright 2007 Google
Inc.\n*/\n\n\npackage kotlin.collections\n\n/**\n * Provides a skeletal implementation of the [MutableList] interface.\n *\n * @param E the type of elements contained in the list. The list is invariant in its element type.\n */\npublic actual abstract class AbstractMutableList<E> protected actual constructor() : AbstractMutableCollection<E>(), MutableList<E> {\n protected var modCount: Int = 0\n\n abstract override fun add(index: Int, element: E): Unit\n abstract override fun removeAt(index: Int): E\n abstract override fun set(index: Int, element: E): E\n\n /**\n * Adds the specified element to the end of this list.\n *\n * @return `true` because the list is always modified as the result of this operation.\n */\n actual override fun add(element: E): Boolean {\n checkIsMutable()\n add(size, element)\n return true\n }\n\n actual override fun addAll(index: Int, elements: Collection<E>): Boolean {\n AbstractList.checkPositionIndex(index,
size)\n\n checkIsMutable()\n var _index = index\n var changed = false\n for (e in elements) {\n add(_index++, e)\n changed = true\n }\n return changed\n }\n\n actual override fun clear() {\n checkIsMutable()\n removeRange(0, size)\n }\n\n actual override fun removeAll(elements: Collection<E>): Boolean {\n checkIsMutable()\n return removeAll { it in elements }\n }\n\n actual override fun retainAll(elements: Collection<E>): Boolean {\n checkIsMutable()\n return removeAll { it !in elements }\n }\n\n\n actual override fun iterator(): MutableIterator<E> = IteratorImpl()\n\n actual override fun contains(element: E): Boolean = indexOf(element) >= 0\n\n actual override fun indexOf(element: E): Int {\n for (index in 0..lastIndex) {\n if (get(index) == element) {\n return index\n }\n }\n return -1\n }\n\n actual override fun lastIndexOf(element: E): Int {\n for (index in lastIndex downTo 0) {\n if (get(index) == element) {\n return index\n }\n }\n return -1\n }\n\n actual override fun listIterator(): MutableListIterator<E> = listIterator(0)\n actual override fun listIterator(index: Int): MutableListIterator<E> = ListIteratorImpl(index)\n\n\n actual override fun subList(fromIndex: Int, toIndex: Int): MutableList<E> = SubList(this, fromIndex, toIndex)\n\n /**\n * Removes the range of elements from this list

starting from [fromIndex] and ending with but not including [toIndex].\n */\n protected open fun removeRange(fromIndex: Int, toIndex: Int) {\n val iterator = listIterator(fromIndex)\n repeat(toIndex - fromIndex) {\n iterator.next()\n iterator.remove()\n }\n }\n\n /**\n * Compares this list with another list instance with the ordered
structural equality.\n *\n * @return true, if [other] instance is a [List] of the same size, which contains the same elements in the same order.\n */\n override fun equals(other: Any?): Boolean {\n if (other === this) return true\n if (other !is List<*>) return false\n\n return AbstractList.orderedEquals(this, other)\n }\n\n /**\n * Returns the hash code value for this list.\n */\n override fun hashCode(): Int = AbstractList.orderedHashCode(this)\n\n\n private open inner class IteratorImpl : MutableIterator<E> {\n /** the index of the item that will be returned on the next call to [next]`()` */\n protected var index = 0\n /** the index of the item that was returned on the previous call to [next]`()`\n * or [ListIterator.previous]`()` (for `ListIterator`),\n * -1 if no such item exists\n */\n protected var last = -1\n\n override fun hasNext(): Boolean = index < size\n\n
override fun next(): E {\n if (!hasNext()) throw NoSuchElementException()\n last = index++\n return get(last)\n }\n\n override fun remove() {\n check(last != -1) { \"Call next() or previous() before removing element from the iterator.\" }\n\n removeAt(last)\n index = last\n last = -1\n }\n }\n\n /**\n * Implementation of `MutableListIterator` for abstract lists.\n */\n private inner class ListIteratorImpl(index: Int) : IteratorImpl(), MutableListIterator<E> {\n\n init {\n AbstractList.checkPositionIndex(index, this@AbstractMutableList.size)\n this.index = index\n }\n\n override fun hasPrevious(): Boolean = index > 0\n\n override fun nextIndex(): Int = index\n\n override fun previous(): E {\n if (!hasPrevious()) throw NoSuchElementException()\n\n last = --index\n return get(last)\n
}\n\n override fun previousIndex(): Int = index - 1\n\n override fun add(element: E) {\n add(index, element)\n index++\n last = -1\n }\n\n override fun set(element: E) {\n check(last != -1) { \"Call next() or previous() before updating element value with the iterator.\" }\n set(last, element)\n }\n }\n\n private class SubList<E>(private val list: AbstractMutableList<E>, private val fromIndex: Int, toIndex: Int) : AbstractMutableList<E>(), RandomAccess {\n private var _size: Int = 0\n\n init {\n AbstractList.checkRangeIndexes(fromIndex, toIndex, list.size)\n this._size = toIndex - fromIndex\n }\n\n override fun add(index: Int, element: E) {\n AbstractList.checkPositionIndex(index, _size)\n\n list.add(fromIndex + index, element)\n _size++\n }\n\n override fun get(index: Int): E {\n
AbstractList.checkElementIndex(index, _size)\n\n return list[fromIndex + index]\n }\n\n override fun removeAt(index: Int): E {\n AbstractList.checkElementIndex(index, _size)\n\n val result = list.removeAt(fromIndex + index)\n _size--\n return result\n }\n\n override fun set(index: Int, element: E): E {\n AbstractList.checkElementIndex(index, _size)\n\n return list.set(fromIndex + index, element)\n }\n\n override val size: Int get() = _size\n\n internal override fun checkIsMutable(): Unit = list.checkIsMutable()\n }\n\n}\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n/*\n * Based on GWT AbstractMap\n * Copyright 2007 Google Inc.\n */\n\npackage kotlin.collections\n\n/**\n * Provides
a skeletal implementation of the [MutableMap] interface.\n *\n * The implementor is required to implement [entries] property, which should return mutable set of map entries, and [put] function.\n *\n * @param K the type of map keys. The map is invariant in its key type.\n * @param V the type of map values. The map is invariant in its value type.\n */\npublic actual abstract class AbstractMutableMap<K, V> protected actual constructor() : AbstractMap<K, V>(), MutableMap<K, V> {\n\n /**\n * A mutable [Map.Entry] shared by several [Map] implementations.\n */\n internal open class SimpleEntry<K, V>(override val key: K, value: V) : MutableMap.MutableEntry<K, V> {\n constructor(entry: Map.Entry<K, V>) : this(entry.key, entry.value)\n\n
private var _value = value\n\n override val value: V get() = _value\n\n override fun setValue(newValue:

V): V {\n // Should check if the map containing this entry is mutable.\n // However, to not increase entry memory footprint it might be worthwhile not to check it here and\n // force subclasses that implement `build()` (freezing) operation to implement their own `MutableEntry`.\n// this@AbstractMutableMap.checkIsMutable()\n val oldValue = this._value\n this._value = newValue\n return oldValue\n }\n\n override fun hashCode(): Int = entryHashCode(this)\n override fun toString(): String = entryToString(this)\n override fun equals(other: Any?): Boolean = entryEquals(this, other)\n\n }\n\n // intermediate abstract class to workaround KT-43321\n internal abstract class AbstractEntrySet<E : Map.Entry<K, V>, K, V> : AbstractMutableSet<E>() {\n final override fun contains(element: E): Boolean = containsEntry(element)\n abstract fun containsEntry(element: Map.Entry<K, V>): Boolean\n final override fun remove(element: E): Boolean = removeEntry(element)\n abstract fun removeEntry(element: Map.Entry<K, V>): Boolean\n }\n\n actual override fun clear() {\n entries.clear()\n }\n\n private var _keys: MutableSet<K>? = null\n actual override val keys: MutableSet<K>\n get() {\n if (_keys == null) {\n _keys = object : AbstractMutableSet<K>() {\n override fun add(element: K): Boolean = throw UnsupportedOperationException(\"Add is not supported on keys\")\n override fun clear() {\n this@AbstractMutableMap.clear()\n }\n\n override operator fun contains(element: K): Boolean = containsKey(element)\n\n override operator fun iterator(): MutableIterator<K> {\n val entryIterator = entries.iterator()\n return object : MutableIterator<K> {\n override fun hasNext(): Boolean = entryIterator.hasNext()\n

override fun next(): K = entryIterator.next().key\n override fun remove() = entryIterator.remove()\n }\n }\n\n override fun remove(element: K): Boolean {\n checkIsMutable()\n if (containsKey(element)) {\n this@AbstractMutableMap.remove(element)\n return true\n }\n return false\n }\n\n override val size: Int get() = this@AbstractMutableMap.size\n\n override fun checkIsMutable(): Unit = this@AbstractMutableMap.checkIsMutable()\n }\n }\n return _keys!!\n }\n\n actual abstract override fun put(key: K, value: V): V?\n\n actual override fun putAll(from: Map<out K, V>) {\n checkIsMutable()\n for ((key, value) in from) {\n put(key, value)\n }\n }\n\n private var _values: MutableCollection<V>? = null\n actual override val values: MutableCollection<V>\n get() {\n if (_values == null) {\n _values = object : AbstractMutableCollection<V>() {\n override fun add(element: V): Boolean = throw UnsupportedOperationException(\"Add is not supported on values\")\n override fun clear() = this@AbstractMutableMap.clear()\n\n override operator fun contains(element: V): Boolean = containsValue(element)\n\n override operator fun iterator(): MutableIterator<V> {\n val entryIterator = entries.iterator()\n return object : MutableIterator<V> {\n override fun hasNext(): Boolean = entryIterator.hasNext()\n override fun next(): V = entryIterator.next().value\n override fun remove() = entryIterator.remove()\n }\n }\n\n override val size: Int get() = this@AbstractMutableMap.size\n\n override fun checkIsMutable(): Unit = this@AbstractMutableMap.checkIsMutable()\n }\n }\n return _values!!\n }\n\n actual override fun remove(key: K): V? {\n checkIsMutable()\n val iter = entries.iterator()\n while (iter.hasNext()) {\n val entry = iter.next()\n val k = entry.key\n if (key == k) {\n val value = entry.value\n iter.remove()\n return value\n }\n }\n return null\n }\n\n /**\n * This method is called every time when a mutating method is called on this mutable map.\n * Mutable maps that are built (frozen) must throw `UnsupportedOperationException`.\n */\n internal open fun checkIsMutable(): Unit {}\n}\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin.collections\n\n/**\n * Provides a skeletal implementation of the [MutableSet] interface.\n *\n * @param E the

type of elements contained in the set. The set is invariant in its element type.\n */\npublic actual abstract class AbstractMutableSet<E> protected actual constructor() : AbstractMutableCollection<E>(), MutableSet<E> {\n\n /**\n * Compares this set with another set instance with the unordered structural equality.\n *\n * @return `true`, if [other] instance is a [Set] of the same size, all elements of which are contained in this set.\n */\n override fun equals(other: Any?): Boolean {\n if (other === this) return true\n if (other !is Set<*>) return false\n return AbstractSet.setEquals(this, other)\n }\n\n /**\n * Returns the hash code value for this set.\n */\n override fun hashCode(): Int = AbstractSet.unorderedHashCode(this)\n\n}","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n/**\n * Provides a [MutableList] implementation, which uses a resizable array as its backing storage.\n *\n * This implementation doesn't provide a way to manage capacity, as backing JS array is resizeable itself.\n * There is no speed advantage to pre-allocating array sizes in JavaScript, so this implementation does not include any of the\n * capacity and \"growth increment\" concepts.\n */\npublic actual open class ArrayList<E> internal constructor(private var array: Array<Any?>) : AbstractMutableList<E>(), MutableList<E>, RandomAccess {\n private var isReadOnly: Boolean = false\n\n /**\n * Creates an empty [ArrayList].\n */\n public actual constructor() : this(emptyArray()) {}\n\n /**\n * Creates an empty [ArrayList].\n * @param initialCapacity initial capacity (ignored)\n */\n public actual constructor(initialCapacity: Int) : this(emptyArray()) {}\n\n /**\n * Creates an [ArrayList] filled from the [elements] collection.\n */\n public actual constructor(elements: Collection<E>) : this(elements.toTypedArray<Any?>()) {}\n\n @PublishedApi\n internal fun build(): List<E> {\n checkIsMutable()\n isReadOnly = true\n return this\n }\n\n /** Does nothing in this ArrayList implementation. */\n public actual fun trimToSize() {}\n\n /** Does nothing in this ArrayList implementation. */\n public actual fun ensureCapacity(minCapacity: Int) {}\n\n actual override val size: Int get() = array.size\n @Suppress(\"UNCHECKED_CAST\")\n actual override fun get(index: Int): E = array[rangeCheck(index)] as E\n actual override fun set(index: Int, element: E): E {\n checkIsMutable()\n rangeCheck(index)\n @Suppress(\"UNCHECKED_CAST\")\n return array[index].apply { array[index] = element } as E\n }\n\n actual override fun add(element: E): Boolean {\n checkIsMutable()\n array.asDynamic().push(element)\n modCount++\n return true\n }\n\n actual override fun add(index: Int, element: E): Unit {\n checkIsMutable()\n array.asDynamic().splice(insertionRangeCheck(index), 0, element)\n modCount++\n }\n\n actual override fun addAll(elements: Collection<E>): Boolean {\n checkIsMutable()\n if (elements.isEmpty()) return false\n\n array += elements.toTypedArray<Any?>()\n modCount++\n return true\n }\n\n actual override fun addAll(index: Int, elements: Collection<E>): Boolean {\n checkIsMutable()\n insertionRangeCheck(index)\n\n if (index == size) return addAll(elements)\n if (elements.isEmpty()) return false\n when (index) {\n size -> return addAll(elements)\n 0 -> array = elements.toTypedArray<Any?>() + array\n else -> array = array.copyOfRange(0, index).asDynamic().concat(elements.toTypedArray<Any?>(), array.copyOfRange(index, size))\n }\n\n modCount++\n return true\n }\n\n actual override fun removeAt(index: Int): E {\n checkIsMutable()\n rangeCheck(index)\n modCount++\n return if (index == lastIndex)\n array.asDynamic().pop()\n else\n array.asDynamic().splice(index, 1)[0]\n }\n\n actual override fun remove(element: E): Boolean {\n checkIsMutable()\n for (index in array.indices) {\n if (array[index] == element) {\n array.asDynamic().splice(index, 1)\n modCount++\n return true\n }\n }\n return false\n }\n\n override fun removeRange(fromIndex: Int, toIndex: Int) {\n checkIsMutable()\n modCount++\n array.asDynamic().splice(fromIndex, toIndex - fromIndex)\n }\n\n actual override fun clear() {\n checkIsMutable()\n array = emptyArray()\n modCount++\n }\n\n\n actual override fun indexOf(element: E): Int = array.indexOf(element)\n\n actual override fun lastIndexOf(element: E): Int = array.lastIndexOf(element)\n\n override fun toString() = arrayToString(array)\n\n

```
@Suppress(\"UNCHECKED_CAST\")\n    override fun <T> toArray(array: Array<T>): Array<T> {\n        if (array.size < size) {\n            return toArray() as Array<T>\n        }\n\n        (this.array as Array<T>).copyInto(array)\n\n        if (array.size > size) {\n            array[size] = null as T // null-terminate\n        }\n\n        return array\n    }\n\n    override fun toArray(): Array<Any?> {\n        return js(\"[]\").slice.call(array)\n    }\n\n\n    internal override
 fun checkIsMutable() {\n        if (isReadOnly) throw UnsupportedOperationException()\n    }\n\n    private fun rangeCheck(index: Int) = index.apply {\n        AbstractList.checkElementIndex(index, size)\n    }\n\n    private fun insertionRangeCheck(index: Int) = index.apply {\n        AbstractList.checkPositionIndex(index, size)\n    }\n}","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\ninternal fun <T> sortArrayWith(array: Array<out T>, comparison: (T, T) -> Int) {\n    if (getStableSortingIsSupported()) {\n        array.asDynamic().sort(comparison)\n    } else {\n        mergeSort(array.unsafeCast<Array<T>>(), 0, array.lastIndex, Comparator(comparison))\n    }\n}\n\ninternal fun <T> sortArrayWith(array: Array<out T>, comparator: Comparator<in T>) {\n    if (getStableSortingIsSupported()) {\n        val comparison = { a: T, b: T -> comparator.compare(a, b) }\n        array.asDynamic().sort(comparison)\n    } else {\n        mergeSort(array.unsafeCast<Array<T>>(), 0, array.lastIndex, comparator)\n    }\n}\n\ninternal fun <T> sortArrayWith(array: Array<out T>, fromIndex: Int, toIndex: Int, comparator: Comparator<in T>) {\n    if (fromIndex < toIndex - 1) {\n        mergeSort(array.unsafeCast<Array<T>>(), fromIndex, toIndex - 1, comparator)\n    }\n}\n\ninternal fun <T : Comparable<T>> sortArray(array: Array<out T>) {\n    if (getStableSortingIsSupported()) {\n        val comparison = { a: T, b: T -> a.compareTo(b) }\n        array.asDynamic().sort(comparison)\n    } else {\n        mergeSort(array.unsafeCast<Array<T>>(), 0, array.lastIndex, naturalOrder())\n    }\n}\n\nprivate var _stableSortingIsSupported: Boolean? = null\nprivate fun getStableSortingIsSupported(): Boolean {\n    _stableSortingIsSupported?.let { return it }\n    _stableSortingIsSupported = false\n\n
 val array = js(\"[]\").unsafeCast<Array<Int>>()\n    // known implementations may use stable sort for arrays of up to 512 elements\n    // so we create slightly more elements to test stability\n    for (index in 0 until 600) array.asDynamic().push(index)\n    val comparison = { a: Int, b: Int -> (a and 3) - (b and 3) }\n    array.asDynamic().sort(comparison)\n    for (index in 1 until array.size) {\n        val a = array[index - 1]\n        val b = array[index]\n        if ((a and 3) == (b and 3) && a >= b) return false\n    }\n    _stableSortingIsSupported = true\n
 return true\n}\n\n\nprivate fun <T> mergeSort(array: Array<T>, start: Int, endInclusive: Int, comparator: Comparator<in T>) {\n    val buffer = arrayOfNulls<Any?>(array.size).unsafeCast<Array<T>>()\n    val result = mergeSort(array, buffer, start, endInclusive, comparator)\n    if (result !== array) {\n        for (i in start..endInclusive) array[i] = result[i]\n    }\n}\n\n// Both start and end are inclusive indices.\nprivate
 fun <T> mergeSort(array: Array<T>, buffer: Array<T>, start: Int, end: Int, comparator: Comparator<in T>): Array<T> {\n    if (start == end) {\n        return array\n    }\n\n    val median = (start + end) / 2\n    val left = mergeSort(array, buffer, start, median, comparator)\n    val right = mergeSort(array, buffer, median + 1, end, comparator)\n\n    val target = if (left === buffer) array else buffer\n\n    // Merge.\n    var leftIndex = start\n    var rightIndex = median + 1\n    for (i in start..end) {\n        when {\n            leftIndex <= median && rightIndex <= end -> {\n                val leftValue = left[leftIndex]\n                val rightValue = right[rightIndex]\n\n                if (comparator.compare(leftValue, rightValue) <= 0) {\n                    target[i] = leftValue\n                    leftIndex++\n                } else {\n                    target[i] = rightValue\n                    rightIndex++\n                }\n            }\n            leftIndex
 <= median -> {\n                target[i] = left[leftIndex]\n                leftIndex++\n            }\n            else /* rightIndex <= end */ -> {\n                target[i] = right[rightIndex]\n                rightIndex++\n                Unit  // TODO: Fix KT-31506\n            }\n        }\n    }\n\n    return target\n}","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n\n@OptIn(ExperimentalUnsignedTypes::class)\n@SinceKotlin(\"1.3\")\n@kotlin.js.JsName(\"
```

```
contentDeepHashCodeImpl\")\ninternal fun <T> Array<out T>?.contentDeepHashCodeImpl(): Int {\n    if (this == null) return 0\n    var result = 1\n    for (element in this) {\n        val elementHash = when {\n            element == null -> 0\n            isArrayish(element) -> (element.unsafeCast<Array<*>>()).contentDeepHashCodeImpl()\n\n            element is UByteArray   -> element.contentHashCode()\n            element is UShortArray  -> element.contentHashCode()\n            element is UIntArray    -> element.contentHashCode()\n            element is ULongArray   -> element.contentHashCode()\n\n            else                    -> element.hashCode()\n        }\n\n        result = 31 * result + elementHash\n    }\n    return result\n}","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\ninternal interface EqualityComparator {\n    /**\n     * Subclasses must override to return a value indicating\n     * whether or not two keys or values are equal.\n     */\n    abstract fun equals(value1: Any?, value2: Any?): Boolean\n\n    /**\n     * Subclasses must override to return the hash code of a given key.\n     */\n    abstract fun getHashCode(value: Any?): Int\n\n\n    object HashCode : EqualityComparator {\n        override fun equals(value1: Any?, value2: Any?): Boolean = value1 == value2\n\n        override fun getHashCode(value: Any?): Int = value?.hashCode() ?: 0\n    }\n}","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n/*\n * Based on GWT AbstractHashMap\n * Copyright 2008 Google Inc.\n */\n\npackage kotlin.collections\n\nimport kotlin.collections.MutableMap.MutableEntry\n\n/**\n * Hash table based implementation of the [MutableMap] interface.\n *\n * This implementation makes no guarantees regarding the order of enumeration of [keys], [values] and [entries] collections.\n */\n// Classes that extend HashMap and implement `build()` (freezing) operation\n// have to make sure mutating methods check `checkIsMutable`.\npublic actual open class HashMap<K, V> : AbstractMutableMap<K, V>, MutableMap<K, V> {\n\n    private inner class EntrySet : AbstractEntrySet<MutableEntry<K, V>, K, V>() {\n\n        override fun add(element: MutableEntry<K, V>): Boolean = throw UnsupportedOperationException(\"Add is not supported on entries\")\n        override fun clear() {\n            this@HashMap.clear()\n        }\n\n        override fun containsEntry(element: Map.Entry<K, V>): Boolean = this@HashMap.containsEntry(element)\n\n        override operator fun iterator(): MutableIterator<MutableEntry<K, V>> = internalMap.iterator()\n\n        override fun removeEntry(element: Map.Entry<K, V>): Boolean {\n            if (contains(element)) {\n                this@HashMap.remove(element.key)\n                return true\n            }\n            return false\n        }\n\n        override val size: Int get() = this@HashMap.size\n    }\n\n\n    /**\n     * Internal implementation of the map: either string-based or hashcode-based.\n     */\n    private val internalMap: InternalMap<K, V>\n\n    private val equality: EqualityComparator\n\n    internal constructor(internalMap: InternalMap<K, V>) : super() {\n        this.internalMap = internalMap\n        this.equality = internalMap.equality\n    }\n\n    /**\n     * Constructs an empty [HashMap] instance.\n     */\n    actual constructor() : this(InternalHashCodeMap(EqualityComparator.HashCode))\n\n    /**\n     * Constructs an empty [HashMap] instance.\n     *\n     * @param  initialCapacity the initial capacity (ignored)\n     * @param  loadFactor      the load factor (ignored)\n     *\n     * @throws IllegalArgumentException if the initial capacity or load factor are negative\n     */\n    actual constructor(initialCapacity: Int, loadFactor: Float) : this() {\n        // This implementation of HashMap has no need of load factors or capacities.\n        require(initialCapacity >= 0) { \"Negative initial capacity: $initialCapacity\" }\n        require(loadFactor > 0) { \"Non-positive load factor: $loadFactor\" }\n    }\n\n    actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)\n\n\n    /**\n     * Constructs an instance of [HashMap] filled with the contents of the specified [original] map.\n     */\n    actual constructor(original: Map<out K, V>) : this() {\n        this.putAll(original)\n    }\n\n    actual override fun clear() {\n        internalMap.clear()\n        // structureChanged(this)\n    }\n\n    actual override fun containsKey(key: K): Boolean = internalMap.contains(key)\n\n    actual override fun containsValue(value: V): Boolean = internalMap.any { equality.equals(it.value, value) }\n\n    private var _entries: MutableSet<MutableMap.MutableEntry<K, V>>? =
```

null\n   actual override val entries: MutableSet<MutableMap.MutableEntry<K, V>>\n   get() {\n   if (_entries == null) {\n   _entries = createEntrySet()\n   }\n   return _entries!!\n   }\n\n   internal open fun createEntrySet():
MutableSet<MutableMap.MutableEntry<K, V>> = EntrySet()\n\n   actual override operator fun get(key: K): V? = internalMap.get(key)\n\n   actual override fun put(key: K, value: V): V? = internalMap.put(key, value)\n\n   actual override fun remove(key: K): V? = internalMap.remove(key)\n\n   actual override val size: Int get() = internalMap.size\n\n}\n\n/**\n * Constructs the specialized implementation of [HashMap] with [String] keys, which stores the keys as properties of\n * JS object without hashing them.\n */\npublic fun <V> stringMapOf(vararg pairs: Pair<String, V>): HashMap<String, V> {\n   return HashMap<String, V>(InternalStringMap(EqualityComparator.HashCode)).apply { putAll(pairs) }\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n/*\n * Based on GWT HashSet\n * Copyright 2008 Google Inc.\n */\n\npackage
kotlin.collections\n\n/**\n * The implementation of the [MutableSet] interface, backed by a [HashMap] instance.\n */\n// Classes that extend HashSet and implement `build()` (freezing) operation\n// have to make sure mutating methods check `checkIsMutable`.\npublic actual open class HashSet<E> : AbstractMutableSet<E>, MutableSet<E> {\n\n   internal val map: HashMap<E, Any>\n\n   /**\n   * Constructs a new empty [HashSet].\n   */\n   actual constructor() {\n   map = HashMap<E, Any>()\n   }\n\n   /**\n   * Constructs a new [HashSet] filled with the elements of the specified collection.\n   */\n   actual constructor(elements: Collection<E>) {\n   map = HashMap<E, Any>(elements.size)\n   addAll(elements)\n   }\n\n   /**\n   * Constructs a new empty [HashSet].\n   *\n   * @param initialCapacity the initial capacity (ignored)\n   * @param loadFactor   the load factor (ignored)\n   *\n   * @throws IllegalArgumentException if the initial
capacity or load factor are negative\n   */\n   actual constructor(initialCapacity: Int, loadFactor: Float) {\n   map = HashMap<E, Any>(initialCapacity, loadFactor)\n   }\n\n   actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)\n\n   /**\n   * Protected constructor to specify the underlying map. This is used by\n   * LinkedHashSet.\n\n   * @param map underlying map to use.\n   */\n   internal constructor(map: HashMap<E, Any>) {\n   this.map = map\n   }\n\n   actual override fun add(element: E): Boolean {\n   val old = map.put(element, this)\n   return old == null\n   }\n\n   actual override fun clear() {\n   map.clear()\n   }\n\n// public override fun clone(): Any {\n//   return HashSet<E>(this)\n// }\n\n   actual override operator fun contains(element: E): Boolean = map.containsKey(element)\n\n   actual override fun isEmpty(): Boolean = map.isEmpty()\n\n   actual override fun iterator(): MutableIterator<E>
= map.keys.iterator()\n\n   actual override fun remove(element: E): Boolean = map.remove(element) != null\n\n   actual override val size: Int get() = map.size\n\n}\n\n/**\n * Creates a new instance of the specialized implementation of [HashSet] with the specified [String] elements,\n * which elements the keys as properties of JS object without hashing them.\n */\npublic fun stringSetOf(vararg elements: String): HashSet<String> {\n   return HashSet(stringMapOf<Any>()).apply { addAll(elements) }\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n/*\n * Based on GWT InternalHashCodeMap\n * Copyright 2008 Google Inc.\n */\n\npackage kotlin.collections\n\nimport kotlin.collections.MutableMap.MutableEntry\nimport kotlin.collections.AbstractMutableMap.SimpleEntry\n\n/**\n * A simple wrapper around JavaScriptObject to provide
[java.util.Map]-like semantics for any\n * key type.\n *\n *\n * Implementation notes:\n *\n *\n * A key's hashCode is the index in backingMap which should contain that key. Since several keys may\n * have the same hash, each value in hashCodeMap is actually an array containing all entries whose\n * keys share the same hash.\n */\ninternal class InternalHashCodeMap<K, V>(override val equality: EqualityComparator) : InternalMap<K, V> {\n\n   private var backingMap: dynamic = createJsMap()\n   override var size: Int = 0\n   private set\n\n   override fun put(key: K, value: V): V? {\n   val hashCode = equality.getHashCode(key)\n   val chainOrEntry = getChainOrEntryOrNull(hashCode)\n   if (chainOrEntry == null) {\n   // This is a new chain, put it to the

```
map.\n            backingMap[hashCode] = SimpleEntry(key, value)\n        } else {\n            if (chainOrEntry !is
Array<*>) {\n                // It is an entry\n                val entry: SimpleEntry<K,
 V> = chainOrEntry\n                if (equality.equals(entry.key, key)) {\n                    return entry.setValue(value)\n
        } else {\n                    backingMap[hashCode] = arrayOf(entry, SimpleEntry(key, value))\n
size++\n                    return null\n                }\n            } else {\n                // Chain already exists, perhaps key also
exists.\n                val chain: Array<MutableEntry<K, V>> = chainOrEntry\n                val entry =
chain.findEntryInChain(key)\n                if (entry != null) {\n                    return entry.setValue(value)\n                }\n
                chain.asDynamic().push(SimpleEntry(key, value))\n            }\n        }\n        size++\n//
structureChanged(host)\n        return null\n    }\n\n    override fun remove(key: K): V? {\n        val hashCode =
equality.getHashCode(key)\n        val chainOrEntry = getChainOrEntryOrNull(hashCode) ?: return null\n        if
(chainOrEntry
 !is Array<*>) {\n            val entry: MutableEntry<K, V> = chainOrEntry\n            if (equality.equals(entry.key,
key)) {\n                jsDeleteProperty(backingMap, hashCode)\n                size--\n                return entry.value\n
} else {\n                return null\n            }\n        } else {\n            val chain: Array<MutableEntry<K, V>> =
chainOrEntry\n            for (index in chain.indices) {\n                val entry = chain[index]\n                if
(equality.equals(key, entry.key)) {\n                    if (chain.size == 1) {\n                        chain.asDynamic().length =
0\n                        // remove the whole array\n                        jsDeleteProperty(backingMap, hashCode)\n
} else {\n                        // splice out the entry we're removing\n                        chain.asDynamic().splice(index, 1)\n
                }\n                    size--\n//                  structureChanged(host)\n
                    return entry.value\n                }\n            }\n        }\n        return null\n    }\n\n    override fun clear() {\n
 backingMap = createJsMap()\n        size = 0\n    }\n\n    override fun contains(key: K): Boolean = getEntry(key) !=
null\n\n    override fun get(key: K): V? = getEntry(key)?.value\n\n    private fun getEntry(key: K): MutableEntry<K,
V>? {\n        val chainOrEntry = getChainOrEntryOrNull(equality.getHashCode(key)) ?: return null\n        if
(chainOrEntry !is Array<*>) {\n            val entry: MutableEntry<K, V> = chainOrEntry\n            if
(equality.equals(entry.key, key)) {\n                return entry\n            } else {\n                return null\n            }\n        }
else {\n            val chain: Array<MutableEntry<K, V>> = chainOrEntry\n            return
chain.findEntryInChain(key)\n        }\n    }\n\n    private fun Array<MutableEntry<K, V>>.findEntryInChain(key:
K): MutableEntry<K, V>? =\n        firstOrNull { entry ->
 equality.equals(entry.key, key) }\n\n    override fun iterator(): MutableIterator<MutableEntry<K, V>> {\n\n
return object : MutableIterator<MutableEntry<K, V>> {\n            var state = -1 // -1 not ready, 0 - ready, 1 -
done\n\n            val keys: Array<String> = js(\"Object\").keys(backingMap)\n            var keyIndex = -1\n\n
var chainOrEntry: dynamic = null\n            var isChain = false\n            var itemIndex = -1\n            var lastEntry:
MutableEntry<K, V>? = null\n\n            private fun computeNext(): Int {\n                if (chainOrEntry != null &&
isChain) {\n                    val chainSize: Int = chainOrEntry.unsafeCast<Array<MutableEntry<K, V>>>().size\n
        if (++itemIndex < chainSize)\n                        return 0\n                }\n\n                if (++keyIndex < keys.size)
{\n                    chainOrEntry = backingMap[keys[keyIndex]]\n                    isChain = chainOrEntry is Array<*>\n

    itemIndex = 0\n                    return 0\n                } else {\n                    chainOrEntry = null\n                    return
1\n                }\n            }\n\n            override fun hasNext(): Boolean {\n                if (state == -1)\n                    state
= computeNext()\n                return state == 0\n            }\n\n            override fun next(): MutableEntry<K, V> {\n
        if (!hasNext()) throw NoSuchElementException()\n                val lastEntry = if (isChain) {\n
chainOrEntry.unsafeCast<Array<MutableEntry<K, V>>>()[itemIndex]\n                } else {\n
chainOrEntry.unsafeCast<MutableEntry<K, V>>()\n                }\n                this.lastEntry = lastEntry\n
state = -1\n                return lastEntry\n            }\n\n            override fun remove() {\n
checkNotNull(lastEntry)\n                this@InternalHashCodeMap.remove(lastEntry!!.key)\n
    lastEntry = null\n                // the chain being iterated just got modified by InternalHashCodeMap.remove\n
    itemIndex--\n            }\n        }\n    }\n\n    private fun getChainOrEntryOrNull(hashCode: Int): dynamic {\n
val chainOrEntry = backingMap[hashCode]\n        return if (chainOrEntry === undefined) null else chainOrEntry\n
```

}\n\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n/**\n * The common interface of [InternalStringMap] and [InternalHashCodeMap].\n */\ninternal interface InternalMap<K, V> : MutableIterable<MutableMap.MutableEntry<K, V>> {\n val equality: EqualityComparator\n val size: Int\n operator fun contains(key: K): Boolean\n operator fun get(key: K): V?\n\n fun put(key: K, value: V): V?\n fun remove(key: K): V?\n

fun clear(): Unit\n\n fun createJsMap(): dynamic {\n val result = js(\"Object.create(null)\")\n // force to switch object representation to dictionary mode\n result[\"foo\"] = 1\n jsDeleteProperty(result, \"foo\")\n return result\n }\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n/*\n * Based on GWT InternalStringMap\n * Copyright 2008 Google Inc.\n */\npackage kotlin.collections\n\nimport kotlin.collections.MutableMap.MutableEntry\n\n/**\n * A simple wrapper around JavaScript Map for key type is string.\n *\n * Though this map is instantiated only with K=String, the K type is not fixed to String statically,\n * because we want to have it erased to Any? in order not to generate type-safe override bridges for\n * [get], [contains], [remove] etc, if they ever are generated.\n */\ninternal class InternalStringMap<K, V>(override val equality: EqualityComparator) : InternalMap<K, V> {\n\n private var backingMap: dynamic = createJsMap()\n override var size: Int = 0\n private set\n\n// /**\n// * A mod count to track 'value' replacements in map to ensure that the 'value' that we have in the\n// * iterator entry is guaranteed to be still correct.\n// * This is to optimize for the common scenario where the values are not modified during\n// * iterations where the entries are never stale.\n// */\n// private var valueMod: Int = 0\n\n override operator fun contains(key: K): Boolean {\n if (key !is String) return false\n return backingMap[key] !== undefined\n }\n\n override operator fun get(key: K): V? {\n if (key !is String) return null\n val value = backingMap[key]\n return if (value !== undefined) value.unsafeCast<V>() else null\n }\n\n\n override fun put(key: K, value: V): V?
{\n require(key is String)\n val oldValue = backingMap[key]\n backingMap[key] = value\n\n if (oldValue === undefined) {\n size++\n// structureChanged(host)\n return null\n } else {\n// valueMod++\n return oldValue.unsafeCast<V>()\n }\n }\n\n override fun remove(key: K): V? {\n if (key !is String) return null\n val value = backingMap[key]\n if (value !== undefined) {\n jsDeleteProperty(backingMap, key)\n size--\n// structureChanged(host)\n return value.unsafeCast<V>()\n } else {\n// valueMod++\n return null\n }\n }\n\n\n override fun clear() {\n backingMap = createJsMap()\n size = 0\n }\n\n\n override fun iterator(): MutableIterator<MutableEntry<K, V>> {\n return object : MutableIterator<MutableEntry<K, V>> {\n private val keys: Array<String>
= js(\"Object\").keys(backingMap)\n private val iterator = keys.iterator()\n private var lastKey: String? = null\n\n override fun hasNext(): Boolean = iterator.hasNext()\n\n override fun next(): MutableEntry<K, V> {\n val key = iterator.next()\n lastKey = key\n @Suppress(\"UNCHECKED_CAST\")\n return newMapEntry(key as K)\n }\n\n override fun remove() {\n @Suppress(\"UNCHECKED_CAST\")\n this@InternalStringMap.remove(checkNotNull(lastKey) as K)\n }\n }\n }\n\n private fun newMapEntry(key: K): MutableEntry<K, V> = object : MutableEntry<K, V> {\n override val key: K get() = key\n override val value: V get() = this@InternalStringMap[key].unsafeCast<V>()\n\n override fun setValue(newValue: V): V = this@InternalStringMap.put(key, newValue).unsafeCast<V>()\n\n override fun hashCode(): Int
= AbstractMap.entryHashCode(this)\n override fun toString(): String = AbstractMap.entryToString(this)\n override fun equals(other: Any?): Boolean = AbstractMap.entryEquals(this, other)\n }\n}\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n/*\n * Based on GWT

```
LinkedHashMap\n * Copyright 2008 Google Inc.\n */\npackage kotlin.collections\n\nimport
kotlin.collections.MutableMap.MutableEntry\n\n/**\n * Hash table based implementation of the [MutableMap]
interface, which additionally preserves the insertion order\n * of entries during the iteration.\n *\n * The insertion
order is preserved by maintaining a doubly-linked list of all of its entries.\n */\npublic actual open class
LinkedHashMap<K, V> : HashMap<K, V>, MutableMap<K, V> {\n\n    /**\n     * The entry we use includes
next/prev pointers for a doubly-linked
 circular\n     * list with a head node. This reduces the special cases we have to deal with\n     * in the list
operations.\n\n     * Note that we duplicate the key from the underlying hash map so we can find\n     * the eldest
entry. The alternative would have been to modify HashMap so more\n     * of the code was directly usable here, but
this would have added some\n     * overhead to HashMap, or to reimplement most of the HashMap code here with\n
 * small modifications. Paying a small storage cost only if you use\n     * LinkedHashMap and minimizing code size
seemed like a better tradeoff\n     */\n    private inner class ChainEntry<K, V>(key: K, value: V) :
AbstractMutableMap.SimpleEntry<K, V>(key, value) {\n        internal var next: ChainEntry<K, V>? = null\n
internal var prev: ChainEntry<K, V>? = null\n\n        override fun setValue(newValue: V): V {\n
this@LinkedHashMap.checkIsMutable()\n            return super.setValue(newValue)\n        }\n
  }\n\n    private inner class EntrySet : AbstractEntrySet<MutableEntry<K, V>, K, V>() {\n\n        private inner
class EntryIterator : MutableIterator<MutableEntry<K, V>> {\n            // The last entry that was returned from this
iterator.\n            private var last: ChainEntry<K, V>? = null\n\n            // The next entry to return from this
iterator.\n            private var next: ChainEntry<K, V>? = null\n\n            init {\n                next = head\n//
recordLastKnownStructure(map, this)\n            }\n\n            override fun hasNext(): Boolean {\n                return
next !== null\n            }\n\n            override fun next(): MutableEntry<K, V> {\n//
checkStructuralChange(map, this)\n                if (!hasNext()) throw NoSuchElementException()\n\n                val
current = next!!\n                last = current\n                next = current.next.takeIf { it !== head }\n                return
current\n            }\n\n
        override fun remove() {\n                check(last != null)\n                this@EntrySet.checkIsMutable()\n//
checkStructuralChange(map, this)\n\n                last!!.remove()\n                map.remove(last!!.key)\n//
recordLastKnownStructure(map, this)\n                last = null\n            }\n        }\n\n        override fun add(element:
MutableEntry<K, V>): Boolean = throw UnsupportedOperationException(\"Add is not supported on entries\")\n
override fun clear() {\n            this@LinkedHashMap.clear()\n        }\n\n        override fun containsEntry(element:
Map.Entry<K, V>): Boolean = this@LinkedHashMap.containsEntry(element)\n\n        override operator fun
iterator(): MutableIterator<MutableEntry<K, V>> = EntryIterator()\n\n        override fun removeEntry(element:
Map.Entry<K, V>): Boolean {\n            checkIsMutable()\n            if (contains(element)) {\n
this@LinkedHashMap.remove(element.key)\n
        return true\n            }\n            return false\n        }\n\n        override val size: Int get() =
this@LinkedHashMap.size\n\n        override fun checkIsMutable(): Unit =
this@LinkedHashMap.checkIsMutable()\n    }\n\n\n    /*\n    * The head of the insert order chain, which is a doubly-
linked circular\n    * list.\n    *\n    * The most recently inserted node is at the end of the chain, ie.\n    * chain.prev.\n
*/\n    private var head: ChainEntry<K, V>? = null\n\n    /**\n     * Add this node to the end of the chain.\n     */\n
private fun ChainEntry<K, V>.addToEnd() {\n        // This entry is not in the list.\n        check(next == null && prev
== null)\n\n        val _head = head\n        if (_head == null) {\n            head = this\n            next = this\n            prev =
this\n        } else {\n            // Chain is valid.\n            val _tail = checkNotNull(_head.prev)\n            // Update me.\n
        prev = _tail\n            next = _head\n
       // Update my new siblings: current head and old tail\n            _head.prev = this\n            _tail.next = this\n
}\n    }\n\n    /**\n     * Remove this node from the chain it is a part of.\n     */\n    private fun ChainEntry<K,
V>.remove() {\n        if (this.next === this) {\n            // if this is single element, remove head\n            head = null\n
    } else {\n            if (head === this) {\n                // if this is first element, move head to next\n                head =
next\n            }\n            next!!.prev = prev\n            prev!!.next = next\n        }\n        next = null\n        prev = null\n
 }\n\n    /*\n    * The hashmap that keeps track of our entries and the chain. Note that we\n    * duplicate the key here
```

to eliminate changes to HashMap and minimize the\n * code here, at the expense of additional space.\n */\n private val map: HashMap<K, ChainEntry<K, V>>\n\n private var isReadOnly: Boolean = false\n\n /**\n * Constructs an empty [LinkedHashMap] instance.\n */\n actual constructor() : super() {\n map = HashMap<K, ChainEntry<K, V>>()\n }\n\n internal constructor(backingMap: HashMap<K, Any>) : super() {\n @Suppress(\"UNCHECKED_CAST\") // expected to work due to erasure\n map = backingMap as HashMap<K, ChainEntry<K, V>>\n }\n\n /**\n * Constructs an empty [LinkedHashMap] instance.\n *\n * @param initialCapacity the initial capacity (ignored)\n * @param loadFactor the load factor (ignored)\n *\n * @throws IllegalArgumentException if the initial capacity or load factor are negative\n */\n actual constructor(initialCapacity: Int, loadFactor: Float) : super(initialCapacity, loadFactor) {\n map = HashMap<K, ChainEntry<K, V>>()\n }\n\n actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)\n\n /**\n * Constructs an instance of [LinkedHashMap] filled with the contents of the specified [original] map.\n */\n actual constructor(original: Map<out K, V>) {\n map = HashMap<K, ChainEntry<K, V>>()\n this.putAll(original)\n }\n\n @PublishedApi\n internal fun build(): Map<K, V> {\n checkIsMutable()\n isReadOnly = true\n return this\n }\n\n actual override fun clear() {\n checkIsMutable()\n map.clear()\n head = null\n }\n\n\n// override fun clone(): Any {\n// return LinkedHashMap(this)\n// }\n\n\n actual override fun containsKey(key: K): Boolean = map.containsKey(key)\n\n actual override fun containsValue(value: V): Boolean {\n var node: ChainEntry<K, V> = head ?: return false\n do {\n if (node.value == value) {\n return true\n }\n node = node.next!!\n } while (node !== head)\n return false\n }\n\n\n internal override fun createEntrySet(): MutableSet<MutableMap.MutableEntry<K, V>> = EntrySet()\n\n actual override operator fun get(key: K): V? = map.get(key)?.value\n\n actual override fun put(key: K, value: V): V? {\n checkIsMutable()\n\n val old = map.get(key)\n if (old == null) {\n val newEntry = ChainEntry(key, value)\n map.put(key, newEntry)\n newEntry.addToEnd()\n return null\n } else {\n return old.setValue(value)\n }\n }\n\n actual override fun remove(key: K): V? {\n checkIsMutable()\n\n val entry = map.remove(key)\n if (entry != null) {\n entry.remove()\n return entry.value\n }\n return null\n }\n\n actual override val size: Int get() = map.size\n\n internal override fun checkIsMutable() {\n if (isReadOnly) throw UnsupportedOperationException()\n }\n}\n\n/**\n * Constructs the specialized implementation of [LinkedHashMap] with [String] keys, which stores the keys as properties of\n * JS object without hashing them.\n */\npublic fun <V> linkedStringMapOf(vararg pairs: Pair<String, V>): LinkedHashMap<String, V> {\n return LinkedHashMap<String, V>(stringMapOf<Any>()).apply { putAll(pairs) }\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n/*\n * Based on GWT LinkedHashSet\n * Copyright 2008 Google Inc.\n */\n\npackage kotlin.collections\n\n/**\n * The implementation of the [MutableSet] interface, backed by a [LinkedHashMap] instance.\n *\n * This implementation preserves the insertion order of elements during the iteration.\n */\npublic actual open class LinkedHashSet<E> : HashSet<E>, MutableSet<E> {\n\n internal constructor(map: LinkedHashMap<E, Any>) : super(map)\n\n /**\n * Constructs a new empty [LinkedHashSet].\n */\n actual constructor() : super(LinkedHashMap<E, Any>())\n\n /**\n *

Constructs a new [LinkedHashSet] filled with the elements of the specified collection.\n */\n actual constructor(elements: Collection<E>) : super(LinkedHashMap<E, Any>()) {\n addAll(elements)\n }\n\n /**\n * Constructs a new empty [LinkedHashSet].\n *\n * @param initialCapacity the initial capacity (ignored)\n * @param loadFactor the load factor (ignored)\n *\n * @throws IllegalArgumentException if the initial capacity or load factor are negative\n */\n actual constructor(initialCapacity: Int, loadFactor: Float) : super(LinkedHashMap<E, Any>(initialCapacity, loadFactor))\n\n actual constructor(initialCapacity: Int) : this(initialCapacity, 0.0f)\n\n @PublishedApi\n internal fun build(): Set<E> {\n (map as LinkedHashMap<E, Any>).build()\n return this\n }\n\n internal override fun checkIsMutable(): Unit = map.checkIsMutable()\n\n// public override fun clone(): Any {\n// return LinkedHashSet(this)\n//

```
}\n\n}\n\n/**\n * Creates a new instance of the specialized implementation of [LinkedHashSet] with the specified [String] elements,\n * which elements the keys as properties of JS object without hashing them.\n */\npublic fun linkedStringSetOf(vararg elements: String): LinkedHashSet<String> {\n    return LinkedHashSet(linkedStringMapOf<Any>()).apply { addAll(elements) }\n}\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\nimport kotlin.contracts.*\n\n\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\n@Deprecated(\"Synchronization on any object is not supported in Kotlin/JS\", ReplaceWith(\"run(block)\"))\n@kotlin.internal.InlineOnly\n@Suppress(\"UNUSED_PARAMETER\")\npublic inline fun <R> synchronized(lock: Any, block: () -> R): R {\n    contract {\n        callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n    }\n    return block()\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.io\n\ninternal abstract class BaseOutput {\n    open fun println() {\n        print(\"\\n\")\n    }\n\n    open fun println(message: Any?) {\n        print(message)\n        println()\n    }\n\n    abstract fun print(message: Any?)\n\n    open fun flush() {}\n}\n\n/** JsName used to make the declaration available outside of module to test it */\n@JsName(\"NodeJsOutput\")\ninternal class NodeJsOutput(val outputStream: dynamic) : BaseOutput() {\n    override fun print(message: Any?) {\n        // TODO: Using local variable because of bug in block decomposition lowering in IR backend\n        val messageString = String(message)\n        outputStream.write(messageString)\n    }\n}\n\n/** JsName used to make the declaration available outside of module to test it */\n@JsName(\"OutputToConsoleLog\")\ninternal class OutputToConsoleLog : BaseOutput() {\n    override fun print(message: Any?) {\n        console.log(message)\n    }\n\n    override fun println(message: Any?) {\n        console.log(message)\n    }\n\n    override fun println() {\n        console.log(\"\")\n    }\n}\n\n/** JsName used to make the declaration available outside of module to test it and use at try.kotl.in */\n@JsName(\"BufferedOutput\")\ninternal open class BufferedOutput : BaseOutput() {\n    var buffer = \"\"\n\n    override fun print(message: Any?) {\n        buffer += String(message)\n    }\n\n    override fun flush() {\n        buffer = \"\"\n    }\n}\n\n/** JsName used to make the declaration available outside of module to test it */\n@JsName(\"BufferedOutputToConsoleLog\")\ninternal class BufferedOutputToConsoleLog : BufferedOutput() {\n    override fun print(message: Any?) {\n        var s = String(message)\n        val i = s.nativeLastIndexOf(\"\\n\", 0)\n        if (i >= 0) {\n            buffer += s.substring(0, i)\n            flush()\n            s = s.substring(i + 1)\n        }\n        buffer += s\n    }\n\n    override fun flush() {\n        console.log(buffer)\n        buffer = \"\"\n    }\n}\n\n/** JsName used to make the declaration available outside of module to test it and use at try.kotl.in */\n@JsName(\"output\")\ninternal var output = run {\n    val isNode: Boolean = js(\"typeof process !== 'undefined' && process.versions && !!process.versions.node\")\n    if (isNode) NodeJsOutput(js(\"process.stdout\")) else BufferedOutputToConsoleLog()\n}\n\n@kotlin.internal.InlineOnly\nprivate inline fun String(value: Any?): String = js(\"String\")(value)\n\n/** Prints the line separator to the standard output stream. */\npublic actual fun println() {\n    output.println()\n}\n\n/** Prints the given [message] and the line separator to the standard output stream. */\npublic actual fun println(message: Any?) {\n    output.println(message)\n}\n\n/** Prints the given [message] to the standard output stream. */\npublic actual fun print(message: Any?) {\n    output.print(message)\n}\n\n@SinceKotlin(\"1.6\")\npublic actual fun readln(): String = throw UnsupportedOperationException(\"readln is not supported in Kotlin/JS\")\n\n@SinceKotlin(\"1.6\")\npublic actual fun readlnOrNull(): String? = throw UnsupportedOperationException(\"readlnOrNull is not supported in Kotlin/JS\")","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines\n\nimport kotlin.coroutines.intrinsics.CoroutineSingletons.*\nimport kotlin.coroutines.intrinsics.COROUTINE_SUSPENDED\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\ninternal actual class SafeContinuation<in T>\ninternal actual constructor(\n    private val delegate: Continuation<T>,\n
```

```
    initialResult: Any?\n) : Continuation<T> {\n    @PublishedApi\n    internal actual constructor(delegate:
Continuation<T>) : this(delegate, UNDECIDED)\n\n    public actual override val context: CoroutineContext\n
get() = delegate.context\n\n    private var result: Any? = initialResult\n\n    public actual override fun
resumeWith(result: Result<T>) {\n        val cur = this.result\n        when {\n            cur === UNDECIDED -> {\n
        this.result = result.value\n            }\n            cur === COROUTINE_SUSPENDED -> {\n                this.result =
RESUMED\n                delegate.resumeWith(result)\n            }\n            else -> throw
IllegalStateException(\"Already resumed\")\n        }\n    }\n\n    @PublishedApi\n    internal actual fun
getOrThrow(): Any? {\n        if (result === UNDECIDED) {\n            result = COROUTINE_SUSPENDED\n
return COROUTINE_SUSPENDED\n        }\n        val result = this.result\n        return when {\n
      result === RESUMED -> COROUTINE_SUSPENDED // already called continuation, indicate
COROUTINE_SUSPENDED upstream\n            result is Result.Failure -> throw result.exception\n            else ->
result // either COROUTINE_SUSPENDED or data\n        }\n    }\n}\n","/*\n * Copyright 2010-2020 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.coroutines.cancellation\n\n@SinceKotlin(\"1.4\")\npublic actual open class CancellationException :
IllegalStateException {\n    actual constructor() : super()\n    actual constructor(message: String?) : super(message)\n
  constructor(message: String?, cause: Throwable?) : super(message, cause)\n    constructor(cause: Throwable?) :
super(cause)\n}","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n *
Use of this source code is governed by the Apache 2.0
 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines.js.internal\n\nimport
kotlin.coroutines.Continuation\nimport
kotlin.coroutines.EmptyCoroutineContext\n\n@PublishedApi\n@SinceKotlin(\"1.3\")\ninternal val
EmptyContinuation = Continuation<Any?>(EmptyCoroutineContext) { result ->\n    result.getOrThrow()\n}","/*\n
* Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage
kotlin.js\n\n/**\n * Exposes the [Date API](https://developer.mozilla.org/en-
US/docs/Web/JavaScript/Reference/Global_Objects/Date) to Kotlin.\n
*/\n@Suppress(\"NOT_DOCUMENTED\")\npublic external class Date() {\n    public constructor(milliseconds:
Number)\n\n    public constructor(dateString: String)\n\n    public constructor(year: Int, month: Int)\n\n    public
constructor(year: Int, month: Int, day: Int)\n\n    public
 constructor(year: Int, month: Int, day: Int, hour: Int)\n\n    public constructor(year: Int, month: Int, day: Int, hour:
Int, minute: Int)\n\n    public constructor(year: Int, month: Int, day: Int, hour: Int, minute: Int, second: Int)\n\n
public constructor(year: Int, month: Int, day: Int, hour: Int, minute: Int, second: Int, millisecond: Number)\n\n
public fun getDate(): Int\n\n    public fun getDay(): Int\n\n    public fun getFullYear(): Int\n\n    public fun
getHours(): Int\n\n    public fun getMilliseconds(): Int\n\n    public fun getMinutes(): Int\n\n    public fun
getMonth(): Int\n\n    public fun getSeconds(): Int\n\n    public fun getTime(): Double\n\n    public fun
getTimezoneOffset(): Int\n\n    public fun getUTCDate(): Int\n\n    public fun getUTCDay(): Int\n\n    public fun
getUTCFullYear(): Int\n\n    public fun getUTCHours(): Int\n\n    public fun getUTCMilliseconds(): Int\n\n    public
fun getUTCMinutes(): Int\n\n    public fun getUTCMonth(): Int\n\n    public fun
 getUTCSeconds(): Int\n\n    public fun toDateString(): String\n\n    public fun toISOString(): String\n\n    public fun
toJSON(): Json\n\n    public fun toLocaleDateString(locales: Array<String> = definedExternally, options:
LocaleOptions = definedExternally): String\n\n    public fun toLocaleDateString(locales: String, options:
LocaleOptions = definedExternally): String\n\n    public fun toLocaleString(locales: Array<String> =
definedExternally, options: LocaleOptions = definedExternally): String\n\n    public fun toLocaleString(locales:
String, options: LocaleOptions = definedExternally): String\n\n    public fun toLocaleTimeString(locales:
Array<String> = definedExternally, options: LocaleOptions = definedExternally): String\n\n    public fun
toLocaleTimeString(locales: String, options: LocaleOptions = definedExternally): String\n\n    public fun
toTimeString(): String\n\n    public fun toUTCString(): String\n\n    public companion object {\n        public fun
```

now(): Double\n\n
public fun parse(dateString: String): Double\n\n public fun UTC(year: Int, month: Int): Double\n\n public fun UTC(year: Int, month: Int, day: Int): Double\n\n public fun UTC(year: Int, month: Int, day: Int, hour: Int): Double\n\n public fun UTC(year: Int, month: Int, day: Int, hour: Int, minute: Int): Double\n\n public fun UTC(year: Int, month: Int, day: Int, hour: Int, minute: Int, second: Int): Double\n\n public fun UTC(year: Int, month: Int, day: Int, hour: Int, minute: Int, second: Int, millisecond: Number): Double\n }\n\n public interface LocaleOptions {\n public var localeMatcher: String?\n\n public var timeZone: String?\n\n public var hour12: Boolean?\n\n public var formatMatcher: String?\n\n public var weekday: String?\n\n public var era: String?\n\n public var year: String?\n\n public var month: String?\n\n public var day: String?\n\n public var hour: String?\n\n
public var minute: String?\n\n public var second: String?\n\n public var timeZoneName: String?\n }\n}\n\npublic inline fun dateLocaleOptions(init: Date.LocaleOptions.() -> Unit): Date.LocaleOptions {\n val result = js(\"new Object()\").unsafeCast<Date.LocaleOptions>()\n init(result)\n return result\n}","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.dom\n\nimport org.w3c.dom.Document\nimport org.w3c.dom.Element\nimport kotlin.internal.LowPriorityInOverloadResolution\nimport kotlinx.dom.appendElement as newAppendElement\nimport kotlinx.dom.createElement as newCreateElement\n\n/**\n * Creates a new element with the specified [name].\n *\n * The element is initialized with the specified [init] function.\n */\n@LowPriorityInOverloadResolution\n@Deprecated(\n message =
\"This API is moved to another package, use 'kotlinx.dom.createElement' instead.\",\n replaceWith = ReplaceWith(\"this.createElement(name, init)\",
\"kotlinx.dom.createElement\")\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\npublic inline fun Document.createElement(name: String, noinline init: Element.() -> Unit): Element = this.newCreateElement(name, init)\n\n/**\n * Appends a newly created element with the specified [name] to this element.\n *\n * The element is initialized with the specified [init] function.\n */\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \"This API is moved to another package, use 'kotlinx.dom.appendElement' instead.\",\n replaceWith = ReplaceWith(\"this.appendElement(name, init)\",
\"kotlinx.dom.appendElement\")\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\npublic inline fun Element.appendElement(name: String, noinline init: Element.() -> Unit): Element = this.newAppendElement(name,
init)\n\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.dom\n\nimport org.w3c.dom.Element\nimport kotlin.internal.LowPriorityInOverloadResolution\nimport kotlinx.dom.addClass as newAddClass\nimport kotlinx.dom.hasClass as newHasClass\nimport kotlinx.dom.removeClass as newRemoveClass\n\n/** Returns true if the element has the given CSS class style in its 'class' attribute
*/\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \"This API is moved to another package, use 'kotlinx.dom.hasClass' instead.\",\n replaceWith = ReplaceWith(\"this.hasClass(cssClass)\",
\"kotlinx.dom.hasClass\")\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\ninline fun Element.hasClass(cssClass: String): Boolean = this.newHasClass(cssClass)\n\n/**\n * Adds CSS class to element. Has no effect
if all specified classes are already in class attribute of the element\n *\n * @return true if at least one class has been added\n */\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \"This API is moved to another package, use 'kotlinx.dom.addClass' instead.\",\n replaceWith = ReplaceWith(\"this.addClass(cssClasses)\",
\"kotlinx.dom.addClass\")\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\ninline fun Element.addClass(vararg cssClasses: String): Boolean = this.newAddClass(*cssClasses)\n\n/**\n * Removes all [cssClasses] from element. Has no effect if all specified classes are missing in class attribute of the element\n *\n *

@return true if at least one class has been removed\n */\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \"This API is moved to another package, use 'kotlinx.dom.removeClass' instead.\",\n replaceWith = ReplaceWith(\"this.removeClass(cssClasses)\", \"kotlinx.dom.removeClass\")\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\ninline fun Element.removeClass(vararg cssClasses: String): Boolean = this.newRemoveClass(*cssClasses)","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.dom\n\nimport org.w3c.dom.Element\nimport org.w3c.dom.Node\nimport kotlin.internal.LowPriorityInOverloadResolution\nimport kotlinx.dom.isElement as newIsElement\nimport kotlinx.dom.isText as newIsText\n\n/**\n * Gets a value indicating whether this node is a TEXT_NODE or a CDATA_SECTION_NODE.\n */\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \"This API is moved to another package, use 'kotlinx.dom.isText' instead.\",\n replaceWith = ReplaceWith(\"this.isText\", \"kotlinx.dom.isText\")\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\npublic val Node.isText: Boolean\n inline get() = this.newIsText\n\n/**\n * Gets a value indicating whether this node is an [Element].\n */\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \"This API is moved to another package, use 'kotlinx.dom.isElement' instead.\",\n replaceWith = ReplaceWith(\"this.isElement\", \"kotlinx.dom.isElement\")\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\npublic val Node.isElement: Boolean\n inline get() = this.newIsElement\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage org.w3c.dom.events\n\npublic fun EventListener(handler: (Event) -> Unit): EventListener = EventListenerHandler(handler)\n\nprivate class EventListenerHandler(private val handler: (Event) -> Unit) : EventListener {\n public override fun handleEvent(event: Event) {\n handler(event)\n }\n\n public override fun toString(): String = \"EventListenerHandler($handler)\"\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage org.w3c.dom\n\npublic external interface ItemArrayLike<out T> {\n val length: Int\n fun item(index: Int): T?\n}\n\n/**\n * Returns the view of this `ItemArrayLike<T>` collection as `List<T>`\n */\npublic fun <T> ItemArrayLike<T>.asList(): List<T> = object : AbstractList<T>() {\n override val size: Int get() = this@asList.length\n\n override fun get(index: Int): T = when (index) {\n in 0..lastIndex -> this@asList.item(index).unsafeCast<T>()\n else -> throw IndexOutOfBoundsException(\"index $index is not in range [0..$lastIndex]\")\n }\n}","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.dom\n\nimport org.w3c.dom.Element\nimport org.w3c.dom.Node\nimport kotlin.internal.LowPriorityInOverloadResolution\nimport kotlinx.dom.appendText as newAppendText\nimport kotlinx.dom.clear as newClear\n\n/** Removes all the children from this node.\n */\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \"This API is moved to another package, use 'kotlinx.dom.clear' instead.\",\n replaceWith = ReplaceWith(\"this.clear()\", \"kotlinx.dom.clear\")\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\npublic inline fun Node.clear() = this.newClear()\n\n/**\n * Creates text node and append it to the element.\n *\n * @return this element\n */\n@LowPriorityInOverloadResolution\n@Deprecated(\n message = \"This API is moved to another package, use 'kotlinx.dom.appendText' instead.\",\n replaceWith = ReplaceWith(\"this.appendText(text)\", \"kotlinx.dom.appendText\")\n)\n@DeprecatedSinceKotlin(warningSince = \"1.4\", errorSince = \"1.6\")\ninline fun Element.appendText(text: String): Element = this.newAppendText(text)\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n/**\n * Reinterprets this value as a value of the [dynamic type](/docs/reference/dynamic-type.html).\n */\n@kotlin.internal.InlineOnly\npublic

inline fun Any?.asDynamic(): dynamic = this\n\n/**\n * Reinterprets this value as a value of the specified type [T] without any actual type checking.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <T> Any?.unsafeCast(): @kotlin.internal.NoInfer T = this.asDynamic()\n\n/**\n * Reinterprets this `dynamic` value as a value of the specified type [T] without any actual type checking.\n */\n@kotlin.internal.DynamicExtension\n@JsName(\"unsafeCastDynamic\")\n@kotlin.internal.InlineOnly\npublic inline fun <T> dynamic.unsafeCast(): @kotlin.internal.NoInfer T = this\n\n/**\n * Allows to iterate this `dynamic` object in the following cases:\n * - when it has an `iterator` function,\n * - when it is an array\n * - when it is an instance of [kotlin.collections.Iterable]\n */\n@kotlin.internal.DynamicExtension\npublic operator fun dynamic.iterator(): Iterator<dynamic> {\n val r: Any? = this\n\n return when {\n this[\"iterator\"] != null ->\n this[\"iterator\"]()\n isArrayish(r) ->\n r.unsafeCast<Array<*>>().iterator()\n\n else ->\n (r as Iterable<*>).iterator()\n }\n}\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.enums\n\n// Unused stub\ninternal actual class EnumEntriesSerializationProxy<E
: Enum<E>> actual constructor(entries: Array<E>)\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// a package is omitted to get declarations directly under the module\n\n@JsName(\"throwNPE\")\ninternal fun throwNPE(message: String) {\n throw NullPointerException(message)\n}\n\n@JsName(\"throwCCE\")\ninternal fun throwCCE() {\n throw ClassCastException(\"Illegal cast\")\n}\n\n@JsName(\"throwISE\")\ninternal fun throwISE(message: String) {\n throw IllegalStateException(message)\n}\n\n@JsName(\"throwUPAE\")\ninternal fun throwUPAE(propertyName: String) {\n throw UninitializedPropertyAccessException(\"lateinit property ${propertyName} has not been initialized\")\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed
by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin.collections\n\n/**\n * Groups elements from the [Grouping] source by key and counts elements in each group.\n *\n * @return a [Map] associating the key of each group with the count of elements in the group.\n *\n * @sample samples.collections.Grouping.groupingByEachCount\n */\n@SinceKotlin(\"1.1\")\npublic actual fun <T, K> Grouping<T, K>.eachCount(): Map<K, Int> =\n fold(0) { acc, _ -> acc + 1 }\n\n/*\n/**\n * Groups elements from the [Grouping] source by key and sums values provided by the [valueSelector] function for elements in each group.\n *\n * @return a [Map] associating the key of each group with the count of element in the group.\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <T, K> Grouping<T, K>.eachSumOf(valueSelector: (T) -> Int): Map<K, Int> =\n fold(0) { acc, e -> acc + valueSelector(e) }\n*/","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmName(\"GroupingKt\")\n@file:kotlin.jvm.JvmMultifileClass\n\npackage kotlin.collections\n\n/**\n * Represents a source of elements with a [keyOf] function, which can be applied to each element to get its key.\n *\n * A [Grouping] structure serves as an intermediate step in group-and-fold operations:\n * they group elements by their keys and then fold each group with some aggregating operation.\n *\n * It is created by attaching `keySelector: (T) -> K` function to a source of elements.\n * To get an instance of [Grouping] use one of `groupingBy` extension functions:\n * - [Iterable.groupingBy]\n * - [Sequence.groupingBy]\n * - [Array.groupingBy]\n * - [CharSequence.groupingBy]\n *\n * For the list of group-and-fold operations available, see the [extension functions](#extension-functions) for `Grouping`.\n */\n@SinceKotlin(\"1.1\")\npublic
interface Grouping<T, out K> {\n /** Returns an [Iterator] over the elements of the source of this grouping. */\n fun sourceIterator(): Iterator<T>\n /** Extracts the key of an [element]. */\n fun keyOf(element: T): K\n}\n\n/**\n * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially,\n * passing the previously accumulated value and the current element as arguments, and stores

the results in a new map.\n *\n * The key for each element is provided by the [Grouping.keyOf] function.\n *\n * @param operation function is invoked on each element with the following parameters:\n * - `key`: the key of the group this element belongs to;\n * - `accumulator`: the current value of the accumulator of the group, can be `null` if it's the first `element` encountered in the group;\n * - `element`: the element from the source being aggregated;\n * - `first`: indicates whether it's the first `element` encountered
in the group.\n *\n * @return a [Map] associating the key of each group with the result of aggregation of the group elements.\n * @sample samples.collections.Grouping.aggregateByRadix\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <T, K, R> Grouping<T, K>.aggregate(\n operation: (key: K, accumulator: R?, element: T, first: Boolean) -> R\n): Map<K, R> {\n return aggregateTo(mutableMapOf<K, R>(), operation)\n}\n\n/**\n * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially,\n * passing the previously accumulated value and the current element as arguments,\n * and stores the results in the given [destination] map.\n *\n * The key for each element is provided by the [Grouping.keyOf] function.\n *\n * @param operation a function that is invoked on each element with the following parameters:\n * - `key`: the key of the group this element belongs to;\n * - `accumulator`: the current value of the accumulator of the
group, can be `null` if it's the first `element` encountered in the group;\n * - `element`: the element from the source being aggregated;\n * - `first`: indicates whether it's the first `element` encountered in the group.\n *\n * If the [destination] map already has a value corresponding to some key,\n * then the elements being aggregated for that key are never considered as `first`.\n *\n * @return the [destination] map associating the key of each group with the result of aggregation of the group elements.\n * @sample samples.collections.Grouping.aggregateByRadixTo\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <T, K, R, M : MutableMap<in K, R>> Grouping<T, K>.aggregateTo(\n destination: M,\n operation: (key: K, accumulator: R?, element: T, first: Boolean) -> R\n): M {\n for (e in this.sourceIterator()) {\n val key = keyOf(e)\n val accumulator = destination[key]\n destination[key] = operation(key, accumulator, e, accumulator == null && !destination.containsKey(key))\n }\n return destination\n}\n\n/**\n * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially,\n * passing the previously accumulated value and the current element as arguments, and stores the results in a new map.\n * An initial value of accumulator is provided by [initialValueSelector] function.\n *\n * @param initialValueSelector a function that provides an initial value of accumulator for each group.\n * It's invoked with parameters:\n * - `key`: the key of the group;\n * - `element`: the first element being encountered in that group.\n *\n * @param operation a function that is invoked on each element with the following parameters:\n * - `key`: the key of the group this element belongs to;\n * - `accumulator`: the current value of the accumulator of the group;\n * - `element`: the element from the source being accumulated.\n *\n * @return a [Map] associating the key of each group with the result
of accumulating the group elements.\n * @sample samples.collections.Grouping.foldByEvenLengthWithComputedInitialValue\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <T, K, R> Grouping<T, K>.fold(\n initialValueSelector: (key: K, element: T) -> R,\n operation: (key: K, accumulator: R, element: T) -> R\n): Map<K, R> =\n @Suppress(\"UNCHECKED_CAST\")\n aggregate { key, acc, e, first -> operation(key, if (first) initialValueSelector(key, e) else acc as R, e) }\n\n/**\n * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially,\n * passing the previously accumulated value and the current element as arguments,\n * and stores the results in the given [destination] map.\n * An initial value of accumulator is provided by [initialValueSelector] function.\n *\n * @param initialValueSelector a function that provides an initial value of accumulator for each group.\n * It's invoked with parameters:\n * - `key`: the key of the
group;\n * - `element`: the first element being encountered in that group.\n *\n * If the [destination] map already has a value corresponding to some key, that value is used as an initial value of\n * the accumulator for that group and the [initialValueSelector] function is not called for that group.\n *\n * @param operation a function that is invoked on each element with the following parameters:\n * - `key`: the key of the group this element belongs to;\n * - `accumulator`: the current value of the accumulator of the group;\n * - `element`: the element from the source being accumulated.\n *\n * @return the [destination] map associating the key of each group with the result of

accumulating the group elements.\n * @sample samples.collections.Grouping.foldByEvenLengthWithComputedInitialValueTo\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <T, K, R, M : MutableMap<in K, R>> Grouping<T, K>.foldTo(\n destination: M,\n initialValueSelector: (key: K, element: T) -> R,\n operation: (key: K, accumulator: R, element: T) -> R\n): M =\n @Suppress(\"UNCHECKED_CAST\")\n aggregateTo(destination) { key, acc, e, first -> operation(key, if (first) initialValueSelector(key, e) else acc as R, e) }\n\n\n/**\n * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially,\n * passing the previously accumulated value and the current element as arguments, and stores the results in a new map.\n * An initial value of accumulator is the same [initialValue] for each group.\n *\n * @param operation a function that is invoked on each element with the following parameters:\n * - `accumulator`: the current value of the accumulator of the group;\n * - `element`: the element from the source being accumulated.\n *\n * @return a [Map] associating the key of each group with the result of accumulating the group elements.\n * @sample samples.collections.Grouping.foldByEvenLengthWithConstantInitialValue\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <T, K, R> Grouping<T, K>.fold(\n initialValue: R,\n operation: (accumulator: R, element: T) -> R\n): Map<K, R> =\n @Suppress(\"UNCHECKED_CAST\")\n aggregate { _, acc, e, first -> operation(if (first) initialValue else acc as R, e) }\n\n/**\n * Groups elements from the [Grouping] source by key and applies [operation] to the elements of each group sequentially,\n * passing the previously accumulated value and the current element as arguments,\n * and stores the results in the given [destination] map.\n * An initial value of accumulator is the same [initialValue] for each group.\n *\n * If the [destination] map already has a value corresponding to the key of some group,\n * that value is used as an initial value of the accumulator for that group.\n *\n * @param operation a function that is invoked on each element with the following parameters:\n * - `accumulator`: the current value of the accumulator of the group;\n * - `element`: the element from the source being accumulated.\n *\n * @return the [destination] map associating the key of each group with the result of accumulating the group elements.\n * @sample samples.collections.Grouping.foldByEvenLengthWithConstantInitialValueTo\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <T, K, R, M : MutableMap<in K, R>> Grouping<T, K>.foldTo(\n destination: M,\n initialValue: R,\n operation: (accumulator: R, element: T) -> R\n): M =\n @Suppress(\"UNCHECKED_CAST\")\n aggregateTo(destination) { _, acc, e, first -> operation(if (first) initialValue else acc as R, e) }\n\n\n/**\n * Groups elements from the [Grouping] source by key and applies the reducing [operation] to the elements of each group\n * sequentially starting from the second element of the group,\n * passing the previously accumulated value and the current element as arguments,\n * and stores the results in a new map.\n * An initial value of accumulator is the first element of the group.\n *\n * @param operation a function that is invoked on each subsequent element of the group with the following parameters:\n * - `key`: the key of the group this element belongs to;\n * - `accumulator`: the current value of the accumulator of the group;\n * - `element`: the element from the source being accumulated.\n *\n * @return a [Map] associating the key of each group with the result of accumulating the group elements.\n * @sample samples.collections.Grouping.reduceByMaxVowels\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <S, T : S, K> Grouping<T, K>.reduce(\n operation: (key: K, accumulator: S, element: T) -> S\n): Map<K, S> =\n aggregate { key, acc, e, first ->\n @Suppress(\"UNCHECKED_CAST\")\n if (first) e else operation(key, acc as S, e)\n }\n\n/**\n * Groups elements from the [Grouping] source by key and applies the reducing [operation] to the elements of each group\n * sequentially starting from the second element of the group,\n * passing the previously accumulated value and the current element as arguments,\n * and stores the results in the given [destination] map.\n * An initial value of accumulator is the first element of the group.\n *\n * If the [destination] map already has a value corresponding to the key of some group,\n * that value is used as an initial value of the accumulator for that group and the first element of that group is also\n * subjected to the [operation].\n\n * @param operation a function that is invoked on each subsequent element of the group with the following parameters:\n * - `accumulator`: the current value of the

```
accumulator of the group;\n *  - `element`: the element from the source being folded;\n *\n * @return the [destination] map associating the key of each group with the result of accumulating the group elements.\n * @sample samples.collections.Grouping.reduceByMaxVowelsTo\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <S, T : S, K, M : MutableMap<in K, S>> Grouping<T, K>.reduceTo(\n    destination: M,\n    operation: (key: K, accumulator: S, element: T) -> S\n): M =\n    aggregateTo(destination) { key, acc, e, first ->\n        @Suppress(\"UNCHECKED_CAST\")\n        if (first) e else operation(key, acc as S, e)\n    }\n\n\n/**\n * Groups elements from the [Grouping] source by key and counts elements in each group to the given [destination] map.\n *\n * If the [destination] map already has a value corresponding to the key of some group,\n * that value is used as an initial value of the counter for that group.\n *\n * @return the [destination] map associating the key of each group with the count of elements in the group.\n *\n * @sample samples.collections.Grouping.groupingByEachCount\n */\n@SinceKotlin(\"1.1\")\npublic fun <T, K, M : MutableMap<in K, Int>> Grouping<T, K>.eachCountTo(destination: M): M =\n    foldTo(destination, 0) { acc, _ -> acc + 1 }\n\n/*\n/**\n * Groups elements from the [Grouping] source by key and sums values provided by the [valueSelector] function for elements in each group\n * to the given [destination] map.\n *\n *\n * If the [destination] map already has a value corresponding to the key of some group,\n * that value is used as an initial value of the sum for that group.\n *\n * @return the [destination] map associating the key of each group with the sum of elements in the group.\n */\n@SinceKotlin(\"1.1\")\npublic inline fun <T, K, M : MutableMap<in K, Int>> Grouping<T, K>.eachSumOfTo(destination: M, valueSelector: (T) -> Int): M =\n        foldTo(destination, 0) { acc, e -> acc + valueSelector(e)}\n*/\n\n\n/*\n// TODO: sum by long and by double overloads\n\npublic inline fun <T, K, M : MutableMap<in K, Long>> Grouping<T, K>.sumEachByLongTo(destination: M, valueSelector: (T) -> Long): M =\n        foldTo(destination, 0L) { acc, e -> acc + valueSelector(e)}\n\npublic inline fun <T, K> Grouping<T, K>.sumEachByLong(valueSelector: (T) -> Long): Map<K, Long> =\n        fold(0L) { acc, e -> acc + valueSelector(e)}\n\npublic inline fun <T, K, M : MutableMap<in K, Double>> Grouping<T, K>.sumEachByDoubleTo(destination: M, valueSelector: (T) -> Double): M =\n    foldTo(destination, 0.0) { acc, e -> acc + valueSelector(e)}\n\npublic inline fun <T, K> Grouping<T, K>.sumEachByDouble(valueSelector: (T) -> Double): Map<K, Double> =\n        fold(0.0) { acc, e -> acc + valueSelector(e)}\n*/\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n@Retention(AnnotationRetention.BINARY)\n@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY)\ninternal annotation class JsPolyfill(val implementation: String)\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n/**\n * An interface for indexing access to a collection of key-value pairs, where type of key is [String] and type of value is [Any?][Any].\n */\npublic external interface Json {\n    /**\n     * Calls to the function will be translated to indexing operation (square brackets) on the receiver with [propertyName] as the argument.\n     *\n     * E.g. for next code:\n     * ```kotlin\n     * fun test(j: Json, p: String) = j[\"prop\"] + j.get(p)\n     * ```\n     *\n     * will be generated:\n     * ```js\n     * function test(j, p) {\n     *     return j[\"prop\"] + j[p];\n     * }\n     * ```\n     */\n    operator fun get(propertyName: String): Any?\n\n    /**\n     * Calls of the function will be translated to an assignment of [value] to the receiver indexed (with square brackets/index operation) with [propertyName].\n     *\n     * E.g. for the following code:\n     * ```kotlin\n     * fun test(j: Json, p: String, newValue: Any) {\n     *     j[\"prop\"] = 1\n     *     j.set(p, newValue)\n     * }\n
     * ```\n     *\n     * will be generated:\n     * ```js\n     * function test(j, p, newValue) {\n     *     j[\"prop\"] = 1;\n     *     j[p] = newValue;\n     * }\n     * }\n     * ```\n     */\n    operator fun set(propertyName: String, value: Any?): Unit\n}\n\n/**\n * Returns a simple JavaScript object (as [Json]) using provided key-value pairs as names and values of its properties.\n */\npublic fun json(vararg pairs: Pair<String, Any?>): Json {\n    val res: dynamic =
```

```
js(\"({})\")\n    for ((name, value) in pairs) {\n        res[name] = value\n    }\n    return res\n}\n\n/**\n * Adds key-value pairs from [other] to [this].\n * Returns the original receiver.\n */\npublic fun Json.add(other: Json): Json {\n val keys: Array<String> = js(\"Object\").keys(other)\n    for (key in keys) {\n        if (other.asDynamic().hasOwnProperty(key)) {\n            this[key] = other[key];\n        }\n    }\n    return this\n}\n\n/**\n * Exposes the JavaScript [JSON object](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON)
 to Kotlin.\n */\n@Suppress(\"NOT_DOCUMENTED\")\npublic external object JSON {\n    public fun stringify(o: Any?): String\n    public fun stringify(o: Any?, replacer: ((key: String, value: Any?) -> Any?)): String\n    public fun stringify(o: Any?, replacer: ((key: String, value: Any?) -> Any?)? = definedExternally, space: Int): String\n    public fun stringify(o: Any?, replacer: ((key: String, value: Any?) -> Any?)? = definedExternally, space: String): String\n    public fun stringify(o: Any?, replacer: Array<String>): String\n    public fun stringify(o: Any?, replacer: Array<String>, space: Int): String\n    public fun stringify(o: Any?, replacer: Array<String>, space: String): String\n\n    public fun <T> parse(text: String): T\n    public fun <T> parse(text: String, reviver: ((key: String, value: Any?) -> Any?)): T\n}\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use
 of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin.math\n\n\nimport kotlin.internal.InlineOnly\nimport kotlin.js.JsMath as nativeMath\n\n\n// region ================ Double Math ======================================\n\n/** Computes the sine of the angle [x] given in radians.\n *\n *  Special cases:\n *    - `sin(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun sin(x: Double): Double = nativeMath.sin(x)\n\n/** Computes the cosine of the angle [x] given in radians.\n *\n *  Special cases:\n *    - `cos(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun cos(x: Double): Double = nativeMath.cos(x)\n\n/** Computes the tangent of the angle [x] given in radians.\n *\n *  Special cases:\n *    - `tan(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun tan(x: Double): Double = nativeMath.tan(x)\n\n/**\n
 * Computes the arc sine of the value [x];\n * the returned value is an angle in the range from `-PI/2` to `PI/2` radians.\n *\n * Special cases:\n *    - `asin(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun asin(x: Double): Double = nativeMath.asin(x)\n\n/**\n * Computes the arc cosine of the value [x];\n * the returned value is an angle in the range from `0.0` to `PI` radians.\n *\n * Special cases:\n *    - `acos(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun acos(x: Double): Double = nativeMath.acos(x)\n\n/**\n * Computes the arc tangent of the value [x];\n * the returned value is an angle in the range from `-PI/2` to `PI/2` radians.\n *\n * Special cases:\n *   - `atan(NaN)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun atan(x: Double): Double = nativeMath.atan(x)\n\n/**\n * Returns the angle `theta` of the polar
 coordinates `(r, theta)` that correspond\n * to the rectangular coordinates `(x, y)` by computing the arc tangent of the value [y] / [x];\n * the returned value is an angle in the range from `-PI` to `PI` radians.\n *\n * Special cases:\n *  - `atan2(0.0, 0.0)` is `0.0`\n *   - `atan2(0.0, x)` is  `0.0` for `x > 0` and `PI` for `x < 0`\n *   - `atan2(-0.0, x)` is `-0.0` for 'x > 0` and `-PI` for `x < 0`\n *   - `atan2(y, +Inf)` is `0.0` for `0 < y < +Inf` and `-0.0` for '-Inf < y < 0`\n *   - `atan2(y, -Inf)` is `PI` for `0 < y < +Inf` and `-PI` for `-Inf < y < 0`\n *   - `atan2(y, 0.0)` is `PI/2` for `y > 0` and `-PI/2` for `y < 0`\n *   - `atan2(+Inf, x)` is `PI/2` for finite `x`y\n *   - `atan2(-Inf, x)` is `-PI/2` for finite `x`\n *   - `atan2(NaN, x)` and `atan2(y, NaN)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun atan2(y: Double, x: Double): Double = nativeMath.atan2(y, x)\n\n/**\n * Computes the hyperbolic sine of the value [x].\n *\n * Special
 cases:\n *   - `sinh(NaN)` is `NaN`\n *   - `sinh(+Inf)` is `+Inf`\n *   - `sinh(-Inf)` is `-Inf`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun sinh(x: Double): Double = nativeSinh(x)\n\n/**\n * Computes the hyperbolic cosine of the value [x].\n *\n * Special cases:\n *   - `cosh(NaN)` is `NaN`\n *   - `cosh(+Inf|-Inf)` is `+Inf`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun cosh(x: Double):
```

Double = nativeCosh(x)\n\n/**\n * Computes the hyperbolic tangent of the value [x].\n *\n * Special cases:\n * - `tanh(NaN)` is `NaN`\n * - `tanh(+Inf)` is `1.0`\n * - `tanh(-Inf)` is `-1.0`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun tanh(x: Double): Double = nativeTanh(x)\n\n/**\n * Computes the inverse hyperbolic sine of the value [x].\n *\n * The returned value is `y` such that `sinh(y) == x`.\n *\n * Special cases:\n * - `asinh(NaN)` is `NaN`\n * - `asinh(+Inf)` is `+Inf`\n * - `asinh(-Inf)` is `-Inf`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun asinh(x: Double): Double = nativeAsinh(x)\n\n/**\n * Computes the inverse hyperbolic cosine of the value [x].\n *\n * The returned value is positive `y` such that `cosh(y) == x`.\n *\n * Special cases:\n * - `acosh(NaN)` is `NaN`\n * - `acosh(x)` is `NaN` when `x < 1`\n * - `acosh(+Inf)` is `+Inf`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun acosh(x: Double): Double = nativeAcosh(x)\n\n/**\n * Computes the inverse hyperbolic tangent of the value [x].\n *\n * The returned value is `y` such that `tanh(y) == x`.\n *\n * Special cases:\n * - `tanh(NaN)` is `NaN`\n * - `tanh(x)` is `NaN` when `x > 1` or `x < -1`\n * - `tanh(1.0)` is `+Inf`\n * - `tanh(-1.0)` is `-Inf`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun atanh(x: Double): Double = nativeAtanh(x)\n\n/**\n * Computes `sqrt(x^2 + y^2)` without intermediate overflow or underflow.\n *\n * Special cases:\n * - returns `+Inf` if any of arguments is infinite\n * - returns `NaN` if any of arguments is `NaN` and the other is not infinite\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun hypot(x: Double, y: Double): Double = nativeHypot(x, y)\n\n/**\n * Computes the positive square root of the value [x].\n *\n * Special cases:\n * - `sqrt(x)` is `NaN` when `x < 0` or `x` is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun sqrt(x: Double): Double = nativeMath.sqrt(x)\n\n/**\n * Computes Euler's number `e` raised to the power of the value [x].\n *\n * Special cases:\n * - `exp(NaN)` is `NaN`\n * - `exp(+Inf)` is `+Inf`\n * - `exp(-Inf)` is `0.0`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun exp(x: Double): Double = nativeMath.exp(x)\n\n/**\n * Computes `exp(x) - 1`.\n *\n * This function can be implemented to produce more precise result for [x] near zero.\n *\n * Special cases:\n * - `expm1(NaN)` is `NaN`\n * - `expm1(+Inf)` is `+Inf`\n *
 - `expm1(-Inf)` is `-1.0`\n *\n * @see [exp] function.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun expm1(x: Double): Double = nativeExpm1(x)\n\n/**\n * Computes the logarithm of the value [x] to the given [base].\n *\n * Special cases:\n * - `log(x, b)` is `NaN` if either `x` or `b` are `NaN`\n * - `log(x, b)` is `NaN` when `x < 0` or `b <= 0` or `b == 1.0`\n * - `log(+Inf, +Inf)` is `NaN`\n * - `log(+Inf, b)` is `+Inf` for `b > 1` and `-Inf` for `b < 1`\n * - `log(0.0, b)` is `-Inf` for `b > 1` and `+Inf` for `b > 1`\n *\n * See also logarithm functions for common fixed bases: [ln], [log10] and [log2].\n */\n@SinceKotlin(\"1.2\")\npublic actual fun log(x: Double, base: Double): Double {\n   if (base <= 0.0 || base == 1.0) return Double.NaN\n   return nativeMath.log(x) / nativeMath.log(base)\n}\n\n/**\n * Computes the natural logarithm (base `E`) of the value [x].\n *\n * Special cases:\n * - `ln(NaN)` is `NaN`\n * - `ln(x)` is `NaN` when `x < 0.0`\n * - `ln(+Inf)` is `+Inf`\n * - `ln(0.0)` is `-Inf`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun ln(x: Double): Double = nativeMath.log(x)\n\n/**\n * Computes the common logarithm (base 10) of the value [x].\n *\n * @see [ln] function for special cases.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun log10(x: Double): Double = nativeLog10(x)\n\n/**\n * Computes the binary logarithm (base 2) of the value [x].\n *\n * @see [ln] function for special cases.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun log2(x: Double): Double = nativeLog2(x)\n\n/**\n * Computes `ln(x + 1)`.\n *\n * This function can be implemented to produce more precise result for [x] near zero.\n *\n * Special cases:\n * - `ln1p(NaN)` is `NaN`\n * - `ln1p(x)` is `NaN` where `x < -1.0`\n * - `ln1p(-1.0)` is `-Inf`\n * - `ln1p(+Inf)` is `+Inf`\n *\n * @see [ln] function\n * @see [expm1] function\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun ln1p(x: Double): Double = nativeLog1p(x)\n\n/**\n * Rounds the given value [x] to an integer towards positive infinity.\n\n * @return the smallest double value that is greater than or equal to the given value [x] and is a mathematical integer.\n *\n * Special cases:\n * - `ceil(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun ceil(x: Double): Double

```
= nativeMath.ceil(x)\n\n/**\n * Rounds the given value [x] to an integer towards negative infinity.\n\n * @return the largest double value that is smaller than or equal to the given value [x] and is a mathematical integer.\n *\n * Special cases:\n *   - `floor(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun floor(x: Double): Double = nativeMath.floor(x)\n\n/**\n * Rounds the given value [x] to an integer towards zero.\n *\n * @return the value [x] having its fractional part truncated.\n *\n * Special cases:\n *   - `truncate(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun truncate(x: Double): Double = nativeTrunc(x)\n\n/**\n * Rounds the given value [x] towards the closest integer with ties rounded towards even integer.\n *\n * Special cases:\n *   - `round(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n */\n@SinceKotlin(\"1.2\")\npublic actual fun round(x: Double): Double {\n    if (x % 0.5 != 0.0) {\n        return nativeMath.round(x)\n    }\n    val floor = floor(x)\n    return if (floor % 2 == 0.0) floor else ceil(x)\n}\n\n/**\n * Returns the absolute value of the given value [x].\n *\n * Special cases:\n *   - `abs(NaN)` is `NaN`\n *\n * @see absoluteValue extension property for [Double]\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun abs(x: Double): Double = nativeMath.abs(x)\n\n/**\n * Returns the sign of the given value [x]:\n *   - `-1.0` if the value is negative,\n *   - zero if the value is zero,\n *   - `1.0` if the value is positive\n *\n * Special case:\n *   - `sign(NaN)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun sign(x: Double): Double = nativeSign(x)\n\n\n\n/**\n * Returns the smaller of two values.\n *\n * If either value is `NaN`, then the result is `NaN`.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun min(a: Double, b: Double): Double = nativeMath.min(a, b)\n\n/**\n * Returns the greater of two values.\n *\n * If either value is `NaN`, then the result is `NaN`.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun max(a: Double, b: Double): Double = nativeMath.max(a, b)\n\n\n\n/**\n * Returns the cube root of [x]. For any `x`, `cbrt(-x) == -cbrt(x)`;\n * that is, the cube root of a negative value is the negative of the cube root\n * of that value's magnitude. Special cases:\n *\n * Special cases:\n *   - If the argument is `NaN`, then the result is `NaN`.\n *   - If the argument is infinite, then the result is an infinity with the same sign as the argument.\n *   - If the argument is zero, then the result is a zero with the same sign as the argument.\n */\n@SinceKotlin(\"1.8\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@InlineOnly\npublic actual inline fun cbrt(x: Double): Double = nativeMath.cbrt(x)\n\n\n\n// extensions\n\n/**\n * Raises this value to the power [x].\n *\n * Special cases:\n *   - `b.pow(0.0)` is `1.0`\n *   - `b.pow(1.0) == b`\n *   - `b.pow(NaN)` is `NaN`\n *   - `NaN.pow(x)` is `NaN` for `x != 0.0`\n *   - `b.pow(Inf)` is `NaN` for `abs(b) == 1.0`\n *   - `b.pow(x)` is `NaN` for `b < 0` and `x` is finite and not an integer\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun Double.pow(x: Double): Double = nativeMath.pow(this, x)\n\n/**\n * Raises this value to the integer power [n].\n *\n * See the other overload of [pow] for details.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun Double.pow(n: Int): Double = nativeMath.pow(this, n.toDouble())\n\n/**\n * Returns the absolute value of this value.\n *\n * Special cases:\n *   - `NaN.absoluteValue` is `NaN`\n *\n * @see abs function\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline val Double.absoluteValue: Double get() = nativeMath.abs(this)\n\n/**\n * Returns the sign of this value:\n *   - `-1.0` if the value is negative,\n *   - zero if the value is zero,\n *   - `1.0` if the value is positive\n *\n * Special case:\n *   - `NaN.sign` is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline val Double.sign: Double get() = nativeSign(this)\n\n\n\n/**\n * Returns this value with the sign bit same as of the [sign] value.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun Double.withSign(sign: Int): Double = this.withSign(sign.toDouble())\n\n\n\n/**\n * Returns the ulp (unit in the last place) of this value.\n *\n * An ulp is a positive distance between this value and the next nearest [Double] value larger in magnitude.\n *\n * Special Cases:\n *   - `NaN.ulp` is `NaN`\n *   - `x.ulp` is `+Inf` when `x` is `+Inf` or `-Inf`\n *   - `0.0.ulp` is `Double.MIN_VALUE`\n */\n@SinceKotlin(\"1.2\")\npublic actual val Double.ulp: Double get() = when {\n    this < 0 -> (-this).ulp\n    this.isNaN() || this == Double.POSITIVE_INFINITY -> this\n    this == Double.MAX_VALUE -> this - this.nextDown()\n    else ->
```

this.nextUp() - this\n}\n\n/**\n * Returns the [Double] value nearest to this value in direction of positive infinity.\n */\n@SinceKotlin(\"1.2\")\npublic actual fun Double.nextUp(): Double = when {\n   this.isNaN() || this == Double.POSITIVE_INFINITY -> this\n   this == 0.0 -> Double.MIN_VALUE\n   else -> Double.fromBits(this.toRawBits() + if (this > 0) 1 else -1)\n}\n\n/**\n * Returns the [Double] value nearest to this value in direction of negative
 infinity.\n */\n@SinceKotlin(\"1.2\")\npublic actual fun Double.nextDown(): Double = when {\n   this.isNaN() || this == Double.NEGATIVE_INFINITY -> this\n   this == 0.0 -> -Double.MIN_VALUE\n   else -> Double.fromBits(this.toRawBits() + if (this > 0) -1 else 1)\n}\n\n\n/**\n * Returns the [Double] value nearest to this value in direction from this value towards the value [to].\n *\n * Special cases:\n *   - `x.nextTowards(y)` is `NaN` if either `x` or `y` are `NaN`\n *   - `x.nextTowards(x) == x`\n *\n */\n@SinceKotlin(\"1.2\")\npublic actual fun Double.nextTowards(to: Double): Double = when {\n   this.isNaN() || to.isNaN() -> Double.NaN\n   to == this -> to\n   to > this -> this.nextUp()\n   else /* to < this */ -> this.nextDown()\n}\n\n\n/**\n * Rounds this [Double] value to the nearest integer and converts the result to [Int].\n * Ties are rounded towards positive infinity.\n *\n * Special cases:\n *   - `x.roundToInt() == Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n *
 - `x.roundToInt() == Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n *\n * @throws IllegalArgumentException when this value is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic actual fun Double.roundToInt(): Int = when {\n   isNaN() -> throw IllegalArgumentException(\"Cannot round NaN value.\")\n   this > Int.MAX_VALUE -> Int.MAX_VALUE\n   this < Int.MIN_VALUE -> Int.MIN_VALUE\n   else -> nativeMath.round(this).toInt()\n}\n\n/**\n * Rounds this [Double] value to the nearest integer and converts the result to [Long].\n * Ties are rounded towards positive infinity.\n *\n * Special cases:\n *   - `x.roundToLong() == Long.MAX_VALUE` when `x > Long.MAX_VALUE`\n *   - `x.roundToLong() == Long.MIN_VALUE` when `x < Long.MIN_VALUE`\n *\n * @throws IllegalArgumentException when this value is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic actual fun Double.roundToLong(): Long = when {\n   isNaN() -> throw IllegalArgumentException(\"Cannot round NaN value.\")\n   this > Long.MAX_VALUE -> Long.MAX_VALUE\n   this
 < Long.MIN_VALUE -> Long.MIN_VALUE\n   else -> nativeMath.round(this).toLong()\n}\n\n// endregion\n\n\n\n// region ================ Float Math ==\n\n/** Computes the sine of the angle [x] given in radians.\n *\n * Special cases:\n *   - `sin(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun sin(x: Float): Float = nativeMath.sin(x.toDouble()).toFloat()\n\n/** Computes the cosine of the angle [x] given in radians.\n *\n * Special cases:\n *   - `cos(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun cos(x: Float): Float = nativeMath.cos(x.toDouble()).toFloat()\n\n/** Computes the tangent of the angle [x] given in radians.\n *\n * Special cases:\n *   - `tan(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun tan(x: Float): Float = nativeMath.tan(x.toDouble()).toFloat()\n\n/**\n * Computes the arc sine of the value [x];\n
 * the returned value is an angle in the range from `-PI/2` to `PI/2` radians.\n *\n * Special cases:\n *   - `asin(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun asin(x: Float): Float = nativeMath.asin(x.toDouble()).toFloat()\n\n/**\n * Computes the arc cosine of the value [x];\n * the returned value is an angle in the range from `0.0` to `PI` radians.\n *\n * Special cases:\n *   - `acos(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun acos(x: Float): Float = nativeMath.acos(x.toDouble()).toFloat()\n\n/**\n * Computes the arc tangent of the value [x];\n * the returned value is an angle in the range from `-PI/2` to `PI/2` radians.\n *\n * Special cases:\n *   - `atan(NaN)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun atan(x: Float): Float = nativeMath.atan(x.toDouble()).toFloat()\n\n/**\n * Returns the angle `theta` of the polar
 coordinates `(r, theta)` that correspond\n * to the rectangular coordinates `(x, y)` by computing the arc tangent of the value [y] / [x];\n * the returned value is an angle in the range from `-PI` to `PI` radians.\n *\n * Special cases:\n *
 - `atan2(0.0, 0.0)` is `0.0`\n *   - `atan2(0.0, x)` is `0.0` for `x > 0` and `PI` for `x < 0`\n *   - `atan2(-0.0, x)` is `-

0.0` for 'x > 0` and `-PI` for `x < 0`\n * - `atan2(y, +Inf)` is `0.0` for `0 < y < +Inf` and `-0.0` for '-Inf < y < 0`\n *  - `atan2(y, -Inf)` is `PI` for `0 < y < +Inf` and `-PI` for `-Inf < y < 0`\n *  - `atan2(y, 0.0)` is `PI/2` for `y > 0` and `-PI/2` for `y < 0`\n *  - `atan2(+Inf, x)` is `PI/2` for finite `x`y\n *  - `atan2(-Inf, x)` is `-PI/2` for finite `x`\n *  - `atan2(NaN, x)` and `atan2(y, NaN)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun atan2(y: Float, x: Float): Float = nativeMath.atan2(y.toDouble(), x.toDouble()).toFloat()\n\n/**\n * Computes the hyperbolic sine of
the value [x].\n *\n * Special cases:\n * - `sinh(NaN)` is `NaN`\n * - `sinh(+Inf)` is `+Inf`\n * - `sinh(-Inf)` is `-Inf`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun sinh(x: Float): Float = nativeSinh(x.toDouble()).toFloat()\n\n/**\n * Computes the hyperbolic cosine of the value [x].\n *\n * Special cases:\n * - `cosh(NaN)` is `NaN`\n * - `cosh(+Inf|-Inf)` is `+Inf`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun cosh(x: Float): Float = nativeCosh(x.toDouble()).toFloat()\n\n/**\n * Computes the hyperbolic tangent of the value [x].\n *\n * Special cases:\n * - `tanh(NaN)` is `NaN`\n * - `tanh(+Inf)` is `1.0`\n * - `tanh(-Inf)` is `-1.0`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun tanh(x: Float): Float = nativeTanh(x.toDouble()).toFloat()\n\n/**\n * Computes the inverse hyperbolic sine of the value [x].\n *\n * The returned value is `y` such that `sinh(y) == x`.\n *\n * Special cases:\n * - `asinh(NaN)`
is `NaN`\n * - `asinh(+Inf)` is `+Inf`\n * - `asinh(-Inf)` is `-Inf`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun asinh(x: Float): Float = nativeAsinh(x.toDouble()).toFloat()\n\n/**\n * Computes the inverse hyperbolic cosine of the value [x].\n *\n * The returned value is positive `y` such that `cosh(y) == x`.\n *\n * Special cases:\n * - `acosh(NaN)` is `NaN`\n * - `acosh(x)` is `NaN` when `x < 1`\n * - `acosh(+Inf)` is `+Inf`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun acosh(x: Float): Float = nativeAcosh(x.toDouble()).toFloat()\n\n/**\n * Computes the inverse hyperbolic tangent of the value [x].\n *\n * The returned value is `y` such that `tanh(y) == x`.\n *\n * Special cases:\n * - `tanh(NaN)` is `NaN`\n * - `tanh(x)` is `NaN` when `x > 1` or `x < -1`\n * - `tanh(1.0)` is `+Inf`\n * - `tanh(-1.0)` is `-Inf`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun atanh(x: Float): Float = nativeAtanh(x.toDouble()).toFloat()\n\n/**\n
 * Computes `sqrt(x^2 + y^2)` without intermediate overflow or underflow.\n *\n * Special cases:\n * - returns `+Inf` if any of arguments is infinite\n * - returns `NaN` if any of arguments is `NaN` and the other is not infinite\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun hypot(x: Float, y: Float): Float = nativeHypot(x.toDouble(), y.toDouble()).toFloat()\n\n/**\n * Computes the positive square root of the value [x].\n *\n * Special cases:\n * - `sqrt(x)` is `NaN` when `x < 0` or `x` is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun sqrt(x: Float): Float = nativeMath.sqrt(x.toDouble()).toFloat()\n\n/**\n * Computes Euler's number `e` raised to the power of the value [x].\n *\n * Special cases:\n * - `exp(NaN)` is `NaN`\n * - `exp(+Inf)` is `+Inf`\n * - `exp(-Inf)` is `0.0`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun exp(x: Float): Float = nativeMath.exp(x.toDouble()).toFloat()\n\n/**\n
 * Computes `exp(x) - 1`.\n *\n * This function can be implemented to produce more precise result for [x] near zero.\n *\n * Special cases:\n * - `expm1(NaN)` is `NaN`\n * - `expm1(+Inf)` is `+Inf`\n * - `expm1(-Inf)` is `-1.0`\n *\n * @see [exp] function.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun expm1(x: Float): Float = nativeExpm1(x.toDouble()).toFloat()\n\n/**\n * Computes the logarithm of the value [x] to the given [base].\n *\n * Special cases:\n * - `log(x, b)` is `NaN` if either `x` or `b` are `NaN`\n * - `log(x, b)` is `NaN` when `x < 0` or `b <= 0` or `b == 1.0`\n * - `log(+Inf, +Inf)` is `NaN`\n * - `log(+Inf, b)` is `+Inf` for `b > 1` and `-Inf` for `b < 1`\n * - `log(0.0, b)` is `-Inf` for `b > 1` and `+Inf` for `b > 1`\n *\n * See also logarithm functions for common fixed bases: [ln], [log10] and [log2].\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun log(x: Float, base: Float): Float = log(x.toDouble(), base.toDouble()).toFloat()\n\n/**\n
 * Computes the natural logarithm (base `E`) of the value [x].\n *\n * Special cases:\n * - `ln(NaN)` is `NaN`\n * - `ln(x)` is `NaN` when `x < 0.0`\n * - `ln(+Inf)` is `+Inf`\n * - `ln(0.0)` is `-Inf`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun ln(x: Float): Float =

nativeMath.log(x.toDouble()).toFloat()\n\n/**\n * Computes the common logarithm (base 10) of the value [x].\n *\n * @see [ln] function for special cases.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun log10(x: Float): Float = nativeLog10(x.toDouble()).toFloat()\n\n/**\n * Computes the binary logarithm (base 2) of the value [x].\n *\n * @see [ln] function for special cases.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun log2(x: Float): Float = nativeLog2(x.toDouble()).toFloat()\n\n/**\n * Computes `ln(x + 1)`.\n *\n * This function can be implemented to produce more precise result for [x] near zero.\n *\n * Special cases:\n * - `ln1p(NaN)` is `NaN`\n * - `ln1p(x)` is `NaN` where `x < -1.0`\n * - `ln1p(-1.0)` is `-Inf`\n * - `ln1p(+Inf)` is `+Inf`\n *\n * @see [ln] function\n * @see [expm1] function\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun ln1p(x: Float): Float = nativeLog1p(x.toDouble()).toFloat()\n\n/**\n * Rounds the given value [x] to an integer towards positive infinity.\n\n * @return the smallest Float value that is greater than or equal to the given value [x] and is a mathematical integer.\n *\n * Special cases:\n * - `ceil(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun ceil(x: Float): Float = nativeMath.ceil(x.toDouble()).toFloat()\n\n/**\n * Rounds the given value [x] to an integer towards negative infinity.\n\n * @return the largest Float value that is smaller than or equal to the given value [x] and is a mathematical integer.\n *\n * Special cases:\n * - `floor(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun floor(x: Float): Float = nativeMath.floor(x.toDouble()).toFloat()\n\n/**\n * Rounds the given value [x] to an integer towards zero.\n *\n * @return the value [x] having its fractional part truncated.\n *\n * Special cases:\n * - `truncate(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun truncate(x: Float): Float = truncate(x.toDouble()).toFloat()\n\n/**\n * Rounds the given value [x] towards the closest integer with ties rounded towards even integer.\n *\n * Special cases:\n * - `round(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun round(x: Float): Float = round(x.toDouble()).toFloat()\n\n\n/**\n * Returns the absolute value of the given value [x].\n *\n * Special cases:\n * - `abs(NaN)` is `NaN`\n *\n * @see absoluteValue extension property for [Float]\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun abs(x: Float): Float = nativeMath.abs(x.toDouble()).toFloat()\n\n/**\n * Returns the sign of the given value [x]:\n * - `-1.0` if the value is negative,\n * - zero if the value is zero,\n * - `1.0` if the value is positive\n *\n * Special case:\n * - `sign(NaN)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun sign(x: Float): Float = nativeSign(x.toDouble()).toFloat()\n\n\n\n/**\n * Returns the smaller of two values.\n *\n * If either value is `NaN`, then the result is `NaN`.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun min(a: Float, b: Float): Float = nativeMath.min(a, b)\n\n/**\n * Returns the greater of two values.\n *\n * If either value is `NaN`, then the result is `NaN`.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun max(a: Float, b: Float): Float = nativeMath.max(a, b)\n\n\n\n/**\n * Returns the cube root of [x]. For any `x`, `cbrt(-x) == -cbrt(x)`;\n * that is, the cube root of a negative value is the negative of the cube root\n * of that value's magnitude. Special cases:\n *\n * Special cases:\n * - If the argument is `NaN`, then the result is `NaN`.\n * - If the argument is infinite, then the result is an infinity with the same sign as the argument.\n * - If the argument is zero, then the result is a zero with the same sign as the argument.\n */\n@SinceKotlin(\"1.8\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@InlineOnly\npublic actual inline fun cbrt(x: Float): Float = nativeMath.cbrt(x.toDouble()).toFloat()\n\n\n\n// extensions\n\n\n/**\n * Raises this value to the power [x].\n *\n * Special cases:\n * - `b.pow(0.0)` is `1.0`\n * - `b.pow(1.0) == b`\n * - `b.pow(NaN)` is `NaN`\n * - `NaN.pow(x)` is `NaN` for `x != 0.0`\n * - `b.pow(Inf)` is `NaN` for `abs(b) == 1.0`\n * - `b.pow(x)` is `NaN` for `b < 0` and `x` is finite and not an integer\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun Float.pow(x: Float): Float = nativeMath.pow(this.toDouble(), x.toDouble()).toFloat()\n\n/**\n * Raises this value to the integer power [n].\n *\n * See the other overload of [pow] for details.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun Float.pow(n: Int): Float = nativeMath.pow(this.toDouble(), n.toDouble()).toFloat()\n\n/**\n * Returns the absolute

value of this value.\n *\n * Special cases:\n * - `NaN.absoluteValue` is `NaN`\n *\n * @see abs function\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline val Float.absoluteValue: Float get() = nativeMath.abs(this.toDouble()).toFloat()\n\n/**\n * Returns the sign of this value:\n * - `-1.0` if the value is negative,\n * - zero if the value is zero,\n * - `1.0` if the value is positive\n *\n * Special case:\n * - `NaN.sign` is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline val Float.sign: Float get() = nativeSign(this.toDouble()).toFloat()\n\n/**\n * Returns this value with the sign bit same as of the [sign] value.\n *\n * If [sign] is `NaN` the sign of the result is undefined.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun Float.withSign(sign: Float): Float = this.toDouble().withSign(sign.toDouble()).toFloat()\n\n/**\n * Returns this value with the sign bit same as of the [sign] value.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun Float.withSign(sign: Int): Float = this.toDouble().withSign(sign.toDouble()).toFloat()\n\n\n/**\n * Rounds this [Float] value to the nearest integer and converts the result to [Int].\n * Ties are rounded towards positive infinity.\n *\n * Special cases:\n * - `x.roundToInt() == Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `x.roundToInt() == Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n *\n * @throws IllegalArgumentException when this value is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun Float.roundToInt(): Int = toDouble().roundToInt()\n\n/**\n * Rounds this [Float] value to the nearest integer and converts the result to [Long].\n * Ties are rounded towards positive infinity.\n *\n * Special cases:\n * - `x.roundToLong() == Long.MAX_VALUE` when `x > Long.MAX_VALUE`\n * - `x.roundToLong() == Long.MIN_VALUE` when `x < Long.MIN_VALUE`\n *\n * @throws IllegalArgumentException when this value is `NaN`\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun Float.roundToLong(): Long = toDouble().roundToLong()\n\n\n// endregion\n\n// region ================ Integer Math ==\n\n\n/**\n * Returns the absolute value of the given value [n].\n *\n * Special cases:\n * - `abs(Int.MIN_VALUE)` is `Int.MIN_VALUE` due to an overflow\n *\n * @see absoluteValue extension property for [Int]\n */\n// TODO: remove manual 'or' when KT-19290 is fixed\n@SinceKotlin(\"1.2\")\npublic actual fun abs(n: Int): Int = if (n < 0) (-n or 0) else n\n\n/**\n * Returns the smaller of two values.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun min(a: Int, b: Int): Int = nativeMath.min(a, b)\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline fun max(a: Int, b: Int): Int = nativeMath.max(a, b)\n\n/**\n * Returns the absolute value of this value.\n *\n * Special cases:\n * - `Int.MIN_VALUE.absoluteValue` is `Int.MIN_VALUE` due to an overflow\n *\n * @see abs function\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline val Int.absoluteValue: Int get() = abs(this)\n\n/**\n * Returns the sign of this value:\n * - `-1` if the value is negative,\n * - `0` if the value is zero,\n * - `1` if the value is positive\n */\n@SinceKotlin(\"1.2\")\npublic actual val Int.sign: Int get() = when {\n   this < 0 -> -1\n   this > 0 -> 1\n   else -> 0\n}\n\n\n\n/**\n * Returns the absolute value of the given value [n].\n *\n * Special cases:\n * - `abs(Long.MIN_VALUE)` is `Long.MIN_VALUE` due to an overflow\n *\n * @see absoluteValue extension property for [Long]\n */\n@SinceKotlin(\"1.2\")\npublic actual fun abs(n: Long): Long = if (n < 0) -n else n\n\n/**\n * Returns the smaller of two values.\n */\n@SinceKotlin(\"1.2\")\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline fun min(a: Long, b: Long): Long = if (a <= b) a else b\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin(\"1.2\")\n@Suppress(\"NOTHING_TO_INLINE\")\npublic actual inline fun max(a: Long, b: Long): Long = if (a >= b) a else b\n\n/**\n * Returns the absolute value of this value.\n *\n * Special cases:\n * - `Long.MIN_VALUE.absoluteValue` is `Long.MIN_VALUE` due to an overflow\n *\n * @see abs function\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\npublic actual inline val Long.absoluteValue: Long get() = abs(this)\n\n/**\n * Returns the sign of this value:\n * - `-1` if the value is negative,\n * - `0` if the value is zero,\n * - `1` if the value is positive\n */\n@SinceKotlin(\"1.2\")\npublic actual val Long.sign: Int get() = when {\n   this < 0 -> -1\n   this > 0 -> 1\n   else -> 0\n}\n\n\n\n// endregion\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the

license/LICENSE.txt file.\n */\n\npackage kotlin\n\n/**\n * Returns `true` if the specified number is a\n * Not-a-Number (NaN) value, `false` otherwise.\n */\npublic actual fun Double.isNaN(): Boolean = this != this\n\n/**\n * Returns `true` if the specified number is a\n * Not-a-Number (NaN) value, `false` otherwise.\n */\npublic actual fun Float.isNaN(): Boolean = this != this\n\n/**\n * Returns `true` if this value is infinitely large in magnitude.\n */\npublic actual fun Double.isInfinite(): Boolean = this == Double.POSITIVE_INFINITY || this == Double.NEGATIVE_INFINITY\n\n/**\n
* Returns `true` if this value is infinitely large in magnitude.\n */\npublic actual fun Float.isInfinite(): Boolean = this == Float.POSITIVE_INFINITY || this == Float.NEGATIVE_INFINITY\n\n/**\n * Returns `true` if the argument is a finite floating-point value; returns `false` otherwise (for `NaN` and infinity arguments).\n */\npublic actual fun Double.isFinite(): Boolean = !isInfinite() && !isNaN()\n\n/**\n * Returns `true` if the argument is a finite floating-point value; returns `false` otherwise (for `NaN` and infinity arguments).\n */\npublic actual fun Float.isFinite(): Boolean = !isInfinite() && !isNaN()\n\n\n/**\n * Counts the number of set bits in the binary representation of this [Int] number.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Int.countOneBits(): Int {\n   // Hacker's Delight 5-1 algorithm\n   var v = this\n   v = (v and 0x55555555) + (v.ushr(1) and 0x55555555)\n   v = (v and 0x33333333) + (v.ushr(2)
and 0x33333333)\n   v = (v and 0x0F0F0F0F) + (v.ushr(4) and 0x0F0F0F0F)\n   v = (v and 0x00FF00FF) + (v.ushr(8) and 0x00FF00FF)\n   v = (v and 0x0000FFFF) + (v.ushr(16))\n   return v\n}\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Int] number.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic actual inline fun Int.countLeadingZeroBits(): Int = nativeClz32(this)\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Int] number.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Int.countTrailingZeroBits(): Int =\n   // Hacker's Delight 5-4 algorithm for expressing countTrailingZeroBits with countLeadingZeroBits\n   Int.SIZE_BITS - (this or -this).inv().countLeadingZeroBits()\n\n/**\n * Returns a number having a single bit set in the position
of the most significant set bit of this [Int] number,\n * or zero, if this number is zero.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Int.takeHighestOneBit(): Int =\n   if (this == 0) 0 else 1.shl(Int.SIZE_BITS - 1 - countLeadingZeroBits())\n\n/**\n * Returns a number having a single bit set in the position of the least significant set bit of this [Int] number,\n * or zero, if this number is zero.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Int.takeLowestOneBit(): Int =\n   // Hacker's Delight 2-1 algorithm for isolating rightmost 1-bit\n   this and -this\n\n/**\n * Rotates the binary representation of this [Int] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n *\n * Rotating the number left by a negative bit count is the same as rotating it
right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n *\n * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 32)`\n
*/\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Int.rotateLeft(bitCount: Int): Int =\n   shl(bitCount) or ushr(Int.SIZE_BITS - bitCount)\n\n\n/**\n * Rotates the binary representation of this [Int] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n *\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n *\n * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 32)`\n
*/\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic

actual fun Int.rotateRight(bitCount: Int): Int =\n   shl(Int.SIZE_BITS - bitCount) or ushr(bitCount)\n\n\n/**\n * Counts the number of set bits in the binary representation of this [Long] number.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Long.countOneBits(): Int =\n   high.countOneBits() + low.countOneBits()\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Long] number.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Long.countLeadingZeroBits(): Int =\n   when (val high = this.high) {\n   0 -> Int.SIZE_BITS + low.countLeadingZeroBits()\n   else -> high.countLeadingZeroBits()\n   }\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Long] number.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Long.countTrailingZeroBits(): Int =\n   when (val low = this.low) {\n   0 -> Int.SIZE_BITS + high.countTrailingZeroBits()\n   else -> low.countTrailingZeroBits()\n   }\n\n/**\n * Returns a number having a single bit set in the position of the most significant set bit of this [Long] number,\n * or zero, if this number is zero.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Long.takeHighestOneBit(): Long =\n   when (val high = this.high) {\n   0 -> Long(low.takeHighestOneBit(), 0)\n   else -> Long(0, high.takeHighestOneBit())\n   }\n\n/**\n * Returns a number having a single bit set in the position of the least significant set bit of this [Long] number,\n * or zero, if this number is zero.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Long.takeLowestOneBit(): Long =\n   when (val low = this.low)\n   {\n   0 -> Long(0, high.takeLowestOneBit())\n   else -> Long(low.takeLowestOneBit(), 0)\n   }\n\n/**\n * Rotates the binary representation of this [Long] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n *\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n *\n * Rotating by a multiple of [Long.SIZE_BITS] (64) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 64)`\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual fun Long.rotateLeft(bitCount: Int): Long {\n   if ((bitCount and 31) != 0) {\n   val low = this.low\n   val high = this.high\n   val newLow = low.shl(bitCount) or high.ushr(-bitCount)\n   val newHigh = high.shl(bitCount) or low.ushr(-bitCount)\n   return if ((bitCount and 32) == 0) Long(newLow, newHigh) else Long(newHigh, newLow)\n   } else {\n   return if ((bitCount and 32) == 0) this else Long(high, low)\n   }\n}\n\n\n/**\n * Rotates the binary representation of this [Long] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n *\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n *\n * Rotating by a multiple of [Long.SIZE_BITS] (64) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 64)`\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic actual inline fun Long.rotateRight(bitCount: Int): Long = rotateLeft(-bitCount)\n","/*\n *
Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\nimport kotlin.internal.LowPriorityInOverloadResolution\n\n/**\n * Exposes the JavaScript [Promise object](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Promise) to Kotlin.\n */\n@Suppress(\"NOT_DOCUMENTED\")\npublic open external class Promise<out T>(executor: (resolve: (T) -> Unit, reject: (Throwable) -> Unit) -> Unit) {\n   @LowPriorityInOverloadResolution\n   public open fun <S> then(onFulfilled: ((T) -> S)?): Promise<S>\n\n   @LowPriorityInOverloadResolution\n   public open fun <S> then(onFulfilled: ((T) -> S)?, onRejected: ((Throwable) -> S)?): Promise<S>\n\n   public open fun <S> catch(onRejected: (Throwable) -> S): Promise<S>\n\n   public open fun finally(onFinally: () -> Unit): Promise<T>\n\n   companion object {\n

```
public fun <S> all(promise: Array<out Promise<S>>): Promise<Array<out S>>\n\n        public fun <S> race(promise: Array<out Promise<S>>): Promise<S>\n\n        public fun reject(e: Throwable): Promise<Nothing>\n\n        public fun <S> resolve(e: S): Promise<S>\n        public fun <S> resolve(e: Promise<S>): Promise<S>\n    }\n}\n\n// It's workaround for KT-19672 since we can fix it properly until KT-11265 isn't fixed.\ninline fun <T, S> Promise<Promise<T>>.then(\n    noinline onFulfilled: ((T) -> S)?\n): Promise<S> {\n  return this.unsafeCast<Promise<T>>().then(onFulfilled)\n}\n\ninline fun <T, S> Promise<Promise<T>>.then(\n noinline onFulfilled: ((T) -> S)?,\n    noinline onRejected: ((Throwable) -> S)?\n): Promise<S> {\n    return this.unsafeCast<Promise<T>>().then(onFulfilled, onRejected)\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in
 the license/LICENSE.txt file.\n */\n\npackage kotlin.random\n\nimport kotlin.math.pow\n\ninternal actual fun defaultPlatformRandom(): Random =\n    Random(js(\"(Math.random() * Math.pow(2, 32)) | 0\").unsafeCast<Int>())\n\n\nprivate val INV_2_26: Double = 2.0.pow(-26)\nprivate val INV_2_53: Double = 2.0.pow(-53)\ninternal actual fun doubleFromParts(hi26: Int, low27: Int): Double =\n    hi26 * INV_2_26 + low27 * INV_2_53","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.reflect\n\nimport findAssociatedObject\n\n/**\n * The experimental marker for associated objects API.\n *\n * Any usage of a declaration annotated with `@ExperimentalAssociatedObjects` must be accepted either by\n * annotating that usage with the [OptIn] annotation, e.g. `@OptIn(ExperimentalAssociatedObjects::class)`,\n * or by using the compiler
 argument `-opt-in=kotlin.reflect.ExperimentalAssociatedObjects`.\n */\n@RequiresOptIn(level = RequiresOptIn.Level.ERROR)\n@Retention(value = AnnotationRetention.BINARY)\npublic annotation class ExperimentalAssociatedObjects\n\n/**\n * Makes the annotated annotation class an associated object key.\n *\n * An associated object key annotation should have single [KClass] parameter.\n * When applied to a class with reference to an object declaration as an argument, it binds\n * the object to the class, making this binding discoverable at runtime using [findAssociatedObject].\n */\n@ExperimentalAssociatedObjects\n@Retention(AnnotationRetention.BINARY)\n@Target(AnnotationTarget.ANNOTATION_CLASS)\npublic annotation class AssociatedObjectKey\n\n/**\n * If [T] is an @[AssociatedObjectKey]-annotated annotation class and [this] class is annotated with @[T] (`S::class`),\n * returns object `S`.\n *\n * Otherwise returns `null`.\n */\n@ExperimentalAssociatedObjects\npublic inline fun <reified
 T : Annotation> KClass<*>.findAssociatedObject(): Any? =\n    this.findAssociatedObject(T::class)","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\nimport getKClass\nimport kotlin.reflect.KClass\nimport kotlin.reflect.js.internal.KClassImpl\n\n/**\n * Represents the constructor of a class. Instances of `JsClass` can be passed to JavaScript APIs that expect a constructor reference.\n */\nexternal interface JsClass<T : Any> {\n    /**\n     * Returns the unqualified name of the class represented by this instance.\n     */\n    val name: String\n}\n\n/**\n * Obtains a constructor reference for the given `KClass`.\n */\nval <T : Any> KClass<T>.js: JsClass<T>\n    get() = (this as KClassImpl<T>).jClass\n\n/**\n * Obtains a `KClass` instance for the given constructor reference.\n */\nval <T : Any> JsClass<T>.kotlin:
 KClass<T>\n    get() = getKClass(this)\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.reflect.js.internal\n\nimport kotlin.reflect.*\n\ninternal abstract class KClassImpl<T : Any>(\n    internal open val jClass: JsClass<T>\n) : KClass<T> {\n\n    override val qualifiedName: String?\n        get() = TODO()\n\n    override fun equals(other: Any?): Boolean {\n        return other is KClassImpl<*> && jClass == other.jClass\n    }\n\n    // TODO: use FQN\n    override fun hashCode(): Int = simpleName?.hashCode() ?: 0\n\n    override fun toString(): String {\n        // TODO: use FQN\n        return \"class $simpleName\"\n    }\n}\n\ninternal class SimpleKClassImpl<T : Any>(jClass: JsClass<T>) : KClassImpl<T>(jClass) {\n    override val simpleName: String? =
```

jClass.asDynamic().`$metadata$`?.simpleName.unsafeCast<String?>()\n\n
override fun isInstance(value: Any?): Boolean {\n return jsIsType(value, jClass)\n }\n}\n\ninternal class PrimitiveKClassImpl<T : Any>(\n jClass: JsClass<T>,\n private val givenSimpleName: String,\n private val isInstanceFunction: (Any?) -> Boolean\n) : KClassImpl<T>(jClass) {\n override fun equals(other: Any?): Boolean {\n if (other !is PrimitiveKClassImpl<*>) return false\n return super.equals(other) && givenSimpleName == other.givenSimpleName\n }\n\n override val simpleName: String? get() = givenSimpleName\n\n override fun isInstance(value: Any?): Boolean {\n return isInstanceFunction(value)\n }\n}\n\ninternal object NothingKClassImpl : KClassImpl<Nothing>(js(\"Object\")) {\n override val simpleName: String = \"Nothing\"\n\n override fun isInstance(value: Any?): Boolean = false\n\n override val jClass: JsClass<Nothing>\n get() = throw UnsupportedOperationException(\"There's no native JS class for Nothing type\")\n\n override fun equals(other: Any?): Boolean = other === this\n\n override fun hashCode(): Int = 0\n}\n\ninternal class ErrorKClass : KClass<Nothing> {\n override val simpleName: String? get() = error(\"Unknown simpleName for ErrorKClass\")\n override val qualifiedName: String? get() = error(\"Unknown qualifiedName for ErrorKClass\")\n\n override fun isInstance(value: Any?): Boolean = error(\"Can's check isInstance on ErrorKClass\")\n\n override fun equals(other: Any?): Boolean = other === this\n\n override fun hashCode(): Int = 0\n}","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.reflect\n\ninternal actual inline val KClass<*>.qualifiedOrSimpleName: String?\n get() = simpleName","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// a package is omitted to get declarations directly under the module\n\n// TODO: Remove once JsReflectionAPICallChecker supports more reflection types\n@file:Suppress(\"Unsupported\")\n\nimport kotlin.reflect.*\nimport kotlin.reflect.js.internal.*\n\n@JsName(\"createKType\")\ninternal fun createKType(\n classifier: KClassifier,\n arguments: Array<KTypeProjection>,\n isMarkedNullable: Boolean\n) =\n KTypeImpl(classifier, arguments.asList(), isMarkedNullable)\n\n@JsName(\"createDynamicKType\")\ninternal fun createDynamicKType(): KType = DynamicKType\n\n@JsName(\"markKTypeNullable\")\ninternal fun markKTypeNullable(kType: KType) = KTypeImpl(kType.classifier!!, kType.arguments, true)\n\n@JsName(\"createKTypeParameter\")\ninternal fun createKTypeParameter(\n name: String,\n upperBounds: Array<KType>,\n variance: String\n): KTypeParameter {\n val kVariance = when (variance) {\n \"in\" -> KVariance.IN\n \"out\" -> KVariance.OUT\n else -> KVariance.INVARIANT\n }\n\n return KTypeParameterImpl(name, upperBounds.asList(), kVariance, false)\n}\n\n@JsName(\"getStarKTypeProjection\")\ninternal fun getStarKTypeProjection(): KTypeProjection =\n KTypeProjection.STAR\n\n@JsName(\"createCovariantKTypeProjection\")\ninternal fun createCovariantKTypeProjection(type: KType): KTypeProjection =\n KTypeProjection.covariant(type)\n\n@JsName(\"createInvariantKTypeProjection\")\ninternal fun createInvariantKTypeProjection(type: KType): KTypeProjection =\n KTypeProjection.invariant(type)\n\n@JsName(\"createContravariantKTypeProjection\")\ninternal fun createContravariantKTypeProjection(type: KType): KTypeProjection =\n KTypeProjection.contravariant(type)\n","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.reflect.js.internal\n\nimport kotlin.reflect.*\n\ninternal class KTypeImpl(\n override val classifier: KClassifier,\n override val arguments: List<KTypeProjection>,\n override val isMarkedNullable: Boolean\n) : KType {\n override fun equals(other: Any?): Boolean =\n other is KTypeImpl &&\n classifier == other.classifier && arguments == other.arguments && isMarkedNullable == other.isMarkedNullable\n\n override fun hashCode(): Int =\n (classifier.hashCode() * 31 + arguments.hashCode()) * 31 +

```
isMarkedNullable.hashCode()\n\n    override fun toString(): String {\n        val kClass = (classifier as? KClass<*>)\n        val classifierName = when {\n            kClass == null -> classifier.toString()\n            kClass.simpleName != null -> kClass.simpleName\n            else -> \"(non-denotable type)\"\n        }\n\n        val args =\n            if (arguments.isEmpty()) \"\"\n            else arguments.joinToString(\", \", \"<\", \">\")\n        val nullable = if (isMarkedNullable) \"?\" else \"\"\n\n        return classifierName + args + nullable\n    }\n}\n\ninternal object DynamicKType : KType {\n    override val classifier: KClassifier? = null\n    override val arguments: List<KTypeProjection> = emptyList()\n    override val isMarkedNullable: Boolean = false\n    override fun toString(): String = \"dynamic\"\n}\n","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.reflect.js.internal\n\nimport kotlin.reflect.*\n\ninternal data class KTypeParameterImpl(\n    override val name: String,\n    override val upperBounds: List<KType>,\n    override val variance: KVariance,\n    override val isReified: Boolean\n) : KTypeParameter {\n    override fun toString(): String = name\n}","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.reflect.js.internal\n\nimport kotlin.js.JsClass\n\n@JsName(\"PrimitiveClasses\")\ninternal object PrimitiveClasses {\n    @JsName(\"anyClass\")\n    val anyClass = PrimitiveKClassImpl(js(\"Object\").unsafeCast<JsClass<Any>>(), \"Any\", { it is Any })\n\n    @JsName(\"numberClass\")\n    val numberClass = PrimitiveKClassImpl(js(\"Number\").unsafeCast<JsClass<Number>>(), \"Number\", { it is Number })\n\n    @JsName(\"nothingClass\")\n    val nothingClass = NothingKClassImpl\n\n    @JsName(\"booleanClass\")\n    val booleanClass = PrimitiveKClassImpl(js(\"Boolean\").unsafeCast<JsClass<Boolean>>(), \"Boolean\", { it is Boolean })\n\n    @JsName(\"byteClass\")\n    val byteClass = PrimitiveKClassImpl(js(\"Number\").unsafeCast<JsClass<Byte>>(), \"Byte\", { it is Byte })\n\n    @JsName(\"shortClass\")\n    val shortClass = PrimitiveKClassImpl(js(\"Number\").unsafeCast<JsClass<Short>>(), \"Short\", { it is Short })\n\n    @JsName(\"intClass\")\n    val intClass = PrimitiveKClassImpl(js(\"Number\").unsafeCast<JsClass<Int>>(), \"Int\", { it is Int })\n\n    @JsName(\"floatClass\")\n    val floatClass = PrimitiveKClassImpl(js(\"Number\").unsafeCast<JsClass<Float>>(), \"Float\", { it is Float })\n\n    @JsName(\"doubleClass\")\n    val doubleClass = PrimitiveKClassImpl(js(\"Number\").unsafeCast<JsClass<Double>>(), \"Double\", { it is Double })\n\n    @JsName(\"arrayClass\")\n    val arrayClass = PrimitiveKClassImpl(js(\"Array\").unsafeCast<JsClass<Array<*>>>(), \"Array\", { it is Array<*> })\n\n    @JsName(\"stringClass\")\n    val stringClass = PrimitiveKClassImpl(js(\"String\").unsafeCast<JsClass<String>>(), \"String\", { it is String })\n\n    @JsName(\"throwableClass\")\n    val throwableClass = PrimitiveKClassImpl(js(\"Error\").unsafeCast<JsClass<Throwable>>(), \"Throwable\", { it is Throwable })\n\n    @JsName(\"booleanArrayClass\")\n    val booleanArrayClass = PrimitiveKClassImpl(js(\"Array\").unsafeCast<JsClass<BooleanArray>>(), \"BooleanArray\", { it is BooleanArray })\n\n    @JsName(\"charArrayClass\")\n    val charArrayClass = PrimitiveKClassImpl(js(\"Uint16Array\").unsafeCast<JsClass<CharArray>>(), \"CharArray\", { it is CharArray })\n\n    @JsName(\"byteArrayClass\")\n    val byteArrayClass = PrimitiveKClassImpl(js(\"Int8Array\").unsafeCast<JsClass<ByteArray>>(), \"ByteArray\", { it is ByteArray })\n\n    @JsName(\"shortArrayClass\")\n    val shortArrayClass = PrimitiveKClassImpl(js(\"Int16Array\").unsafeCast<JsClass<ShortArray>>(), \"ShortArray\", { it is ShortArray })\n\n    @JsName(\"intArrayClass\")\n    val intArrayClass = PrimitiveKClassImpl(js(\"Int32Array\").unsafeCast<JsClass<IntArray>>(), \"IntArray\", { it is IntArray })\n\n    @JsName(\"longArrayClass\")\n    val longArrayClass = PrimitiveKClassImpl(js(\"Array\").unsafeCast<JsClass<LongArray>>(),
```

```
\"LongArray\", { it is LongArray })\n\n    @JsName(\"floatArrayClass\")\n    val floatArrayClass =
PrimitiveKClassImpl(js(\"Float32Array\").unsafeCast<JsClass<FloatArray>>(), \"FloatArray\", { it is FloatArray
})\n\n    @JsName(\"doubleArrayClass\")\n    val doubleArrayClass =
PrimitiveKClassImpl(js(\"Float64Array\").unsafeCast<JsClass<DoubleArray>>(), \"DoubleArray\", { it is
DoubleArray })\n\n    @JsName(\"functionClass\")\n    fun functionClass(arity: Int): KClassImpl<Any> {\n
return functionClasses.get(arity) ?: run {\n            val result =
PrimitiveKClassImpl(js(\"Function\").unsafeCast<JsClass<Any>>(), \"Function$arity\",\n
 { jsTypeOf(it) === \"function\" && it.asDynamic().length === arity })\n            functionClasses.asDynamic()[arity]
= result\n            result\n        }\n    }\n}\n\nprivate val functionClasses =
arrayOfNulls<KClassImpl<Any>>(0)","/*\n *
 Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code
is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// a package is
omitted to get declarations directly under the module\n\nimport kotlin.reflect.*\nimport
kotlin.reflect.js.internal.*\n\n@JsName(\"getKClass\")\ninternal fun <T : Any> getKClass(jClass: Any /*
JsClass<T> | Array<JsClass<T>> */): KClass<T> {\n    return if (js(\"Array\").isArray(jClass)) {\n
getKClassM(jClass.unsafeCast<Array<JsClass<T>>>())\n    } else {\n
getKClass1(jClass.unsafeCast<JsClass<T>>())\n    }\n}\n\n@JsName(\"getKClassM\")\ninternal fun <T : Any>
getKClassM(jClasses: Array<JsClass<T>>): KClass<T> = when (jClasses.size) {\n    1 ->
getKClass1(jClasses[0])\n    0 -> NothingKClassImpl.unsafeCast<KClass<T>>()\n    else ->
ErrorKClass().unsafeCast<KClass<T>>()\n}\n\n@JsName(\"getKClassFromExpression\")\ninternal fun <T : Any>
getKClassFromExpression(e:
 T): KClass<T> =\n    when (jsTypeOf(e)) {\n        \"string\" -> PrimitiveClasses.stringClass\n        \"number\" -> if
(jsBitwiseOr(e, 0).asDynamic() === e) PrimitiveClasses.intClass else PrimitiveClasses.doubleClass\n
\"boolean\" -> PrimitiveClasses.booleanClass\n        \"function\" ->
PrimitiveClasses.functionClass(e.asDynamic().length)\n        else -> {\n            when {\n                e is BooleanArray
-> PrimitiveClasses.booleanArrayClass\n                e is CharArray -> PrimitiveClasses.charArrayClass\n                e
is ByteArray -> PrimitiveClasses.byteArrayClass\n                e is ShortArray -> PrimitiveClasses.shortArrayClass\n
            e is IntArray -> PrimitiveClasses.intArrayClass\n                e is LongArray ->
PrimitiveClasses.longArrayClass\n                e is FloatArray -> PrimitiveClasses.floatArrayClass\n                e is
DoubleArray -> PrimitiveClasses.doubleArrayClass\n                e is KClass<*> -> KClass::class\n
            e is Array<*> -> PrimitiveClasses.arrayClass\n                else -> {\n                    val constructor =
js(\"Object\").getPrototypeOf(e).constructor\n                    when {\n                        constructor === js(\"Object\") ->
PrimitiveClasses.anyClass\n                        constructor === js(\"Error\") -> PrimitiveClasses.throwableClass\n
            else -> {\n                            val jsClass: JsClass<T> = constructor\n                            getKClass1(jsClass)\n
                }\n                    }\n                }\n            }\n        }\n
}.unsafeCast<KClass<T>>()\n\n@JsName(\"getKClass1\")\ninternal fun <T : Any> getKClass1(jClass:
JsClass<T>): KClass<T> {\n    if (jClass === js(\"String\")) return
PrimitiveClasses.stringClass.unsafeCast<KClass<T>>()\n\n    val metadata = jClass.asDynamic().`$metadata$`\n\n
return if (metadata != null) {\n        if (metadata.`$kClass$` == null) {\n            val kClass =
 SimpleKClassImpl(jClass)\n            metadata.`$kClass$` = kClass\n            kClass\n        } else {\n
metadata.`$kClass$`\n        }\n    } else {\n        SimpleKClassImpl(jClass)\n    }\n}","/*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n/**\n * Exposes
the JavaScript [RegExp
object](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/RegExp) to Kotlin.\n
*/\n@Suppress(\"NOT_DOCUMENTED\")\npublic external class RegExp(pattern: String, flags: String? =
definedExternally) {\n\n    public fun test(str: String): Boolean\n\n    public fun exec(str: String): RegExpMatch?\n\n
  public override fun toString(): String\n\n    /**\n     * The lastIndex is a read/write integer property of regular
```

expressions that specifies the index at which to start the next match.\n */\n public var lastIndex: Int\n\n public val global: Boolean\n public val ignoreCase: Boolean\n public val multiline: Boolean\n}\n\n/**\n * Resets the regular expression so that subsequent [RegExp.test] and [RegExp.exec] calls will match starting with the beginning of the input string.\n */\npublic fun RegExp.reset() {\n lastIndex = 0\n}\n\n// TODO: Inherit from array or introduce asArray() extension\n/**\n * Represents the return value of [RegExp.exec].\n */\n@Suppress(\"NOT_DOCUMENTED\")\npublic external interface RegExpMatch {\n public val index: Int\n public val input: String\n public val length: Int\n}\n\n/**\n * Returns the entire text matched by [RegExp.exec] if the [index] parameter is 0, or the text matched by the capturing parenthesis\n * at the given index.\n */\npublic inline operator fun RegExpMatch.get(index: Int): String? = asDynamic()[index]\n\n/**\n * Converts the result of [RegExp.exec] to an array where the first element contains the entire
matched text and each subsequent\n * element is the text matched by each capturing parenthesis.\n */\npublic inline fun RegExpMatch.asArray(): Array<out String?> = unsafeCast<Array<out String?>>()\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.sequences\n\ninternal actual class ConstrainedOnceSequence<T> actual constructor(sequence: Sequence<T>) : Sequence<T> {\n private var sequenceRef: Sequence<T>? = sequence\n\n actual override fun iterator(): Iterator<T> {\n val sequence = sequenceRef ?: throw IllegalStateException(\"This sequence can be consumed only once.\")\n sequenceRef = null\n return sequence.iterator()\n }\n}\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n@SinceKotlin(\"1.5\")\npublic actual enum class CharCategory(internal val value: Int, public actual val code: String) {\n /**\n * General category \"Cn\" in the Unicode specification.\n */\n UNASSIGNED(0, \"Cn\"),\n\n /**\n * General category \"Lu\" in the Unicode specification.\n */\n UPPERCASE_LETTER(1, \"Lu\"),\n\n /**\n * General category \"Ll\" in the Unicode specification.\n */\n LOWERCASE_LETTER(2, \"Ll\"),\n\n /**\n * General category \"Lt\" in the Unicode specification.\n */\n TITLECASE_LETTER(3, \"Lt\"),\n\n /**\n * General category \"Lm\" in the Unicode specification.\n */\n MODIFIER_LETTER(4, \"Lm\"),\n\n /**\n * General category \"Lo\" in the Unicode specification.\n */\n OTHER_LETTER(5, \"Lo\"),\n\n /**\n * General category \"Mn\" in the Unicode specification.\n */\n NON_SPACING_MARK(6,
\"Mn\"),\n\n /**\n * General category \"Me\" in the Unicode specification.\n */\n ENCLOSING_MARK(7, \"Me\"),\n\n /**\n * General category \"Mc\" in the Unicode specification.\n */\n COMBINING_SPACING_MARK(8, \"Mc\"),\n\n /**\n * General category \"Nd\" in the Unicode specification.\n */\n DECIMAL_DIGIT_NUMBER(9, \"Nd\"),\n\n /**\n * General category \"Nl\" in the Unicode specification.\n */\n LETTER_NUMBER(10, \"Nl\"),\n\n /**\n * General category \"No\" in the Unicode specification.\n */\n OTHER_NUMBER(11, \"No\"),\n\n /**\n * General category \"Zs\" in the Unicode specification.\n */\n SPACE_SEPARATOR(12, \"Zs\"),\n\n /**\n * General category \"Zl\" in the Unicode specification.\n */\n LINE_SEPARATOR(13, \"Zl\"),\n\n /**\n * General category \"Zp\" in the Unicode specification.\n */\n PARAGRAPH_SEPARATOR(14, \"Zp\"),\n\n /**\n * General category \"Cc\" in the Unicode
specification.\n */\n CONTROL(15, \"Cc\"),\n\n /**\n * General category \"Cf\" in the Unicode specification.\n */\n FORMAT(16, \"Cf\"),\n\n /**\n * General category \"Co\" in the Unicode specification.\n */\n PRIVATE_USE(18, \"Co\"),\n\n /**\n * General category \"Cs\" in the Unicode specification.\n */\n SURROGATE(19, \"Cs\"),\n\n /**\n * General category \"Pd\" in the Unicode specification.\n */\n DASH_PUNCTUATION(20, \"Pd\"),\n\n /**\n * General category \"Ps\" in the Unicode specification.\n */\n START_PUNCTUATION(21, \"Ps\"),\n\n /**\n * General category \"Pe\" in the Unicode specification.\n */\n END_PUNCTUATION(22, \"Pe\"),\n\n /**\n * General category \"Pc\" in the Unicode specification.\n */\n CONNECTOR_PUNCTUATION(23, \"Pc\"),\n\n /**\n * General category \"Po\" in the Unicode specification.\n */\n OTHER_PUNCTUATION(24, \"Po\"),\n\n /**\n *

General
category \"Sm\" in the Unicode specification.\n */\n MATH_SYMBOL(25, \"Sm\"),\n\n /**\n * General category \"Sc\" in the Unicode specification.\n */\n CURRENCY_SYMBOL(26, \"Sc\"),\n\n /**\n * General category \"Sk\" in the Unicode specification.\n */\n MODIFIER_SYMBOL(27, \"Sk\"),\n\n /**\n * General category \"So\" in the Unicode specification.\n */\n OTHER_SYMBOL(28, \"So\"),\n\n /**\n * General category \"Pi\" in the Unicode specification.\n */\n INITIAL_QUOTE_PUNCTUATION(29, \"Pi\"),\n\n /**\n * General category \"Pf\" in the Unicode specification.\n */\n FINAL_QUOTE_PUNCTUATION(30, \"Pf\");\n\n /**\n * Returns `true` if [char] character belongs to this category.\n */\n public actual operator fun contains(char: Char): Boolean = char.getCategoryValue() == this.value\n\n companion object {\n internal fun valueOf(category: Int): CharCategory =\n when (category) {\n in 0..16 -> values()[category]\n in 18..30 -> values()[category - 1]\n else -> throw IllegalArgumentException(\"Category #$category is not defined.\")\n }\n }\n}\n","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/**\n * The exception thrown when a character encoding or decoding error occurs.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic actual open class CharacterCodingException(message: String?) : Exception(message) {\n actual constructor() : this(null)\n}\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/**\n * A mutable
sequence of characters.\n *\n * String builder can be used to efficiently perform multiple string manipulation operations.\n */\npublic actual class StringBuilder actual constructor(content: String) : Appendable, CharSequence {\n /**\n * Constructs an empty string builder with the specified initial [capacity].\n *\n * In Kotlin/JS implementation of StringBuilder the initial capacity has no effect on the further performance of operations.\n */\n actual constructor(capacity: Int) : this() {\n }\n\n /** Constructs a string builder that contains the same characters as the specified [content] char sequence. */\n actual constructor(content: CharSequence) : this(content.toString()) {}\n\n /** Constructs an empty string builder. */\n actual constructor() : this(\"\")\n\n private var string: String = if (content !== undefined) content else \"\"\n\n actual override val length: Int\n get() = string.asDynamic().length\n\n actual override fun
get(index: Int): Char =\n string.getOrElse(index) { throw IndexOutOfBoundsException(\"index: $index, length: $length}\") }\n\n actual override fun subSequence(startIndex: Int, endIndex: Int): CharSequence = string.substring(startIndex, endIndex)\n\n actual override fun append(value: Char): StringBuilder {\n string += value\n return this\n }\n\n actual override fun append(value: CharSequence?): StringBuilder {\n string += value.toString()\n return this\n }\n\n actual override fun append(value: CharSequence?, startIndex: Int, endIndex: Int): StringBuilder =\n this.appendRange(value ?: \"null\", startIndex, endIndex)\n\n /**\n * Reverses the contents of this string builder and returns this instance.\n *\n * Surrogate pairs included in this string builder are treated as single characters.\n * Therefore, the order of the high-low surrogates is never reversed.\n *\n * Note that the reverse operation may
produce new surrogate pairs that were unpaired low-surrogates and high-surrogates before the operation.\n * For example, reversing `\"\\uDC00\\uD800\"` produces `\"\\uD800\\uDC00\"` which is a valid surrogate pair.\n */\n actual fun reverse(): StringBuilder {\n var reversed = \"\"\n var index = string.length - 1\n while (index >= 0) {\n val low = string[index--]\n if (low.isLowSurrogate() && index >= 0) {\n val high = string[index--]\n if (high.isHighSurrogate()) {\n reversed = reversed + high + low\n } else {\n reversed = reversed + low + high\n }\n } else {\n reversed += low\n }\n }\n string = reversed\n return this\n }\n\n /**\n * Appends the string representation of the specified object [value] to this string builder and returns this instance.\n *\n * The overall

```
 effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n     * and then that string was appended to this string builder.\n     */\n    actual fun append(value: Any?): StringBuilder {\n        string += value.toString()\n        return this\n    }\n\n    /**\n     * Appends the string representation of the specified boolean [value] to this string builder and returns this instance.\n     *\n     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n     * and then that string was appended to this string builder.\n     */\n    @SinceKotlin(\"1.3\")\n    actual fun append(value: Boolean): StringBuilder {\n        string += value\n        return this\n    }\n\n    /**\n     * Appends characters in the specified character array [value] to this string builder and returns this instance.\n     *\n     * Characters are appended in order, starting at the index 0.\n     */\n    @SinceKotlin(\"1.4\")\n
    @WasExperimental(ExperimentalStdlibApi::class)\n    actual fun append(value: CharArray): StringBuilder {\n        string += value.concatToString()\n        return this\n    }\n\n    /**\n     * Appends the specified string [value] to this string builder and returns this instance.\n     *\n     * If [value] is `null`, then the four characters `\"null\"` are appended.\n     */\n    @SinceKotlin(\"1.3\")\n    actual fun append(value: String?): StringBuilder {\n        this.string += value ?: \"null\"\n        return this\n    }\n\n    /**\n     * Returns the current capacity of this string builder.\n     *\n
     * The capacity is the maximum length this string builder can have before an allocation occurs.\n     *\n     * In Kotlin/JS implementation of StringBuilder the value returned from this method may not indicate the actual size of the backing storage.\n     */\n    @SinceKotlin(\"1.3\")\n// @ExperimentalStdlibApi\n    @Deprecated(\"Obtaining StringBuilder capacity is
        not supported in JS and common code.\", level = DeprecationLevel.ERROR)\n    actual fun capacity(): Int = length\n\n    /**\n     * Ensures that the capacity of this string builder is at least equal to the specified [minimumCapacity].\n     *\n     * If the current capacity is less than the [minimumCapacity], a new backing storage is allocated with greater capacity.\n     * Otherwise, this method takes no action and simply returns.\n     *\n     * In Kotlin/JS implementation of StringBuilder the size of the backing storage is not extended to comply the given [minimumCapacity],\n     * thus calling this method has no effect on the further performance of operations.\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    actual fun ensureCapacity(minimumCapacity: Int) {\n    }\n\n    /**\n     * Returns the index within this string builder of the first occurrence of the specified [string].\n     *\n     * Returns `-1` if the specified [string] does not
        occur in this string builder.\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    actual fun indexOf(string: String): Int = this.string.asDynamic().indexOf(string)\n\n    /**\n     * Returns the index within this string builder of the first occurrence of the specified [string],\n     * starting at the specified [startIndex].\n     *\n     * Returns `-1` if the specified [string] does not occur in this string builder starting at the specified [startIndex].\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    actual fun indexOf(string: String, startIndex: Int): Int = this.string.asDynamic().indexOf(string, startIndex)\n\n    /**\n     * Returns the index within this string builder of the last occurrence of the specified [string].\n     * The last occurrence of empty string `\"\"` is considered to be at the index equal to `this.length`.\n     *\n     * Returns `-1` if the specified [string] does not occur in
        this string builder.\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    actual fun lastIndexOf(string: String): Int = this.string.asDynamic().lastIndexOf(string)\n\n    /**\n     * Returns the index within this string builder of the last occurrence of the specified [string],\n     * starting from the specified [startIndex] toward the beginning.\n     *\n     * Returns `-1` if the specified [string] does not occur in this string builder starting at the specified [startIndex].\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    actual fun lastIndexOf(string: String, startIndex: Int): Int {\n
        if (string.isEmpty() && startIndex < 0) return -1\n        return this.string.asDynamic().lastIndexOf(string, startIndex)\n    }\n\n    /**\n     * Inserts the string representation of the specified boolean [value] into this string builder at the specified [index] and returns this instance.\n     *\n     * The
        overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n     * and then that string was inserted into this string builder at the specified [index].\n     *\n     * @throws
```

```
IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n     */\n
@SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    actual fun insert(index: Int, value:
Boolean): StringBuilder {\n        AbstractList.checkPositionIndex(index, length)\n\n        string = string.substring(0,
index) + value + string.substring(index)\n        return this\n    }\n\n    /**\n     * Inserts the specified character [value]
into this string builder at the specified [index] and returns this instance.\n     *\n     * @throws
IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n     */\n
@SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n
    actual fun insert(index: Int, value: Char): StringBuilder {\n        AbstractList.checkPositionIndex(index,
length)\n\n        string = string.substring(0, index) + value + string.substring(index)\n        return this\n    }\n\n
/**\n     * Inserts characters in the specified character array [value] into this string builder at the specified [index]
and returns this instance.\n     *\n     * The inserted characters go in same order as in the [value] character array,
starting at [index].\n     *\n     * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the
length of this string builder.\n     */\n    @SinceKotlin(\"1.4\")\n
@WasExperimental(ExperimentalStdlibApi::class)\n    actual fun insert(index: Int, value: CharArray): StringBuilder
{\n        AbstractList.checkPositionIndex(index, length)\n\n        string = string.substring(0, index) +
value.concatToString() + string.substring(index)\n        return this\n    }\n\n    /**\n     * Inserts
 characters in the specified character sequence [value] into this string builder at the specified [index] and returns this
instance.\n     *\n     * The inserted characters go in the same order as in the [value] character sequence, starting at
[index].\n     *\n     * @param index the position in this string builder to insert at.\n     * @param value the character
sequence from which characters are inserted. If [value] is `null`, then the four characters `\"null\"` are inserted.\n
*\n     * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string
builder.\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    actual fun
insert(index: Int, value: CharSequence?): StringBuilder {\n        AbstractList.checkPositionIndex(index, length)\n\n
    string = string.substring(0, index) + value.toString() + string.substring(index)\n        return this\n    }\n\n    /**\n
* Inserts the string representation
 of the specified object [value] into this string builder at the specified [index] and returns this instance.\n     *\n     *
The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n     * and
then that string was inserted into this string builder at the specified [index].\n     *\n     * @throws
IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n     */\n
@SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    actual fun insert(index: Int, value:
Any?): StringBuilder {\n        AbstractList.checkPositionIndex(index, length)\n\n        string = string.substring(0,
index) + value.toString() + string.substring(index)\n        return this\n    }\n\n    /**\n     * Inserts the string [value]
into this string builder at the specified [index] and returns this instance.\n     *\n     * If [value] is `null`, then the four
characters `\"null\"` are inserted.\n
    *\n     * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string
builder.\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    actual fun
insert(index: Int, value: String?): StringBuilder {\n        AbstractList.checkPositionIndex(index, length)\n\n        val
toInsert = value ?: \"null\"\n        this.string = this.string.substring(0, index) + toInsert + this.string.substring(index)\n
    return this\n    }\n\n    /**\n     *  Sets the length of this string builder to the specified [newLength].\n     *\n     *
If the [newLength] is less than the current length, it is changed to the specified [newLength].\n     *  Otherwise, null
characters '\\u0000' are appended to this string builder until its length is less than the [newLength].\n     *\n     *
Note that in Kotlin/JS [set] operator function has non-constant execution time complexity.\n     *  Therefore,
increasing length of this string
 builder and then updating each character by index may slow down your program.\n     *\n     *  @throws
IndexOutOfBoundsException or [IllegalArgumentException] if [newLength] is less than zero.\n     */\n
@SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    actual fun setLength(newLength:
Int) {\n        if (newLength < 0) {\n            throw IllegalArgumentException(\"Negative new length:
$newLength.\")\n        }\n\n        if (newLength <= length) {\n            string = string.substring(0, newLength)\n        }
```

```
else {\n            for (i in length until newLength) {\n                string += '\\u0000'\n            }\n        }\n    }\n\n    /**\n     * Returns a new [String] that contains characters in this string builder at [startIndex] (inclusive) and up to the [length] (exclusive).\n     *\n     * @throws IndexOutOfBoundsException if [startIndex] is less than zero or greater than the length of this string builder.\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    actual fun substring(startIndex: Int): String {\n        AbstractList.checkPositionIndex(startIndex, length)\n\n        return string.substring(startIndex)\n    }\n\n    /**\n     * Returns a new [String] that contains characters in this string builder at [startIndex] (inclusive) and up to the [endIndex] (exclusive).\n     *\n     * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this string builder indices or when `startIndex > endIndex`.\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    actual fun substring(startIndex: Int, endIndex: Int): String {\n        AbstractList.checkBoundsIndexes(startIndex, endIndex, length)\n\n        return string.substring(startIndex, endIndex)\n    }\n\n    /**\n     * Attempts to reduce storage used for this string builder.\n     *\n     * If the backing storage of this string builder
    is larger than necessary to hold its current contents,\n     * then it may be resized to become more space efficient.\n     * Calling this method may, but is not required to, affect the value of the [capacity] property.\n     *\n     * In Kotlin/JS implementation of StringBuilder the size of the backing storage is always equal to the length of the string builder.\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    actual fun trimToSize() {\n    }\n\n    override fun toString(): String = string\n\n    /**\n     * Clears the content of this string builder making it empty and returns this instance.\n     *\n     * @sample samples.text.Strings.clearStringBuilder\n     */\n    @SinceKotlin(\"1.3\")\n    public fun clear(): StringBuilder {\n        string = \"\"\n        return this\n    }\n\n    /**\n     * Sets the character at the specified [index] to the specified [value].\n     *\n     * @throws IndexOutOfBoundsException if [index] is out of bounds
    of this string builder.\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    public operator fun set(index: Int, value: Char) {\n        AbstractList.checkElementIndex(index, length)\n\n        string = string.substring(0, index) + value + string.substring(index + 1)\n    }\n\n    /**\n     * Replaces characters in the specified range of this string builder with characters in the specified string [value] and returns this instance.\n     *\n     * @param startIndex the beginning (inclusive) of the range to replace.\n     * @param endIndex the end (exclusive) of the range to replace.\n     * @param value the string to replace with.\n     *\n     * @throws IndexOutOfBoundsException or [IllegalArgumentException] if [startIndex] is less than zero, greater than the length of this string builder, or `startIndex > endIndex`.\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    public fun setRange(startIndex: Int,
    endIndex: Int, value: String): StringBuilder {\n        checkReplaceRange(startIndex, endIndex, length)\n\n        this.string = this.string.substring(0, startIndex) + value + this.string.substring(endIndex)\n        return this\n    }\n\n    private fun checkReplaceRange(startIndex: Int, endIndex: Int, length: Int) {\n        if (startIndex < 0 || startIndex > length) {\n            throw IndexOutOfBoundsException(\"startIndex: $startIndex, length: $length\")\n        }\n        if (startIndex > endIndex) {\n            throw IllegalArgumentException(\"startIndex($startIndex) > endIndex($endIndex)\")\n        }\n    }\n\n    /**\n     * Removes the character at the specified [index] from this string builder and returns this instance.\n     *\n     * If the `Char` at the specified [index] is part of a supplementary code point, this method does not remove the entire supplementary character.\n     *\n     * @param index the index of `Char` to remove.\n     *\n     * @throws IndexOutOfBoundsException
    if [index] is out of bounds of this string builder.\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    public fun deleteAt(index: Int): StringBuilder {\n        AbstractList.checkElementIndex(index, length)\n\n        string = string.substring(0, index) + string.substring(index + 1)\n        return this\n    }\n\n    /**\n     * Removes characters in the specified range from this string builder and returns this instance.\n     *\n     * @param startIndex the beginning (inclusive) of the range to remove.\n     * @param endIndex the end (exclusive) of the range to remove.\n     *\n     * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] is out of range of this string builder indices or when `startIndex > endIndex`.\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    public fun
```

```
deleteRange(startIndex: Int, endIndex: Int): StringBuilder {\n        checkReplaceRange(startIndex,
 endIndex, length)\n\n        string = string.substring(0, startIndex) + string.substring(endIndex)\n        return this\n
}\n\n    /**\n     * Copies characters from this string builder into the [destination] character array.\n     *\n     *
@param destination the array to copy to.\n     * @param destinationOffset the position in the array to copy to, 0 by
default.\n     * @param startIndex the beginning (inclusive) of the range to copy, 0 by default.\n     * @param
endIndex the end (exclusive) of the range to copy, length of this string builder by default.\n     *\n     * @throws
IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this
string builder indices or when `startIndex > endIndex`.\n     * @throws IndexOutOfBoundsException when the
subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n     *  or when that index
is out of the [destination] array indices range.\n
     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    public fun
toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = this.length) {\n
   AbstractList.checkBoundsIndexes(startIndex, endIndex, length)\n
AbstractList.checkBoundsIndexes(destinationOffset, destinationOffset + endIndex - startIndex, destination.size)\n\n
     var dstIndex = destinationOffset\n        for (index in startIndex until endIndex) {\n            destination[dstIndex++]
= string[index]\n        }\n    }\n\n    /**\n     * Appends characters in a subarray of the specified character array
[value] to this string builder and returns this instance.\n     *\n     * Characters are appended in order, starting at
specified [startIndex].\n     *\n     * @param value the array from which characters are appended.\n     * @param
startIndex the beginning (inclusive) of the subarray to append.\n     * @param endIndex the end (exclusive)
 of the subarray to append.\n     *\n     * @throws IndexOutOfBoundsException or [IllegalArgumentException]
when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.\n
*/\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    public fun
appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder {\n        string +=
value.concatToString(startIndex, endIndex)\n        return this\n    }\n\n    /**\n     * Appends a subsequence of the
specified character sequence [value] to this string builder and returns this instance.\n     *\n     * @param value the
character sequence from which a subsequence is appended.\n     * @param startIndex the beginning (inclusive) of
the subsequence to append.\n     * @param endIndex the end (exclusive) of the subsequence to append.\n     *\n     *
@throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of
 range of the [value] character sequence indices or when `startIndex > endIndex`.\n     */\n
@SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    public fun appendRange(value:
CharSequence, startIndex: Int, endIndex: Int): StringBuilder {\n        val stringCsq = value.toString()\n
AbstractList.checkBoundsIndexes(startIndex, endIndex, stringCsq.length)\n\n        string +=
stringCsq.substring(startIndex, endIndex)\n        return this\n    }\n\n    /**\n     * Inserts characters in a subarray of
the specified character array [value] into this string builder at the specified [index] and returns this instance.\n     *\n
  * The inserted characters go in same order as in the [value] array, starting at [index].\n     *\n     * @param index
the position in this string builder to insert at.\n     * @param value the array from which characters are inserted.\n
* @param startIndex the beginning (inclusive) of the subarray to insert.\n     * @param endIndex
 the end (exclusive) of the subarray to insert.\n     *\n     * @throws IndexOutOfBoundsException or
[IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when
`startIndex > endIndex`.\n     * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the
length of this string builder.\n     */\n    @SinceKotlin(\"1.4\")\n
@WasExperimental(ExperimentalStdlibApi::class)\n    public fun insertRange(index: Int, value: CharArray,
startIndex: Int, endIndex: Int): StringBuilder {\n        AbstractList.checkPositionIndex(index, this.length)\n\n
string = string.substring(0, index) + value.concatToString(startIndex, endIndex) + string.substring(index)\n
return this\n    }\n\n    /**\n     * Inserts characters in a subsequence of the specified character sequence [value] into
this string builder at the specified [index] and returns this instance.\n     *\n     * The inserted characters go in the
same order
```

as in the [value] character sequence, starting at [index].\n *\n * @param index the position in this string builder to insert at.\n * @param value the character sequence from which a subsequence is inserted.\n * @param startIndex the beginning (inclusive) of the subsequence to insert.\n * @param endIndex the end (exclusive) of the subsequence to insert.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin(\"1.4\")\n @WasExperimental(ExperimentalStdlibApi::class)\n public fun insertRange(index: Int, value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder {\n AbstractList.checkPositionIndex(index, length)\n\n val stringCsq = value.toString()\n

AbstractList.checkBoundsIndexes(startIndex, endIndex, stringCsq.length)\n\n string = string.substring(0, index) + stringCsq.substring(startIndex, endIndex) + string.substring(index)\n return this\n }\n}\n\n\n/**\n * Clears the content of this string builder making it empty and returns this instance.\n *\n * @sample samples.text.Strings.clearStringBuilder\n */\n@SinceKotlin(\"1.3\")\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\", \"NOTHING_TO_INLINE\")\npublic actual inline fun StringBuilder.clear(): StringBuilder = this.clear()\n\n/**\n * Sets the character at the specified [index] to the specified [value].\n *\n * @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\", \"NOTHING_TO_INLINE\")\npublic actual inline operator fun StringBuilder.set(index: Int, value: Char) = this.set(index, value)\n\n/**\n * Replaces characters in the specified range of this string builder with characters in the specified string [value] and returns this instance.\n *\n * @param startIndex the beginning (inclusive) of the range to replace.\n * @param endIndex the end (exclusive) of the range to replace.\n * @param value the string to replace with.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] if [startIndex] is less than zero, greater than the length of this string builder, or `startIndex > endIndex`.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\", \"NOTHING_TO_INLINE\")\npublic actual inline fun StringBuilder.setRange(startIndex: Int, endIndex: Int, value: String): StringBuilder =\n this.setRange(startIndex, endIndex, value)\n\n/**\n * Removes the character at the specified [index] from this string builder and returns this instance.\n *\n * If the `Char` at the specified [index] is part of a supplementary code point, this method does not remove the entire supplementary character.\n *\n * @param index the index of `Char` to remove.\n *\n * @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\", \"NOTHING_TO_INLINE\")\npublic actual inline fun StringBuilder.deleteAt(index: Int): StringBuilder = this.deleteAt(index)\n\n/**\n * Removes characters in the specified range from this string builder and returns this instance.\n *\n * @param startIndex the beginning (inclusive) of the range to remove.\n * @param endIndex the end (exclusive) of the range to remove.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] is out of range of this string builder indices or when `startIndex > endIndex`.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\", \"NOTHING_TO_INLINE\")\npublic actual inline fun StringBuilder.deleteRange(startIndex: Int, endIndex: Int): StringBuilder = this.deleteRange(startIndex, endIndex)\n\n/**\n * Copies characters from this string builder into the [destination] character array.\n *\n * @param destination the array to copy to.\n * @param destinationOffset the position in the array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the range to copy, 0 by default.\n * @param endIndex the end (exclusive) of the range to copy, length of this string builder by default.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of

range of this string builder indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\",
\"NOTHING_TO_INLINE\", \"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual inline fun StringBuilder.toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = this.length) =\n   this.toCharArray(destination, destinationOffset, startIndex, endIndex)\n\n/**\n * Appends characters in a subarray of the specified character array [value] to this string builder and returns this instance.\n *\n * Characters are appended in order, starting at specified [startIndex].\n *\n * @param value the array from which characters are appended.\n * @param startIndex the beginning (inclusive) of the subarray to append.\n * @param endIndex the end (exclusive) of the subarray to append.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\",
\"NOTHING_TO_INLINE\")\npublic actual inline fun StringBuilder.appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder =\n   this.appendRange(value, startIndex, endIndex)\n\n/**\n * Appends a subsequence of the specified character sequence [value] to this string builder and returns this instance.\n *\n * @param value the character sequence from which a subsequence is appended.\n * @param startIndex the beginning (inclusive) of the subsequence to append.\n * @param endIndex the end (exclusive) of the subsequence to append.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\", \"NOTHING_TO_INLINE\")\npublic
 actual inline fun StringBuilder.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder =\n   this.appendRange(value, startIndex, endIndex)\n\n/**\n * Inserts characters in a subarray of the specified character array [value] into this string builder at the specified [index] and returns this instance.\n *\n * The inserted characters go in same order as in the [value] array, starting at [index].\n *\n * @param index the position in this string builder to insert at.\n * @param value the array from which characters are inserted.\n * @param startIndex the beginning (inclusive) of the subarray to insert.\n * @param endIndex the end (exclusive) of the subarray to insert.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\", \"NOTHING_TO_INLINE\")\npublic actual inline fun StringBuilder.insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder =\n   this.insertRange(index, value, startIndex, endIndex)\n\n/**\n * Inserts characters in a subsequence of the specified character sequence [value] into this string builder at the specified [index] and returns this instance.\n *\n * The inserted characters go in the same order as in the [value] character sequence, starting at [index].\n *\n * @param index the position in this string builder to insert at.\n * @param value the character sequence from which a subsequence is inserted.\n * @param startIndex the beginning (inclusive) of the subsequence to insert.\n * @param endIndex the end (exclusive) of the subsequence to insert.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex]
 or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\", \"NOTHING_TO_INLINE\")\npublic actual inline fun

StringBuilder.insertRange(index: Int, value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder =\n this.insertRange(index, value, startIndex, endIndex)\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n\n/**\n * Returns `true` if the content of this string is equal to the word \"true\", ignoring case, and `false` otherwise.\n */\n@Deprecated(\"Use Kotlin compiler 1.4 to avoid deprecation warning.\")\n@DeprecatedSinceKotlin(hiddenSince = \"1.4\")\n@kotlin.internal.InlineOnly\npublic actual inline fun String.toBoolean(): Boolean = this.toBoolean()\n\n/**\n * Returns `true` if this string is not `null` and its content is equal to the word \"true\", ignoring case, and `false` otherwise.\n *\n * There are also strict versions of the function available on non-nullable String, [toBooleanStrict] and [toBooleanStrictOrNull].\n */\n@SinceKotlin(\"1.4\")\npublic actual fun String?.toBoolean(): Boolean = this != null && this.lowercase() == \"true\"\n\n/**\n * Parses the string as a signed [Byte] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n */\npublic actual fun String.toByte(): Byte = toByteOrNull() ?: numberFormatError(this)\n\n/**\n * Parses the string as a signed [Byte] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n */\npublic actual fun String.toByte(radix: Int): Byte = toByteOrNull(radix) ?: numberFormatError(this)\n\n\n/**\n * Parses the string as a [Short] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n */\npublic actual fun String.toShort(): Short = toShortOrNull() ?: numberFormatError(this)\n\n/**\n * Parses the string as a [Short] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n */\npublic actual fun String.toShort(radix: Int): Short = toShortOrNull(radix) ?: numberFormatError(this)\n\n/**\n * Parses the string as an [Int] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n */\npublic actual fun String.toInt(): Int = toIntOrNull() ?: numberFormatError(this)\n\n/**\n * Parses the string as an [Int] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n */\npublic actual fun String.toInt(radix: Int): Int = toIntOrNull(radix) ?: numberFormatError(this)\n\n/**\n * Parses the string as a [Long] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n */\npublic actual fun String.toLong(): Long = toLongOrNull() ?: numberFormatError(this)\n\n/**\n * Parses the string as a [Long] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n */\npublic actual fun String.toLong(radix: Int): Long = toLongOrNull(radix) ?: numberFormatError(this)\n\n/**\n * Parses the string as a [Double] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n */\npublic actual fun String.toDouble(): Double = (+(this.asDynamic())).unsafeCast<Double>().also {\n if (it.isNaN() && !this.isNaN() || it == 0.0 && this.isBlank())\n numberFormatError(this)\n}\n\n/**\n * Parses the string as a [Float] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun String.toFloat(): Float = toDouble().unsafeCast<Float>()\n\n/**\n * Parses the string as a [Double] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n */\npublic actual fun String.toDoubleOrNull(): Double? = (+(this.asDynamic())).unsafeCast<Double>().takeIf {\n !(it.isNaN() && !this.isNaN() || it == 0.0 && this.isBlank())\n}\n\n/**\n * Parses the string as a [Float] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n */\n@kotlin.internal.InlineOnly\npublic actual inline fun String.toFloatOrNull(): Float? = toDoubleOrNull().unsafeCast<Float?>()\n\n/**\n * Returns a string representation of this [Byte] value in the specified [radix].\n *\n * @throws IllegalArgumentException when

```
[radix] is not a valid radix for number to string conversion.\n
*/\n@SinceKotlin(\"1.2\")\n@kotlin.internal.InlineOnly\npublic actual inline fun Byte.toString(radix: Int): String =
this.toInt().toString(radix)\n\n/**\n * Returns a string representation of this [Short] value in the specified [radix].\n
*\n * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.\n
*/\n@SinceKotlin(\"1.2\")\n@kotlin.internal.InlineOnly\npublic actual inline fun Short.toString(radix: Int): String =
this.toInt().toString(radix)\n\n/**\n
 * Returns a string representation of this [Int] value in the specified [radix].\n *\n * @throws
IllegalArgumentException when [radix] is not a valid radix for number to string conversion.\n
*/\n@SinceKotlin(\"1.2\")\npublic actual fun Int.toString(radix: Int): String =
asDynamic().toString(checkRadix(radix))\n\nprivate fun String.isNaN(): Boolean = when (this.lowercase()) {\n
\"nan\", \"+nan\", \"-nan\" -> true\n    else -> false\n}\n\n/**\n * Checks whether the given [radix] is valid radix for
string to number and number to string conversion.\n */\n@PublishedApi\ninternal actual fun checkRadix(radix: Int):
Int {\n    if (radix !in 2..36) {\n        throw IllegalArgumentException(\"radix $radix was not in valid range 2..36\")\n
  }\n    return radix\n}\n\ninternal actual fun digitOf(char: Char, radix: Int): Int = when {\n    char >= '0' && char <=
'9' -> char - '0'\n    char >= 'A' && char <= 'Z' -> char - 'A' + 10\n    char >= 'a' && char <= 'z' -> char
 - 'a' + 10\n    char < '\\u0080' -> -1\n    char >= '\\uFF21' && char <= '\\uFF3A' -> char - '\\uFF21' + 10 // full-width
latin capital letter\n    char >= '\\uFF41' && char <= '\\uFF5A' -> char - '\\uFF41' + 10 // full-width latin small
letter\n    else -> char.digitToIntImpl()\n}.let { if (it >= radix) -1 else it }\n","/*\n * Copyright 2010-2021 JetBrains
s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\nimport
kotlin.js.RegExp\n\n/**\n * Provides enumeration values to use to set regular expression options.\n */\npublic actual
enum class RegexOption(val value: String) {\n    /** Enables case-insensitive matching. */\n
IGNORE_CASE(\"i\"),\n    /** Enables multiline mode.\n     *\n     * In multiline mode the expressions `^` and `$`
match just after or just before,\n     * respectively, a line terminator or the end of the input sequence.
 */\n    MULTILINE(\"m\")\n}\n\nprivate fun Iterable<RegexOption>.toFlags(prepend: String): String =
joinToString(\"\", prefix = prepend) { it.value }\n\n\n/**\n * Represents the results from a single capturing group
within a [MatchResult] of [Regex].\n *\n * @param value The value of captured group.\n */\npublic actual data
class MatchGroup(actual val value: String)\n\n\n/**\n * Returns a named group with the specified [name].\n *\n *
@return An instance of [MatchGroup] if the group with the specified [name] was matched or `null` otherwise.\n *
@throws IllegalArgumentException if there is no group with the specified [name] defined in the regex pattern.\n *
@throws UnsupportedOperationException if this match group collection doesn't support getting match groups by
name,\n * for example, when it's not supported by the current platform.\n */\n@SinceKotlin(\"1.7\")\npublic
operator fun MatchGroupCollection.get(name: String): MatchGroup? {\n    val namedGroups = this as?
MatchNamedGroupCollection\n
        ?: throw UnsupportedOperationException(\"Retrieving groups by name is not supported on this platform.\")\n\n
  return namedGroups[name]\n}\n\n\n/**\n * Represents a compiled regular expression.\n * Provides functions to
match strings in text with a pattern, replace the found occurrences and split text around matches.\n *\n * For pattern
syntax reference see [MDN RegExp](https://developer.mozilla.org/en-
US/docs/Web/JavaScript/Reference/Global_Objects/RegExp#Special_characters_meaning_in_regular_expressions)\
n * and
[http://www.w3schools.com/jsref/jsref_obj_regexp.asp](https://www.w3schools.com/jsref/jsref_obj_regexp.asp).\n
*\n * Note that `RegExp` objects under the hood are constructed with [the \"u\"
flag](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp/unicode)\n * that
enables Unicode-related features in regular expressions. This also makes the pattern syntax more strict,\n * for
example, prohibiting unnecessary escape
 sequences.\n *\n * @constructor Creates a regular expression from the specified [pattern] string and the specified
set of [options].\n */\npublic actual class Regex actual constructor(pattern: String, options: Set<RegexOption>)
{\n\n    /** Creates a regular expression from the specified [pattern] string and the specified single [option].  */\n
```

```
public actual constructor(pattern: String, option: RegexOption) : this(pattern, setOf(option))\n\n    /** Creates a regular expression from the specified [pattern] string and the default options.  */\n    public actual constructor(pattern: String) : this(pattern, emptySet())\n\n\n    /** The pattern string of this regular expression. */\n    public actual val pattern: String = pattern\n    /** The set of options that were used to create this regular expression. */\n    public actual val options: Set<RegexOption> = options.toSet()\n    private val nativePattern: RegExp = RegExp(pattern, options.toFlags(\"gu\"))\n    private var nativeStickyPattern:
 RegExp? = null\n    private fun initStickyPattern(): RegExp =\n        nativeStickyPattern ?: RegExp(pattern, options.toFlags(\"yu\")).also { nativeStickyPattern = it }\n\n    private var nativeMatchesEntirePattern: RegExp? = null\n    private fun initMatchesEntirePattern(): RegExp =\n        nativeMatchesEntirePattern ?: run {\n            if (pattern.startsWith('^') && pattern.endsWith('$'))\n                nativePattern\n            else\n                return RegExp(\"^${pattern.trimStart('^').trimEnd('$')}$\", options.toFlags(\"gu\"))\n        }.also { nativeMatchesEntirePattern = it }\n\n\n    /** Indicates whether the regular expression matches the entire [input]. */\n    public actual infix fun matches(input: CharSequence): Boolean {\n        nativePattern.reset()\n        val match = nativePattern.exec(input.toString())\n        return match != null && match.index == 0 && nativePattern.lastIndex == input.length\n    }\n\n    /** Indicates whether the regular expression
 can find at least one match in the specified [input]. */\n    public actual fun containsMatchIn(input: CharSequence): Boolean {\n        nativePattern.reset()\n        return nativePattern.test(input.toString())\n    }\n\n    @SinceKotlin(\"1.7\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    public actual fun matchesAt(input: CharSequence, index: Int): Boolean {\n        if (index < 0 || index > input.length) {\n            throw IndexOutOfBoundsException(\"index out of bounds: $index, input length: ${input.length}\")\n        }\n        val pattern = initStickyPattern()\n        pattern.lastIndex = index\n        return pattern.test(input.toString())\n    }\n\n    /**\n     * Returns the first match of a regular expression in the [input], beginning at the specified [startIndex].\n     *\n     * @param startIndex An index to start search with, by default 0. Must be not less than zero and not greater than `input.length()`\n     * @return An instance of [MatchResult] if match
 was found or `null` otherwise.\n     * @throws IndexOutOfBoundsException if [startIndex] is less than zero or greater than the length of the [input] char sequence.\n     * @sample samples.text.Regexps.find\n     */\n    @Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\n    public actual fun find(input: CharSequence, startIndex: Int = 0): MatchResult? {\n        if (startIndex < 0 || startIndex > input.length) {\n            throw IndexOutOfBoundsException(\"Start index out of bounds: $startIndex, input length: ${input.length}\")\n        }\n        return nativePattern.findNext(input.toString(), startIndex, nativePattern)\n    }\n\n    /**\n     * Returns a sequence of all occurrences of a regular expression within the [input] string, beginning at the specified [startIndex].\n     *\n     * @throws IndexOutOfBoundsException if [startIndex] is less than zero or greater than the length of the [input] char sequence.\n     *\n     * @sample samples.text.Regexps.findAll\n     */\n
  @Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\n    public actual fun findAll(input: CharSequence, startIndex: Int = 0): Sequence<MatchResult> {\n        if (startIndex < 0 || startIndex > input.length) {\n            throw IndexOutOfBoundsException(\"Start index out of bounds: $startIndex, input length: ${input.length}\")\n        }\n        return generateSequence({ find(input, startIndex) }, { match -> match.next() })\n    }\n\n    /**\n     * Attempts to match the entire [input] CharSequence against the pattern.\n     *\n     * @return An instance of [MatchResult] if the entire input matches or `null` otherwise.\n     */\n    public actual fun matchEntire(input: CharSequence): MatchResult? =\n        initMatchesEntirePattern().findNext(input.toString(), 0, nativePattern)\n\n    @SinceKotlin(\"1.7\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    public actual fun matchAt(input: CharSequence, index: Int): MatchResult? {\n        if (index < 0 || index > input.length)
 {\n            throw IndexOutOfBoundsException(\"index out of bounds: $index, input length: ${input.length}\")\n        }\n        return initStickyPattern().findNext(input.toString(), index, nativePattern)\n    }\n\n\n    /**\n     * Replaces all occurrences of this regular expression in the specified [input] string with specified [replacement] expression.\n     *\n     * The replacement string may contain references to the captured groups during a match. Occurrences of `${name}` or `$index`\n     * in the replacement string will be substituted with the subsequences corresponding to the captured groups with the specified name or index.\n     * In case of `$index`, the first digit after '$' is always
```

```
treated as a part of group reference. Subsequent digits are incorporated\n     * into `index` only if they would form a valid group reference. Only the digits '0'..'9' are considered as potential components\n     * of the group reference. Note that indexes of captured groups start from
 1, and the group with index 0 is the whole match.\n     * In case of `${name}`, the `name` can consist of latin letters 'a'..'z' and 'A'..'Z', or digits '0'..'9'. The first character must be\n     * a letter.\n     *\n     * Backslash character '\\' can be used to include the succeeding character as a literal in the replacement string, e.g, `\\$` or `\\\\`.\n     * [Regex.escapeReplacement] can be used if [replacement] have to be treated as a literal string.\n     *\n     * @param input the char sequence to find matches of this regular expression in\n     * @param replacement the expression to replace found matches with\n     * @return the result of replacing each occurrence of this regular expression in [input] with the result of evaluating the [replacement] expression\n     * @throws RuntimeException if [replacement] expression is malformed, or capturing group with specified `name` or `index` does not exist\n     */\n public actual fun replace(input: CharSequence, replacement: String):
 String {\n         if (!replacement.contains('\\\\') && !replacement.contains('$')) {\n             return input.toString().nativeReplace(nativePattern, replacement)\n         }\n         return replace(input) { substituteGroupRefs(it, replacement) }\n     }\n\n     /**\n     * Replaces all occurrences of this regular expression in the specified [input] string with the result of\n     * the given function [transform] that takes [MatchResult] and returns a string to be used as a\n     * replacement for that match.\n     */\n     public actual fun replace(input: CharSequence, transform: (MatchResult) -> CharSequence): String {\n         var match = find(input)\n         if (match == null) return input.toString()\n\n         var lastStart = 0\n         val length = input.length\n         val sb = StringBuilder(length)\n         do {\n             val foundMatch = match!!\n             sb.append(input, lastStart, foundMatch.range.start)\n             sb.append(transform(foundMatch))\n             lastStart
 = foundMatch.range.endInclusive + 1\n             match = foundMatch.next()\n         } while (lastStart < length && match != null)\n\n         if (lastStart < length) {\n             sb.append(input, lastStart, length)\n         }\n\n         return sb.toString()\n     }\n\n     /**\n     * Replaces the first occurrence of this regular expression in the specified [input] string with specified [replacement] expression.\n     *\n     * The replacement string may contain references to the captured groups during a match. Occurrences of `${name}` or `$index`\n     * in the replacement string will be substituted with the subsequences corresponding to the captured groups with the specified name or index.\n     * In case of `$index`, the first digit after '$' is always treated as a part of group reference. Subsequent digits are incorporated\n     * into `index` only if they would form a valid group reference. Only the digits '0'..'9' are considered as potential components\n     * of the group
 reference. Note that indexes of captured groups start from 1, and the group with index 0 is the whole match.\n     * In case of `${name}`, the `name` can consist of latin letters 'a'..'z' and 'A'..'Z', or digits '0'..'9'. The first character must be\n     * a letter.\n     *\n     * Backslash character '\\' can be used to include the succeeding character as a literal in the replacement string, e.g, `\\$` or `\\\\`.\n     * [Regex.escapeReplacement] can be used if [replacement] have to be treated as a literal string.\n     *\n     * @param input the char sequence to find a match of this regular expression in\n     * @param replacement the expression to replace the found match with\n     * @return the result of replacing the first occurrence of this regular expression in [input] with the result of evaluating the [replacement] expression\n     * @throws RuntimeException if [replacement] expression is malformed, or capturing group with specified `name` or `index` does not exist\n     */\n
    public actual fun replaceFirst(input: CharSequence, replacement: String): String {\n         if (!replacement.contains('\\\\') && !replacement.contains('$')) {\n             val nonGlobalOptions = options.toFlags(\"u\")\n             return input.toString().nativeReplace(RegExp(pattern, nonGlobalOptions), replacement)\n         }\n\n         val match = find(input) ?: return input.toString()\n\n         return buildString {\n             append(input.substring(0, match.range.first))\n             append(substituteGroupRefs(match, replacement))\n             append(input.substring(match.range.last + 1, input.length))\n         }\n     }\n\n     /**\n     * Splits the [input] CharSequence to a list of strings around matches of this regular expression.\n     *\n     * @param limit Non-negative value specifying the maximum number of substrings the string can be split to.\n     * Zero by default means no limit is set.\n     */\n     @Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\n     public actual
```

```
 fun split(input: CharSequence, limit: Int = 0): List<String> {\n        requireNonNegativeLimit(limit)\n        val matches = findAll(input).let { if (limit == 0) it else it.take(limit - 1) }\n        val result = mutableListOf<String>()\n    var lastStart = 0\n\n        for (match in matches) {\n            result.add(input.subSequence(lastStart, match.range.start).toString())\n            lastStart = match.range.endInclusive + 1\n        }\n result.add(input.subSequence(lastStart, input.length).toString())\n        return result\n    }\n\n    /**\n     * Splits the [input] CharSequence to a sequence of strings around matches of this regular expression.\n     *\n     * @param limit Non-negative value specifying the maximum number of substrings the string can be split to.\n     * Zero by default means no limit is set.\n     * @sample samples.text.Regexps.splitToSequence\n     */\n    @SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\n    public actual fun splitToSequence(input: CharSequence, limit: Int = 0): Sequence<String> {\n requireNonNegativeLimit(limit)\n\n        return sequence {\n            var match = find(input)\n            if (match == null || limit == 1) {\n                yield(input.toString())\n                return@sequence\n            }\n\n            var nextStart = 0\n            var splitCount = 0\n\n            do {\n                val foundMatch = match!!\n yield(input.substring(nextStart, foundMatch.range.first))\n                nextStart = foundMatch.range.endInclusive + 1\n                match = foundMatch.next()\n            } while (++splitCount != limit - 1 && match != null)\n\n yield(input.substring(nextStart, input.length))\n        }\n    }\n\n\n    /**\n     * Returns the string representation of this regular expression, namely the [pattern] of this regular expression.\n     *\n     * Note that another regular expression constructed from the same pattern string may have different [options]\n     * and may match strings differently.\n     */\n    public override fun toString(): String = nativePattern.toString()\n\n    actual companion object {\n        /**\n         * Returns a regular expression that matches the specified [literal] string literally.\n         * No characters of that string will have special meaning when searching for an occurrence of the regular expression.\n         */\n        public actual fun fromLiteral(literal: String): Regex = Regex(escape(literal))\n\n        /**\n         * Returns a regular expression pattern string that matches the specified [literal] string literally.\n         * No characters of that string will have special meaning when searching for an occurrence of the regular expression.\n         */\n        public actual fun escape(literal: String): String = literal.nativeReplace(patternEscape, \"\\\\$&\")\n\n        /**\n         * Returns a literal replacement expression for the specified [literal] string.\n         * No characters of that string will have special meaning when it is used as a replacement string in [Regex.replace] function.\n         */\n        public actual fun escapeReplacement(literal: String): String = literal.nativeReplace(replacementEscape, \"\\\\$&\")\n\n        private val patternEscape = RegExp(\"\\\"\\\"[\\\\^$*+?.()|[\\]{}]\\\"\\\", \"g\")\n        private val replacementEscape = RegExp(\"\\\"\\\"[\\\\$]\\\"\\\", \"g\")\n\n        internal fun nativeEscapeReplacement(literal: String): String = literal.nativeReplace(nativeReplacementEscape, \"$$$$\")\n        private val nativeReplacementEscape = RegExp(\"\\\"\\\"\\$\\\"\\\", \"g\")\n    }\n}\n\n\n\nprivate fun RegExp.findNext(input: String, from: Int, nextPattern: RegExp): MatchResult? {\n    this.lastIndex = from\n    val match = exec(input)\n    if (match == null) return null\n    val range = match.index..lastIndex - 1\n\n    return object : MatchResult {\n        override val range: IntRange = range\n        override val value: String\n            get() = match[0]!!\n\n        override val groups: MatchGroupCollection = object : MatchNamedGroupCollection, AbstractCollection<MatchGroup?>() {\n            override val size: Int get() = match.length\n            override fun iterator(): Iterator<MatchGroup?> = indices.asSequence().map { this[it] }.iterator()\n            override fun get(index: Int): MatchGroup? = match[index]?.let { MatchGroup(it) }\n\n            override fun get(name: String): MatchGroup? {\n                // An object of named capturing groups whose keys are the names and values are the capturing groups\n                // or undefined if no named capturing groups were defined.\n                val groups = match.asDynamic().groups\n                    ?: throw IllegalArgumentException(\"Capturing group with name {$name} does not exist. No named capturing group was defined in Regex\")\n\n                // If the match was successful but the group specified failed to match any part of the input sequence,\n                // the associated value is 'undefined'. Value for a non-existent key is also 'undefined'. Thus, explicitly check if the key exists.\n                if (!hasOwnPrototypeProperty(groups, name))\n                    throw IllegalArgumentException(\"Capturing group
```

```
with name {$name} does not exist\")\n\n                val value = groups[name]\n                return if (value == undefined) null else MatchGroup(value as String)\n            }\n        }\n\n        private fun hasOwnPrototypeProperty(o: Any?, name: String): Boolean {\n            return js(\"Object\").prototype.hasOwnProperty.call(o, name).unsafeCast<Boolean>()\n        }\n\n\n        private var groupValues_: List<String>? = null\n\n        override val groupValues: List<String>\n            get() {\n                if (groupValues_ == null) {\n                    groupValues_ = object : AbstractList<String>() {\n                        override val size: Int get() = match.length\n                        override fun get(index: Int): String = match[index] ?: \"\"\n                    }\n                }\n                return groupValues_!!\n            }\n\n        override fun next(): MatchResult? =\n            nextPattern.findNext(input, if (range.isEmpty()) advanceToNextCharacter(range.start) else range.endInclusive + 1, nextPattern)\n\n        private fun advanceToNextCharacter(index: Int): Int {\n            if (index < input.lastIndex) {\n                val code1 = input.asDynamic().charCodeAt(index).unsafeCast<Int>()\n                if (code1 in 0xD800..0xDBFF) {\n                    val code2 = input.asDynamic().charCodeAt(index + 1).unsafeCast<Int>()\n                    if (code2 in 0xDC00..0xDFFF) {\n                        return index + 2\n                    }\n                }\n            }\n            return index + 1\n        }\n    }\n}\n\n// The same code from K/N Regex.kt\nprivate
 fun substituteGroupRefs(match: MatchResult, replacement: String): String {\n    var index = 0\n    val result = StringBuilder()\n\n    while (index < replacement.length) {\n        val char = replacement[index++]\n        if (char == '\\\\') {\n            if (index == replacement.length)\n                throw IllegalArgumentException(\"The Char to be escaped is missing\")\n\n            result.append(replacement[index++])\n        } else if (char == '$') {\n            if (index == replacement.length)\n                throw IllegalArgumentException(\"Capturing group index is missing\")\n\n            if (replacement[index] == '{') {\n                val endIndex = replacement.readGroupName(++index)\n\n                if (index == endIndex)\n                    throw IllegalArgumentException(\"Named capturing group reference should have a non-empty name\")\n                if (endIndex == replacement.length || replacement[endIndex] != '}')\n                    throw IllegalArgumentException(\"Named
 capturing group reference is missing trailing '}'\")\n\n                val groupName = replacement.substring(index, endIndex)\n\n                result.append(match.groups[groupName]?.value ?: \"\")\n                index = endIndex + 1    // skip past '}'\n            } else {\n                if (replacement[index] !in '0'..'9')\n                    throw IllegalArgumentException(\"Invalid capturing group reference\")\n\n                val groups = match.groups\n                val endIndex = replacement.readGroupIndex(index, groups.size)\n                val groupIndex = replacement.substring(index, endIndex).toInt()\n\n                if (groupIndex >= groups.size)\n                    throw IndexOutOfBoundsException(\"Group with index $groupIndex does not exist\")\n\n                result.append(groups[groupIndex]?.value ?: \"\")\n                index = endIndex\n            }\n        } else {\n            result.append(char)\n        }\n    }\n    return
 result.toString()\n}\n\n// The name must be a legal JavaScript identifier. See https://262.ecma-international.org/5.1/#sec-7.6\n// Don't try to validate the referenced group name as it may be time-consuming.\n// If the name is invalid, it won't be found in `match.groups` anyway and will throw.\n// Group names in the target Regex are validated at creation time.\nprivate fun String.readGroupName(startIndex: Int): Int {\n    var index = startIndex\n
  while (index < length) {\n        if (this[index] == '}') {\n            break\n        } else {\n            index++\n        }\n    }\n    return index\n}\n\nprivate fun String.readGroupIndex(startIndex: Int, groupCount: Int): Int {\n    // at least one digit after '$' is always captured\n    var index = startIndex + 1\n    var groupIndex = this[startIndex] - '0'\n\n    // capture the largest valid group index\n    while (index < length && this[index] in '0'..'9') {\n        val newGroupIndex = (groupIndex * 10) + (this[index] - '0')\n
        if (newGroupIndex in 0 until groupCount) {\n            groupIndex = newGroupIndex\n            index++\n        } else {\n            break\n        }\n    }\n    return index\n}","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"StringsKt\")\n@file:Suppress(\"EXTENSI
```

```
ON_SHADOWED_BY_MEMBER\")\n\npackage kotlin.text\n\nimport kotlin.contracts.*\n\n/**\n * A mutable sequence of characters.\n *\n * String builder can be used to efficiently perform multiple string manipulation operations.\n */\nexpect class StringBuilder : Appendable, CharSequence {\n    /** Constructs an empty string builder. */\n    constructor()\n\n    /** Constructs an empty string builder with the specified initial [capacity]. */\n    constructor(capacity: Int)\n\n    /** Constructs
 a string builder that contains the same characters as the specified [content] char sequence. */\n    constructor(content: CharSequence)\n\n    /** Constructs a string builder that contains the same characters as the specified [content] string. */\n    @SinceKotlin(\"1.3\")\n//    @ExperimentalStdlibApi\n    constructor(content: String)\n\n    override val length: Int\n\n    override operator fun get(index: Int): Char\n\n    override fun subSequence(startIndex: Int, endIndex: Int): CharSequence\n\n    override fun append(value: Char): StringBuilder\n    override fun append(value: CharSequence?): StringBuilder\n    override fun append(value: CharSequence?, startIndex: Int, endIndex: Int): StringBuilder\n\n    /**\n     * Reverses the contents of this string builder and returns this instance.\n     *\n     * Surrogate pairs included in this string builder are treated as single characters.\n     * Therefore, the order of the high-low surrogates is never reversed.\n     *\n     * Note that
 the reverse operation may produce new surrogate pairs that were unpaired low-surrogates and high-surrogates before the operation.\n     * For example, reversing `\"\\uDC00\\uD800\"` produces `\"\\uD800\\uDC00\"` which is a valid surrogate pair.\n     */\n    fun reverse(): StringBuilder\n\n    /**\n     * Appends the string representation of the specified object [value] to this string builder and returns this instance.\n     *\n     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n     * and then that string was appended to this string builder.\n     */\n    fun append(value: Any?): StringBuilder\n\n    /**\n     * Appends the string representation of the specified boolean [value] to this string builder and returns this instance.\n     *\n     * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n     * and then that string was appended to this string builder.\n
 */\n    @SinceKotlin(\"1.3\")\n    fun append(value: Boolean): StringBuilder\n\n    /**\n     * Appends characters in the specified character array [value] to this string builder and returns this instance.\n     *\n     * Characters are appended in order, starting at the index 0.\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    fun append(value: CharArray): StringBuilder\n\n    /**\n     * Appends the specified string [value] to this string builder and returns this instance.\n     *\n     * If [value] is `null`, then the four characters `\"null\"` are appended.\n     */\n    @SinceKotlin(\"1.3\")\n    fun append(value: String?): StringBuilder\n\n    /**\n     * Returns the current capacity of this string builder.\n     *\n     * The capacity is the maximum length this string builder can have before an allocation occurs.\n     */\n    @SinceKotlin(\"1.3\")\n//    @ExperimentalStdlibApi\n    @Deprecated(\"Obtaining StringBuilder capacity is not
 supported in JS and common code.\", level = DeprecationLevel.ERROR)\n    fun capacity(): Int\n\n    /**\n     * Ensures that the capacity of this string builder is at least equal to the specified [minimumCapacity].\n     *\n     * If the current capacity is less than the [minimumCapacity], a new backing storage is allocated with greater capacity.\n     * Otherwise, this method takes no action and simply returns.\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    fun ensureCapacity(minimumCapacity: Int)\n\n    /**\n     * Returns the index within this string builder of the first occurrence of the specified [string].\n     *\n     * Returns `-1` if the specified [string] does not occur in this string builder.\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    fun indexOf(string: String): Int\n\n    /**\n     * Returns the index within this string builder of the first occurrence of the specified [string],\n
     * starting at the specified [startIndex].\n     *\n     * Returns `-1` if the specified [string] does not occur in this string builder starting at the specified [startIndex].\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    fun indexOf(string: String, startIndex: Int): Int\n\n    /**\n     * Returns the index within this string builder of the last occurrence of the specified [string].\n     * The last occurrence of empty string `\"\"` is considered to be at the index equal to `this.length`.\n     *\n     * Returns `-1` if the specified [string] does not occur in this string builder.\n     */\n    @SinceKotlin(\"1.4\")\n    @WasExperimental(ExperimentalStdlibApi::class)\n    fun lastIndexOf(string: String): Int\n\n    /**\n     * Returns
```

the index within this string builder of the last occurrence of the specified [string],\n * starting from the specified [startIndex] toward the beginning.\n *\n * Returns `-1` if the specified
[string] does not occur in this string builder starting at the specified [startIndex].\n */\n @SinceKotlin(\"1.4\")\n
@WasExperimental(ExperimentalStdlibApi::class)\n fun lastIndexOf(string: String, startIndex: Int): Int\n\n
/**\n * Inserts the string representation of the specified boolean [value] into this string builder at the specified [index] and returns this instance.\n *\n * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n * and then that string was inserted into this string builder at the specified [index].\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin(\"1.4\")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: Boolean): StringBuilder\n\n /**\n * Inserts the specified character [value] into this string builder at the specified [index]
and returns this instance.\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin(\"1.4\")\n
@WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: Char): StringBuilder\n\n /**\n * Inserts characters in the specified character array [value] into this string builder at the specified [index] and returns this instance.\n *\n * The inserted characters go in same order as in the [value] character array, starting at [index].\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin(\"1.4\")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: CharArray): StringBuilder\n\n /**\n * Inserts characters in the specified character sequence [value] into this string builder at the specified [index] and returns this
instance.\n *\n * The inserted characters go in the same order as in the [value] character sequence, starting at [index].\n *\n * @param index the position in this string builder to insert at.\n * @param value the character sequence from which characters are inserted. If [value] is `null`, then the four characters `\"null\"` are inserted.\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin(\"1.4\")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: CharSequence?): StringBuilder\n\n /**\n * Inserts the string representation of the specified object [value] into this string builder at the specified [index] and returns this instance.\n *\n * The overall effect is exactly as if the [value] were converted to a string by the `value.toString()` method,\n * and then that string was inserted into this string builder
at the specified [index].\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin(\"1.4\")\n
@WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: Any?): StringBuilder\n\n /**\n * Inserts the string [value] into this string builder at the specified [index] and returns this instance.\n *\n * If [value] is `null`, then the four characters `\"null\"` are inserted.\n *\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin(\"1.4\")\n
@WasExperimental(ExperimentalStdlibApi::class)\n fun insert(index: Int, value: String?): StringBuilder\n\n
/**\n * Sets the length of this string builder to the specified [newLength].\n *\n * If the [newLength] is less than the current length, it is changed to the specified [newLength].\n * Otherwise,
null characters '\\u0000' are appended to this string builder until its length is less than the [newLength].\n *\n * Note that in Kotlin/JS [set] operator function has non-constant execution time complexity.\n * Therefore, increasing length of this string builder and then updating each character by index may slow down your program.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] if [newLength] is less than zero.\n */\n @SinceKotlin(\"1.4\")\n @WasExperimental(ExperimentalStdlibApi::class)\n fun setLength(newLength: Int)\n\n /**\n * Returns a new [String] that contains characters in this string builder at [startIndex] (inclusive) and up to the [length] (exclusive).\n *\n * @throws IndexOutOfBoundsException if [startIndex] is less than zero or greater than the length of this string builder.\n */\n @SinceKotlin(\"1.4\")\n
@WasExperimental(ExperimentalStdlibApi::class)\n fun substring(startIndex:

Int): String\n\n   /**\n   * Returns a new [String] that contains characters in this string builder at [startIndex] (inclusive) and up to the [endIndex] (exclusive).\n   *\n   * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this string builder indices or when `startIndex > endIndex`.\n   */\n   @SinceKotlin(\"1.4\")\n   @WasExperimental(ExperimentalStdlibApi::class)\n   fun substring(startIndex: Int, endIndex: Int): String\n\n   /**\n   * Attempts to reduce storage used for this string builder.\n   *\n   * If the backing storage of this string builder is larger than necessary to hold its current contents,\n   * then it may be resized to become more space efficient.\n   * Calling this method may, but is not required to, affect the value of the [capacity] property.\n   */\n   @SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n   fun trimToSize()\n}\n\n\n/**\n * Clears the content of this string builder making it empty and returns this instance.\n *\n * @sample samples.text.Strings.clearStringBuilder\n */\n@SinceKotlin(\"1.3\")\npublic expect fun StringBuilder.clear(): StringBuilder\n\n/**\n * Sets the character at the specified [index] to the specified [value].\n *\n * @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect operator fun StringBuilder.set(index: Int, value: Char)\n\n/**\n * Replaces characters in the specified range of this string builder with characters in the specified string [value] and returns this instance.\n *\n * @param startIndex the beginning (inclusive) of the range to replace.\n * @param endIndex the end (exclusive) of the range to replace.\n * @param value the string to replace with.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] if [startIndex] is less than zero, greater than the length of this string builder, or `startIndex > endIndex`.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun StringBuilder.setRange(startIndex: Int, endIndex: Int, value: String): StringBuilder\n\n/**\n * Removes the character at the specified [index] from this string builder and returns this instance.\n *\n * If the `Char` at the specified [index] is part of a supplementary code point, this method does not remove the entire supplementary character.\n *\n * @param index the index of `Char` to remove.\n *\n * @throws IndexOutOfBoundsException if [index] is out of bounds of this string builder.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun StringBuilder.deleteAt(index: Int): StringBuilder\n\n/**\n * Removes characters in the specified range from this string builder and returns this instance.\n *\n * @param startIndex the beginning (inclusive) of the range to remove.\n * @param endIndex the end (exclusive) of the range to remove.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] is out of range of this string builder indices or when `startIndex > endIndex`.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun StringBuilder.deleteRange(startIndex: Int, endIndex: Int): StringBuilder\n\n/**\n * Copies characters from this string builder into the [destination] character array.\n *\n * @param destination the array to copy to.\n * @param destinationOffset the position in the array to copy to, 0 by default.\n * @param startIndex the beginning (inclusive) of the range to copy, 0 by default.\n * @param endIndex the end (exclusive) of the range to copy, length of this string builder by default.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of this string builder indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException when the subrange doesn't fit into the [destination] array starting at the specified [destinationOffset],\n * or when that index is out of the [destination] array indices range.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun StringBuilder.toCharArray(destination: CharArray, destinationOffset: Int = 0, startIndex: Int = 0, endIndex: Int = this.length)\n\n/**\n * Appends characters in a subarray of the specified character array [value] to this string builder and returns this instance.\n *\n * Characters are appended in order, starting at specified [startIndex].\n *\n * @param value the array from which characters are appended.\n * @param startIndex the beginning (inclusive) of the subarray to append.\n * @param endIndex the end (exclusive) of the subarray to append.\n *\n * @throws

IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex
 > endIndex`.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun StringBuilder.appendRange(value: CharArray, startIndex: Int, endIndex: Int): StringBuilder\n\n/**\n * Appends a subsequence of the specified character sequence [value] to this string builder and returns this instance.\n *\n * @param value the character sequence from which a subsequence is appended.\n * @param startIndex the beginning (inclusive) of the subsequence to append.\n * @param endIndex the end (exclusive) of the subsequence to append.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun StringBuilder.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder\n\n/**\n * Inserts characters in a subarray of the specified
 character array [value] into this string builder at the specified [index] and returns this instance.\n *\n * The inserted characters go in same order as in the [value] array, starting at [index].\n *\n * @param index the position in this string builder to insert at.\n * @param value the array from which characters are inserted.\n * @param startIndex the beginning (inclusive) of the subarray to insert.\n * @param endIndex the end (exclusive) of the subarray to insert.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] array indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun StringBuilder.insertRange(index: Int, value: CharArray, startIndex: Int, endIndex: Int): StringBuilder\n\n/**\n * Inserts characters
 in a subsequence of the specified character sequence [value] into this string builder at the specified [index] and returns this instance.\n *\n * The inserted characters go in the same order as in the [value] character sequence, starting at [index].\n *\n * @param index the position in this string builder to insert at.\n * @param value the character sequence from which a subsequence is inserted.\n * @param startIndex the beginning (inclusive) of the subsequence to insert.\n * @param endIndex the end (exclusive) of the subsequence to insert.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.\n * @throws IndexOutOfBoundsException if [index] is less than zero or greater than the length of this string builder.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun StringBuilder.insertRange(index: Int,
 value: CharSequence, startIndex: Int, endIndex: Int): StringBuilder\n\n@Suppress(\"EXTENSION_SHADOWED_BY_MEMBER\")\n@Deprecated(\"Use append(value: Any?) instead\", ReplaceWith(\"append(value = obj)\"), DeprecationLevel.WARNING)\n@kotlin.internal.InlineOnly\npublic inline fun StringBuilder.append(obj: Any?): StringBuilder = this.append(obj)\n\n/**\n * Builds new string by populating newly created [StringBuilder] using provided [builderAction]\n * and then converting it to [String].\n */\n@kotlin.internal.InlineOnly\npublic inline fun buildString(builderAction: StringBuilder.() -> Unit): String {\n   contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n   return StringBuilder().apply(builderAction).toString()\n}\n\n/**\n * Builds new string by populating newly created [StringBuilder] initialized with the given [capacity]\n * using provided [builderAction] and then converting it to [String].\n */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline fun buildString(capacity:
 Int, builderAction: StringBuilder.() -> Unit): String {\n   contract { callsInPlace(builderAction, InvocationKind.EXACTLY_ONCE) }\n   return StringBuilder(capacity).apply(builderAction).toString()\n}\n\n/**\n * Appends all arguments to the given StringBuilder.\n */\npublic fun StringBuilder.append(vararg value: String?): StringBuilder {\n   for (item in value)\n   append(item)\n   return this\n}\n\n/**\n * Appends all arguments to the given StringBuilder.\n

*/\npublic fun StringBuilder.append(vararg value: Any?): StringBuilder {\n for (item in value)\n append(item)\n return this\n}\n\n// KT-52336\n@Deprecated(\"Use appendRange instead.\", ReplaceWith(\"this.appendRange(str, offset, offset + len)\"), level = DeprecationLevel.ERROR)\n@kotlin.internal.InlineOnly\n@Suppress(\"UNUSED_PARAMETER\")\npublic inline fun StringBuilder.append(str: CharArray, offset: Int, len: Int): StringBuilder = throw NotImplementedError()\n\n/** Appends a line feed character (`\\n`) to this StringBuilder. */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun StringBuilder.appendLine(): StringBuilder = append('\\n')\n\n/** Appends [value] to this [StringBuilder], followed by a line feed character (`\\n`). */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun StringBuilder.appendLine(value: CharSequence?): StringBuilder = append(value).appendLine()\n\n/** Appends [value] to this [StringBuilder], followed by a line feed character (`\\n`). */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun StringBuilder.appendLine(value: String?): StringBuilder = append(value).appendLine()\n\n/** Appends [value] to this [StringBuilder], followed by a line feed character (`\\n`). */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun StringBuilder.appendLine(value: Any?): StringBuilder = append(value).appendLine()\n\n/** Appends [value] to this [StringBuilder], followed by a line feed character (`\\n`). */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun StringBuilder.appendLine(value: CharArray): StringBuilder = append(value).appendLine()\n\n/** Appends [value] to this [StringBuilder], followed by a line feed character (`\\n`). */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun StringBuilder.appendLine(value: Char): StringBuilder = append(value).appendLine()\n\n/** Appends [value] to this [StringBuilder], followed by a line feed character (`\\n`). */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun StringBuilder.appendLine(value: Boolean): StringBuilder = append(value).appendLine()\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\nimport kotlin.js.RegExp\n\n@kotlin.internal.InlineOnly\ninternal actual inline fun String.nativeIndexOf(ch: Char, fromIndex: Int): Int = nativeIndexOf(ch.toString(), fromIndex)\n\n@kotlin.internal.InlineOnly\ninternal actual inline fun String.nativeLastIndexOf(ch: Char, fromIndex: Int): Int = nativeLastIndexOf(ch.toString(), fromIndex)\n\n/**\n * Returns `true` if this string starts with the specified prefix.\n */\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun String.startsWith(prefix: String, ignoreCase: Boolean = false): Boolean {\n if (!ignoreCase)\n return nativeStartsWith(prefix, 0)\n else\n return regionMatches(0, prefix, 0, prefix.length, ignoreCase)\n}\n\n/**\n * Returns `true` if a substring of this string starting at the specified offset [startIndex] starts with the specified prefix.\n */\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun String.startsWith(prefix: String, startIndex: Int, ignoreCase: Boolean = false): Boolean {\n if (!ignoreCase)\n return nativeStartsWith(prefix, startIndex)\n else\n return regionMatches(startIndex, prefix, 0, prefix.length, ignoreCase)\n}\n\n/**\n * Returns `true` if this string ends with the specified suffix.\n */\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun String.endsWith(suffix: String, ignoreCase: Boolean = false): Boolean {\n if (!ignoreCase)\n return nativeEndsWith(suffix)\n else\n return regionMatches(length - suffix.length, suffix, 0, suffix.length, ignoreCase)\n}\n\n@Deprecated(\"Use Regex.matches() instead\", ReplaceWith(\"regex.toRegex().matches(this)\"))\n@DeprecatedSinceKotlin(warningSince = \"1.6\")\npublic fun String.matches(regex: String): Boolean {\n @Suppress(\"DEPRECATION\")\n val result = this.match(regex)\n return result != null && result.size != 0\n}\n\n/**\n * Returns `true` if this string is empty or consists solely of whitespace characters.\n *\n * @sample samples.text.Strings.stringIsBlank\n */\npublic actual fun CharSequence.isBlank():

```
Boolean = length == 0 || indices.all { this[it].isWhitespace() }\n\n/**\n * Returns `true` if this string is equal to [other], optionally ignoring character case.\n *\n * Two strings are considered to be equal if they have the same length and the same character at the same index.\n * If [ignoreCase] is true, the result of `Char.uppercaseChar().lowercaseChar()` on each character is compared.\n *\n * @param ignoreCase `true` to ignore character case when comparing strings. By default `false`.\n */\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun String?.equals(other: String?, ignoreCase: Boolean = false): Boolean {\n    if (this == null) return other == null\n    if (other == null) return false\n    if (!ignoreCase) return this == other\n\n    if (this.length != other.length) return false\n\n    for (index in 0 until this.length) {\n        val thisChar = this[index]\n        val otherChar = other[index]\n        if (!thisChar.equals(otherChar, ignoreCase)) {\n            return false\n        }\n    }\n\n    return true\n}\n\n\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun CharSequence.regionMatches(thisOffset: Int, other: CharSequence, otherOffset: Int, length: Int, ignoreCase: Boolean = false): Boolean =\n    regionMatchesImpl(thisOffset, other, otherOffset, length, ignoreCase)\n\n\n/**\n * Returns a copy of this string having its first letter titlecased using the rules of the default locale,\n * or the original string if it's empty or already starts with a title case letter.\n *\n * The title case of a character is usually the same as its upper case with several exceptions.\n * The particular list of characters with the special title case form depends on the underlying platform.\n *\n * @sample samples.text.Strings.capitalize\n */\n@Deprecated(\"Use replaceFirstChar instead.\", ReplaceWith(\"replaceFirstChar { if (it.isLowerCase()) it.titlecase() else it.toString() }\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic actual fun String.capitalize(): String {\n    return if (isNotEmpty()) substring(0, 1).uppercase() + substring(1) else this\n}\n\n/**\n * Returns a copy of this string having its first letter lowercased using the rules of the default locale,\n * or the original string if it's empty or already starts with a lower case letter.\n *\n * @sample samples.text.Strings.decapitalize\n */\n@Deprecated(\"Use replaceFirstChar instead.\", ReplaceWith(\"replaceFirstChar { it.lowercase() }\"))\n@DeprecatedSinceKotlin(warningSince = \"1.5\")\npublic actual fun String.decapitalize(): String {\n    return if (isNotEmpty()) substring(0, 1).lowercase() + substring(1) else this\n}\n\n/**\n * Returns a string containing this char sequence repeated [n] times.\n * @throws [IllegalArgumentException] when n < 0.\n * @sample samples.text.Strings.repeat\n */\npublic actual fun CharSequence.repeat(n: Int): String {\n    require(n >= 0) { \"Count 'n' must be non-negative, but was $n.\" }\n    return when (n) {\n        0 -> \"\"\n        1 -> this.toString()\n        else -> {\n            var result = \"\"\n            if (!isEmpty()) {\n                var s = this.toString()\n                var count = n\n                while (true) {\n                    if ((count and 1) == 1) {\n                        result += s\n                    }\n                    count = count ushr 1\n                    if (count == 0) {\n                        break\n                    }\n                    s += s\n                }\n            }\n            return result\n        }\n    }\n}\n\n/**\n * Returns a new string obtained by replacing all occurrences of the [oldValue] substring in this string\n * with the specified [newValue] string.\n *\n * @sample samples.text.Strings.replace\n */\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun String.replace(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =\n    nativeReplace(RegExp(Regex.escape(oldValue),\n        if (ignoreCase) \"gui\" else \"gu\"), Regex.nativeEscapeReplacement(newValue))\n\n/**\n * Returns a new string with all occurrences of [oldChar] replaced with [newChar].\n *\n * @sample samples.text.Strings.replace\n */\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun String.replace(oldChar: Char, newChar: Char, ignoreCase: Boolean = false): String =\n    nativeReplace(RegExp(Regex.escape(oldChar.toString()), if (ignoreCase) \"gui\" else \"gu\"),\n        newChar.toString())\n\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGUMENTS\")\npublic actual fun String.replaceFirst(oldValue: String, newValue: String, ignoreCase: Boolean = false): String =\n    nativeReplace(RegExp(Regex.escape(oldValue), if (ignoreCase) \"ui\" else \"u\"),\n        Regex.nativeEscapeReplacement(newValue))\n\n@Suppress(\"ACTUAL_FUNCTION_WITH_DEFAULT_ARGU
```

MENTS\")\npublic actual fun String.replaceFirst(oldChar: Char, newChar: Char, ignoreCase: Boolean = false): String =\n   nativeReplace(RegExp(Regex.escape(oldChar.toString()),
 if (ignoreCase) \"ui\" else \"u\"), newChar.toString())\n","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/** Returns the negative [size] if [throwOnMalformed] is false, throws [CharacterCodingException] otherwise. */\nprivate fun malformed(size: Int, index: Int, throwOnMalformed: Boolean): Int {\n   if (throwOnMalformed) throw CharacterCodingException(\"Malformed sequence starting at ${index - 1}\")\n   return -size\n}\n\n/**\n * Returns code point corresponding to UTF-16 surrogate pair,\n * where the first of the pair is the [high] and the second is in the [string] at the [index].\n * Returns zero if the pair is malformed and [throwOnMalformed] is false.\n *\n * @throws CharacterCodingException if the pair is malformed and [throwOnMalformed] is true.\n */\nprivate fun codePointFromSurrogate(string: String, high: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {\n   if (high !in 0xD800..0xDBFF || index >= endIndex) {\n   return malformed(0, index, throwOnMalformed)\n   }\n   val low = string[index].code\n   if (low !in 0xDC00..0xDFFF) {\n   return malformed(0, index, throwOnMalformed)\n   }\n   return 0x10000 + ((high and 0x3FF) shl 10) or (low and 0x3FF)\n}\n\n/**\n * Returns code point corresponding to UTF-8 sequence of two bytes,\n * where the first byte of the sequence is the [byte1] and the second byte is in the [bytes] array at the [index].\n * Returns zero if the sequence is malformed and [throwOnMalformed] is false.\n *\n * @throws CharacterCodingException if the sequence of two bytes is malformed and [throwOnMalformed] is true.\n */\nprivate fun codePointFrom2(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {\n   if (byte1 and 0x1E == 0 || index >= endIndex) {\n   return malformed(0, index, throwOnMalformed)\n   }\n   val byte2 = bytes[index].toInt()\n   if (byte2 and 0xC0 != 0x80) {\n   return malformed(0, index, throwOnMalformed)\n   }\n   return (byte1 shl 6) xor byte2 xor 0xF80\n}\n\n/**\n * Returns code point corresponding to UTF-8 sequence of three bytes,\n * where the first byte of the sequence is the [byte1] and the others are in the [bytes] array starting from the [index].\n * Returns a non-positive value indicating number of bytes from [bytes] included in malformed sequence\n * if the sequence is malformed and [throwOnMalformed] is false.\n *\n * @throws CharacterCodingException if the sequence of three bytes is malformed and [throwOnMalformed] is true.\n */\nprivate fun codePointFrom3(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {\n   if (index >= endIndex) {\n   return malformed(0, index, throwOnMalformed)\n   }\n\n   val byte2 = bytes[index].toInt()\n   if (byte1 and 0xF == 0) {\n   if (byte2 and 0xE0 != 0xA0) {\n   // Non-shortest form\n   return malformed(0, index, throwOnMalformed)\n   }\n   } else if (byte1 and 0xF == 0xD) {\n   if (byte2 and 0xE0 != 0x80) {\n   // Surrogate code point\n   return malformed(0, index, throwOnMalformed)\n   }\n   } else if (byte2 and 0xC0 != 0x80) {\n   return malformed(0, index, throwOnMalformed)\n   }\n\n   if (index + 1 == endIndex) {\n   return malformed(1, index, throwOnMalformed)\n   }\n   val byte3 = bytes[index + 1].toInt()\n   if (byte3 and 0xC0 != 0x80) {\n   return malformed(1, index, throwOnMalformed)\n   }\n\n   return (byte1 shl 12) xor (byte2 shl 6) xor byte3 xor -0x1E080\n}\n\n/**\n * Returns code point corresponding to UTF-8 sequence of four bytes,\n * where the first byte of the sequence is the [byte1] and the others are in the [bytes] array starting from the [index].\n * Returns a non-positive value indicating
 number of bytes from [bytes] included in malformed sequence\n * if the sequence is malformed and [throwOnMalformed] is false.\n *\n * @throws CharacterCodingException if the sequence of four bytes is malformed and [throwOnMalformed] is true.\n */\nprivate fun codePointFrom4(bytes: ByteArray, byte1: Int, index: Int, endIndex: Int, throwOnMalformed: Boolean): Int {\n   if (index >= endIndex) {\n   malformed(0, index, throwOnMalformed)\n   }\n\n   val byte2 = bytes[index].toInt()\n   if (byte1 and 0xF == 0x0) {\n   if (byte2 and 0xF0 <= 0x80) {\n   // Non-shortest form\n   return malformed(0, index, throwOnMalformed)\n   }\n   } else if (byte1 and 0xF == 0x4) {\n   if (byte2 and 0xF0 != 0x80) {\n   // Out of Unicode code points domain (larger than U+10FFFF)\n   return malformed(0, index, throwOnMalformed)\n   }\n   } else if (byte1 and 0xF > 0x4) {\n   return malformed(0, index, throwOnMalformed)\n   } else if (byte2

```
and 0xC0 != 0x80) {\n        return malformed(0, index, throwOnMalformed)\n    }\n\n    if (index + 1 == endIndex) {\n        return malformed(1, index, throwOnMalformed)\n    }\n    val byte3 = bytes[index + 1].toInt()\n    if (byte3 and 0xC0 != 0x80) {\n        return malformed(1, index, throwOnMalformed)\n    }\n\n    if (index + 2 == endIndex) {\n        return malformed(2, index, throwOnMalformed)\n    }\n    val byte4 = bytes[index + 2].toInt()\n    if (byte4 and 0xC0 != 0x80) {\n        return malformed(2, index, throwOnMalformed)\n    }\n    return (byte1 shl 18) xor (byte2 shl 12) xor (byte3 shl 6) xor byte4 xor 0x381F80\n}\n\n/**\n * Maximum number of bytes needed to encode a single char.\n *\n * Code points in `0..0x7F` are encoded in a single byte.\n * Code points in `0x80..0x7FF` are encoded in two bytes.\n * Code points in `0x800..0xD7FF` or in `0xE000..0xFFFF` are encoded in three bytes.\n * Surrogate code points in `0xD800..0xDFFF` are not Unicode scalar values, therefore
 aren't encoded.\n * Code points in `0x10000..0x10FFFF` are represented by a pair of surrogate `Char`s and are encoded in four bytes.\n */\nprivate const val MAX_BYTES_PER_CHAR = 3\n\n/**\n * The byte sequence a malformed UTF-16 char sequence is replaced by.\n */\nprivate val REPLACEMENT_BYTE_SEQUENCE: ByteArray = byteArrayOf(0xEF.toByte(), 0xBF.toByte(), 0xBD.toByte())\n\n/**\n * Encodes the [string] using UTF-8 and returns the resulting [ByteArray].\n *\n * @param string the string to encode.\n * @param startIndex the start offset (inclusive) of the substring to encode.\n * @param endIndex the end offset (exclusive) of the substring to encode.\n * @param throwOnMalformed whether to throw on malformed char sequence or replace by the [REPLACEMENT_BYTE_SEQUENCE].\n *\n * @throws CharacterCodingException if the char sequence is malformed and [throwOnMalformed] is true.\n */\ninternal fun encodeUtf8(string: String, startIndex: Int, endIndex: Int, throwOnMalformed: Boolean): ByteArray
 {\n    require(startIndex >= 0 && endIndex <= string.length && startIndex <= endIndex)\n\n    val bytes = ByteArray((endIndex - startIndex) * MAX_BYTES_PER_CHAR)\n    var byteIndex = 0\n    var charIndex = startIndex\n\n    while (charIndex < endIndex) {\n        val code = string[charIndex++].code\n        when {\n            code < 0x80 ->\n                bytes[byteIndex++] = code.toByte()\n            code < 0x800 -> {\n                bytes[byteIndex++] = ((code shr 6) or 0xC0).toByte()\n                bytes[byteIndex++] = ((code and 0x3F) or 0x80).toByte()\n            }\n            code < 0xD800 || code >= 0xE000 -> {\n                bytes[byteIndex++] = ((code shr 12) or 0xE0).toByte()\n                bytes[byteIndex++] = (((code shr 6) and 0x3F) or 0x80).toByte()\n                bytes[byteIndex++] = ((code and 0x3F) or 0x80).toByte()\n            }\n            else -> { // Surrogate char value\n                val codePoint = codePointFromSurrogate(string, code, charIndex,
 endIndex, throwOnMalformed)\n                if (codePoint <= 0) {\n                    bytes[byteIndex++] = REPLACEMENT_BYTE_SEQUENCE[0]\n                    bytes[byteIndex++] = REPLACEMENT_BYTE_SEQUENCE[1]\n                    bytes[byteIndex++] = REPLACEMENT_BYTE_SEQUENCE[2]\n                } else {\n                    bytes[byteIndex++] = ((codePoint shr 18) or 0xF0).toByte()\n                    bytes[byteIndex++] = (((codePoint shr 12) and 0x3F) or 0x80).toByte()\n                    bytes[byteIndex++] = (((codePoint shr 6) and 0x3F) or 0x80).toByte()\n                    bytes[byteIndex++] = ((codePoint and 0x3F) or 0x80).toByte()\n                    charIndex++\n                }\n            }\n        }\n    }\n\n    return if (bytes.size == byteIndex) bytes else bytes.copyOf(byteIndex)\n}\n\n/**\n * The character a malformed UTF-8 byte sequence is replaced by.\n */\nprivate const val REPLACEMENT_CHAR = '\\uFFFD'\n\n/**\n * Decodes the UTF-8 [bytes] array and returns
 the resulting [String].\n *\n * @param bytes the byte array to decode.\n * @param startIndex the start offset (inclusive) of the array to be decoded.\n * @param endIndex the end offset (exclusive) of the array to be encoded.\n * @param throwOnMalformed whether to throw on malformed byte sequence or replace by the [REPLACEMENT_CHAR].\n *\n * @throws CharacterCodingException if the array is malformed UTF-8 byte sequence and [throwOnMalformed] is true.\n */\ninternal fun decodeUtf8(bytes: ByteArray, startIndex: Int, endIndex: Int, throwOnMalformed: Boolean): String {\n    require(startIndex >= 0 && endIndex <= bytes.size && startIndex <= endIndex)\n\n    var byteIndex = startIndex\n    val stringBuilder = StringBuilder()\n\n    while (byteIndex < endIndex) {\n        val byte = bytes[byteIndex++].toInt()\n        when {\n            byte >= 0 ->\n                stringBuilder.append(byte.toChar())\n            byte shr 5 == -2 -> {\n                val code = codePointFrom2(bytes, byte,
```

```
 byteIndex, endIndex, throwOnMalformed)\n                if (code <= 0) {\n
stringBuilder.append(REPLACEMENT_CHAR)\n                    byteIndex += -code\n                } else {\n
stringBuilder.append(code.toChar())\n                    byteIndex += 1\n                }\n            }\n            byte shr 4 == -2 -
> {\n                val code = codePointFrom3(bytes, byte, byteIndex, endIndex, throwOnMalformed)\n                if
(code <= 0) {\n                    stringBuilder.append(REPLACEMENT_CHAR)\n                    byteIndex += -code\n
        } else {\n                    stringBuilder.append(code.toChar())\n                    byteIndex += 2\n                }\n
}\n            byte shr 3 == -2 -> {\n                val code = codePointFrom4(bytes, byte, byteIndex, endIndex,
throwOnMalformed)\n                if (code <= 0) {\n                    stringBuilder.append(REPLACEMENT_CHAR)\n
            byteIndex +=
 -code\n                } else {\n                    val high = (code - 0x10000) shr 10 or 0xD800\n                    val low = (code
and 0x3FF) or 0xDC00\n                    stringBuilder.append(high.toChar())\n
stringBuilder.append(low.toChar())\n                    byteIndex += 3\n                }\n            }\n            else -> {\n
  malformed(0, byteIndex, throwOnMalformed)\n                stringBuilder.append(REPLACEMENT_CHAR)\n
 }\n        }\n    }\n\n    return stringBuilder.toString()\n}","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be
found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\n/**\n * Returns the detailed description of this
throwable with its stack trace.\n *\n * The detailed description includes:\n * - the short description (see
[Throwable.toString]) of this throwable;\n * - the complete
 stack trace;\n * - detailed descriptions of the exceptions that were [suppressed][suppressedExceptions] in order to
deliver this exception;\n * - the detailed description of each throwable in the [Throwable.cause] chain.\n
*/\n@SinceKotlin(\"1.4\")\npublic actual fun Throwable.stackTraceToString(): String =
ExceptionTraceBuilder().buildFor(this)\n\n/**\n * Prints the [detailed description][Throwable.stackTraceToString]
of this throwable to console error output.\n */\n@SinceKotlin(\"1.4\")\npublic actual fun
Throwable.printStackTrace() {\n    console.error(this.stackTraceToString())\n}\n\n/**\n * Adds the specified
exception to the list of exceptions that were\n * suppressed in order to deliver this exception.\n
*/\n@SinceKotlin(\"1.4\")\npublic actual fun Throwable.addSuppressed(exception: Throwable) {\n    if (this !==
exception) {\n        val suppressed = this.asDynamic()._suppressed.unsafeCast<MutableList<Throwable>?>()\n
if (suppressed == null) {\n            this.asDynamic()._suppressed
 = mutableListOf(exception)\n        } else {\n            suppressed.add(exception)\n        }\n    }\n}\n\n/**\n * Returns
a list of all exceptions that were suppressed in order to deliver this exception.\n */\n@SinceKotlin(\"1.4\")\npublic
actual val Throwable.suppressedExceptions: List<Throwable>\n    get() {\n        return
this.asDynamic()._suppressed?.unsafeCast<List<Throwable>>() ?: emptyList()\n    }\n\n\nprivate class
ExceptionTraceBuilder {\n    private val target = StringBuilder()\n    private val visited = arrayOf<Throwable>()\n
private var topStack: String = \"\"\n    private var topStackStart: Int = 0\n\n    fun buildFor(exception: Throwable):
String {\n        exception.dumpFullTrace(\"\", \"\")\n        return target.toString()\n    }\n\n    private fun
hasSeen(exception: Throwable): Boolean = visited.any { it === exception }\n\n    private fun
Throwable.dumpFullTrace(indent: String, qualifier: String) {\n        this.dumpSelfTrace(indent, qualifier)
 || return\n\n        var cause = this.cause\n        while (cause != null) {\n            cause.dumpSelfTrace(indent, \"Caused
by: \") || return\n            cause = cause.cause\n        }\n    }\n\n    private fun Throwable.dumpSelfTrace(indent:
String, qualifier: String): Boolean {\n        target.append(indent).append(qualifier)\n        val shortInfo =
this.toString()\n        if (hasSeen(this)) {\n            target.append(\"[CIRCULAR REFERENCE, SEE ABOVE:
\").append(shortInfo).append(\"]\\n\")\n            return false\n        }\n        visited.asDynamic().push(this)\n\n        var
stack = this.asDynamic().stack as String?\n        if (stack != null) {\n            val stackStart =
stack.indexOf(shortInfo).let { if (it < 0) 0 else it + shortInfo.length }\n            if (stackStart == 0)
target.append(shortInfo).append(\"\\n\")\n            if (topStack.isEmpty()) {\n                topStack = stack\n
topStackStart = stackStart\n            } else {\n
        stack = dropCommonFrames(stack, stackStart)\n            }\n            if (indent.isNotEmpty()) {\n                //
indent stack, but avoid indenting exception message lines\n                val messageLines = if (stackStart == 0) 0 else
```

1 + shortInfo.count { c -> c == '\\n' }\n stack.lineSequence().forEachIndexed { index: Int, line: String ->\n if (index >= messageLines) target.append(indent)\n target.append(line).append(\"\\n\")\n }\n } else {\n target.append(stack).append(\"\\n\")\n }\n } else {\n target.append(shortInfo).append(\"\\n\")\n }\n\n val suppressed = suppressedExceptions\n if (suppressed.isNotEmpty()) {\n val suppressedIndent = indent + \" \"\n for (s in suppressed) {\n s.dumpFullTrace(suppressedIndent, \"Suppressed: \")\n }\n }\n return true\n }\n\n private fun dropCommonFrames(stack: String, stackStart: Int): String {\n var commonFrames: Int = 0\n var lastBreak: Int = 0\n var preLastBreak: Int = 0\n for (pos in 0 until minOf(topStack.length - topStackStart, stack.length - stackStart)) {\n val c = stack[stack.lastIndex - pos]\n if (c != topStack[topStack.lastIndex - pos]) break\n if (c == '\\n') {\n commonFrames += 1\n preLastBreak = lastBreak\n lastBreak = pos\n }\n }\n if (commonFrames <= 1) return stack\n while (preLastBreak > 0 && stack[stack.lastIndex - (preLastBreak - 1)] == ' ')\n preLastBreak -= 1\n\n // leave 1 common frame to ease matching with the top exception stack\n return stack.dropLast(preLastBreak) + \"... and ${commonFrames - 1} more common stack frames skipped\"\n }\n}","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport kotlin.js.json\nimport kotlin.math.*\n\ninternal actual inline val durationAssertionsEnabled: Boolean get() = true\n\ninternal actual fun formatToExactDecimals(value: Double, decimals: Int): String {\n val rounded = if (decimals == 0) {\n value\n } else {\n val pow = 10.0.pow(decimals)\n JsMath.round(abs(value) * pow) / pow * sign(value)\n }\n return if (abs(rounded) < 1e21) {\n // toFixed switches to scientific format after 1e21\n rounded.asDynamic().toFixed(decimals).unsafeCast<String>()\n } else {\n // toPrecision outputs the specified number of digits, but only for positive numbers\n val positive = abs(rounded)\n val positiveString = positive.asDynamic().toPrecision(ceil(log10(positive)) + decimals).unsafeCast<String>()\n if (rounded < 0) \"-$positiveString\" else positiveString\n }\n}\n\ninternal actual fun formatUpToDecimals(value: Double, decimals: Int): String {\n return value.asDynamic().toLocaleString(\"en-us\", json(\"maximumFractionDigits\" to decimals)).unsafeCast<String>()\n}\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalTime::class)\npublic actual enum class DurationUnit(internal val scale: Double) {\n /**\n * Time unit representing one nanosecond, which is 1/1000 of a microsecond.\n */\n NANOSECONDS(1e0),\n /**\n * Time unit representing one microsecond, which is 1/1000 of a millisecond.\n */\n MICROSECONDS(1e3),\n /**\n * Time unit representing one millisecond, which is 1/1000 of a second.\n */\n MILLISECONDS(1e6),\n /**\n * Time unit representing one second.\n */\n SECONDS(1e9),\n /**\n * Time unit representing one minute.\n */\n MINUTES(60e9),\n /**\n * Time unit representing one hour.\n */\n HOURS(3600e9),\n /**\n * Time unit representing one day, which is always equal to 24 hours.\n */\n DAYS(86400e9);\n}\n\n@SinceKotlin(\"1.3\")\ninternal actual fun convertDurationUnit(value: Double, sourceUnit: DurationUnit, targetUnit: DurationUnit): Double {\n val sourceCompareTarget = sourceUnit.scale.compareTo(targetUnit.scale)\n return when {\n sourceCompareTarget > 0 -> value * (sourceUnit.scale / targetUnit.scale)\n sourceCompareTarget < 0 -> value / (targetUnit.scale / sourceUnit.scale)\n else -> value\n }\n}\n\n@SinceKotlin(\"1.5\")\ninternal actual fun convertDurationUnitOverflow(value: Long, sourceUnit: DurationUnit, targetUnit: DurationUnit): Long {\n val sourceCompareTarget = sourceUnit.scale.compareTo(targetUnit.scale)\n return when {\n sourceCompareTarget > 0 -> value * (sourceUnit.scale / targetUnit.scale).toLong()\n sourceCompareTarget < 0 -> value / (targetUnit.scale / sourceUnit.scale).toLong()\n else -> value\n }\n}\n\n@SinceKotlin(\"1.5\")\ninternal actual fun convertDurationUnit(value: Long, sourceUnit: DurationUnit, targetUnit: DurationUnit): Long {\n val sourceCompareTarget = sourceUnit.scale.compareTo(targetUnit.scale)\n

```
return when {\n        sourceCompareTarget > 0 -> {\n            val scale = (sourceUnit.scale / targetUnit.scale).toLong()\n            val result = value * scale\n            when {\n                result / scale == value -> result\n                value > 0 -> Long.MAX_VALUE\n                else -> Long.MIN_VALUE\n            }\n        }\n        sourceCompareTarget < 0 -> value / (targetUnit.scale / sourceUnit.scale).toLong()\n        else -> value\n    }\n}\n\n\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport org.w3c.performance.GlobalPerformance\nimport org.w3c.performance.Performance\nimport kotlin.math.truncate\nimport kotlin.time.Duration.Companion.milliseconds\nimport kotlin.time.TimeSource.Monotonic.ValueTimeMark\n\n@Suppress(\"ACTUAL_WITHOUT_EXPECT\") // visibility\ninternal actual typealias ValueTimeMarkReading = Any\n\n@ExperimentalTime\ninternal interface DefaultTimeSource : TimeSource.WithComparableMarks {\n    override fun markNow(): ValueTimeMark\n    fun elapsedFrom(timeMark: ValueTimeMark): Duration\n    fun differenceBetween(one: ValueTimeMark, another: ValueTimeMark): Duration\n    fun adjustReading(timeMark: ValueTimeMark, duration: Duration): ValueTimeMark\n}\n\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\ninternal actual object MonotonicTimeSource : DefaultTimeSource, TimeSource.WithComparableMarks {  // TODO: interface should not be required here\n\n    private val actualSource: DefaultTimeSource = run {\n        val isNode: Boolean = js(\"typeof process !== 'undefined' && process.versions && !!process.versions.node\")\n\n        if (isNode)\n            HrTimeSource(js(\"process\").unsafeCast<Process>())\n        else\n            js(\"typeof self !== 'undefined' ? self : globalThis\")\n                .unsafeCast<GlobalPerformance?>()\n                ?.performance\n                ?.let(::PerformanceTimeSource)\n                ?: DateNowTimeSource\n    }\n\n    actual override fun markNow(): ValueTimeMark = actualSource.markNow()\n    actual override fun elapsedFrom(timeMark: ValueTimeMark): Duration = actualSource.elapsedFrom(timeMark)\n    actual override fun differenceBetween(one: ValueTimeMark, another: ValueTimeMark): Duration = actualSource.differenceBetween(one, another)\n\n    actual override fun adjustReading(timeMark: ValueTimeMark, duration: Duration): ValueTimeMark =\n        actualSource.adjustReading(timeMark, duration)\n}\n\ninternal external interface Process {\n    fun hrtime(time: Array<Double> = definedExternally): Array<Double>\n}\n\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\ninternal class HrTimeSource(private val process: Process) : DefaultTimeSource {\n\n    override fun markNow(): ValueTimeMark = ValueTimeMark(process.hrtime())\n    override fun elapsedFrom(timeMark: ValueTimeMark): Duration =\n        @Suppress(\"UNCHECKED_CAST\")\n        process.hrtime(timeMark.reading as Array<Double>)\n            .let { (seconds, nanos) -> seconds.toDuration(DurationUnit.SECONDS) + nanos.toDuration(DurationUnit.NANOSECONDS) }\n\n    @Suppress(\"UNCHECKED_CAST\")\n    override fun differenceBetween(one: ValueTimeMark, another: ValueTimeMark): Duration {\n        val (s1, n1) = one.reading as Array<Double>\n        val (s2, n2) = another.reading as Array<Double>\n        return (if (s1 == s2 && n1 == n2) Duration.ZERO else (s1 - s2).toDuration(DurationUnit.SECONDS))\n            + (n1 - n2).toDuration(DurationUnit.NANOSECONDS)\n    }\n\n    override fun adjustReading(timeMark: ValueTimeMark, duration: Duration): ValueTimeMark =\n        @Suppress(\"UNCHECKED_CAST\")\n        (timeMark.reading as Array<Double>).let { (seconds, nanos) ->\n            duration.toComponents { _, addNanos ->\n                val resultSeconds = sumCheckNaN(seconds + truncate(duration.toDouble(DurationUnit.SECONDS)))\n                arrayOf<Double>(resultSeconds, if (resultSeconds.isFinite()) nanos + addNanos else 0.0)\n            }\n        }.let(TimeSource.Monotonic::ValueTimeMark)\n\n\n    override fun toString(): String = \"TimeSource(process.hrtime())\"\n}\n\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\ninternal class PerformanceTimeSource(val performance: Performance) :\n    DefaultTimeSource { // AbstractDoubleTimeSource(unit = DurationUnit.MILLISECONDS) {\n    private fun read(): Double = performance.now()\n\n    override fun markNow(): ValueTimeMark = ValueTimeMark(read())\n    override fun elapsedFrom(timeMark: ValueTimeMark): Duration = (read() - timeMark.reading as Double).milliseconds\n\n    override fun differenceBetween(one: ValueTimeMark, another:
```

```
ValueTimeMark): Duration {\n        val ms1 = one.reading as Double\n        val ms2 = another.reading as Double\n
  return if (ms1 == ms2) Duration.ZERO else (ms1 - ms2).milliseconds\n    }\n\n    override fun
adjustReading(timeMark: ValueTimeMark, duration: Duration): ValueTimeMark =\n
ValueTimeMark(sumCheckNaN(timeMark.reading as Double +
duration.toDouble(DurationUnit.MILLISECONDS)))\n\n    override fun toString(): String =
\"TimeSource(self.performance.now())\"\n}\n\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\ninternal object
DateNowTimeSource : DefaultTimeSource {\n    private fun read(): Double = kotlin.js.Date.now()\n\n    override
fun markNow(): ValueTimeMark = ValueTimeMark(read())\n    override fun elapsedFrom(timeMark:
ValueTimeMark): Duration = (read()
 - timeMark.reading as Double).milliseconds\n\n    override fun differenceBetween(one: ValueTimeMark, another:
ValueTimeMark): Duration {\n        val ms1 = one.reading as Double\n        val ms2 = another.reading as Double\n
  return if (ms1 == ms2) Duration.ZERO else (ms1 - ms2).milliseconds\n    }\n\n    override fun
adjustReading(timeMark: ValueTimeMark, duration: Duration): ValueTimeMark =\n
ValueTimeMark(sumCheckNaN(timeMark.reading as Double +
duration.toDouble(DurationUnit.MILLISECONDS)))\n\n    override fun toString(): String =
\"TimeSource(Date.now())\"\n}\n\nprivate fun sumCheckNaN(value: Double): Double = value.also { if (it.isNaN())
throw IllegalArgumentException(\"Summing infinities of different signs\") }","/*\n * Copyright 2010-2020
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlinx.dom\n\nimport
org.w3c.dom.*\nimport
 kotlin.contracts.*\n\n/**\n * Creates a new element with the specified [name].\n *\n * The element is initialized
with the specified [init] function.\n */\n@SinceKotlin(\"1.4\")\npublic fun Document.createElement(name: String,
init: Element.() -> Unit): Element {\n    contract { callsInPlace(init, InvocationKind.EXACTLY_ONCE) }\n
return createElement(name).apply(init)\n}\n\n/**\n * Appends a newly created element with the specified [name] to
this element.\n *\n * The element is initialized with the specified [init] function.\n
*/\n@SinceKotlin(\"1.4\")\npublic fun Element.appendElement(name: String, init: Element.() -> Unit): Element {\n
 contract { callsInPlace(init, InvocationKind.EXACTLY_ONCE) }\n    return
ownerDocument!!.createElement(name, init).also { appendChild(it) }\n}\n\n","/*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt
 file.\n */\n\npackage kotlinx.dom\n\nimport org.w3c.dom.*\n\n/** Returns true if the element has the given CSS
class style in its 'class' attribute */\n@SinceKotlin(\"1.4\")\nfun Element.hasClass(cssClass: String): Boolean =
className.matches(\"\"\"(^|.*\\s+)$cssClass($|\\s+.*)\"\"\".toRegex())\n\n/**\n * Adds CSS class to element. Has no
effect if all specified classes are already in class attribute of the element\n *\n * @return true if at least one class has
been added\n */\n@SinceKotlin(\"1.4\")\nfun Element.addClass(vararg cssClasses: String): Boolean {\n    val
missingClasses = cssClasses.filterNot { hasClass(it) }\n    if (missingClasses.isNotEmpty()) {\n        val
presentClasses = className.trim()\n        className = buildString {\n            append(presentClasses)\n            if
(!presentClasses.isEmpty()) {\n                append(\" \")\n            }\n            missingClasses.joinTo(this, \" \")\n
}\n        return true\n    }\n\n    return false\n}\n\n/**\n
 * Removes all [cssClasses] from element. Has no effect if all specified classes are missing in class attribute of the
element\n *\n * @return true if at least one class has been removed\n */\n@SinceKotlin(\"1.4\")\nfun
Element.removeClass(vararg cssClasses: String): Boolean {\n    if (cssClasses.any { hasClass(it) }) {\n        val
toBeRemoved = cssClasses.toSet()\n        className = className.trim().split(\"\\\\s+\".toRegex()).filter { it !in
toBeRemoved }.joinToString(\" \")\n        return true\n    }\n\n    return false\n}\n","/*\n * Copyright 2010-2018
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"StringsKt\")\n\npackage
kotlin.text\n\n/**\n * Converts the string into a regular expression [Regex] with the default options.\n
```

*/\n@kotlin.internal.InlineOnly\npublic inline
fun String.toRegex(): Regex = Regex(this)\n\n/**\n * Converts the string into a regular expression [Regex] with the specified single [option].\n */\n@kotlin.internal.InlineOnly\npublic inline fun String.toRegex(option: RegexOption): Regex = Regex(this, option)\n\n/**\n * Converts the string into a regular expression [Regex] with the specified set of [options].\n */\n@kotlin.internal.InlineOnly\npublic inline fun String.toRegex(options: Set<RegexOption>): Regex = Regex(this, options)\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlinx.dom\n\nimport org.w3c.dom.*\n\n/**\n * Gets a value indicating whether this node is a TEXT_NODE or a CDATA_SECTION_NODE.\n */\n@SinceKotlin(\"1.4\")\npublic val Node.isText: Boolean\n get() = nodeType == Node.TEXT_NODE || nodeType == Node.CDATA_SECTION_NODE\n\n/**\n *
Gets a value indicating whether this node is an [Element].\n */\n@SinceKotlin(\"1.4\")\npublic val Node.isElement: Boolean\n get() = nodeType == Node.ELEMENT_NODE\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlinx.dom\n\nimport org.w3c.dom.*\n\n/** Removes all the children from this node. */\n@SinceKotlin(\"1.4\")\npublic fun Node.clear() {\n while (hasChildNodes()) {\n removeChild(firstChild!!)\n }\n}\n\n/**\n * Creates text node and append it to the element.\n *\n * @return this element\n */\n@SinceKotlin(\"1.4\")\nfun Element.appendText(text: String): Element {\n appendChild(ownerDocument!!.createTextNode(text))\n return this\n}\n","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license
that can be found in the license/LICENSE.txt file.\n */\n\npackage org.w3c.dom\n\n@Deprecated(\"Use UnionMessagePortOrWindowProxy instead.\", ReplaceWith(\"UnionMessagePortOrWindowProxy\"))\ntypealias UnionMessagePortOrWindow = UnionMessagePortOrWindowProxy\n\n@Deprecated(\"Use `as` instead.\", ReplaceWith(\"`as`\"))\nvar HTMLLinkElement.as_\n get() = `as`\n set(value) {\n `as` = value\n}\n\n@Deprecated(\"Use `is` instead.\", ReplaceWith(\"`is`\"))\nvar ElementCreationOptions.is_\n get() = `is`\n set(value) {\n `is` = value\n }","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.khronos.webgl\n\nimport kotlin.js.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\npublic external
interface WebGLContextAttributes {\n var alpha: Boolean? /* = true */\n get() = definedExternally\n set(value) = definedExternally\n var depth: Boolean? /* = true */\n get() = definedExternally\n set(value) = definedExternally\n var stencil: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var antialias: Boolean? /* = true */\n get() = definedExternally\n set(value) = definedExternally\n var premultipliedAlpha: Boolean? /* = true */\n get() = definedExternally\n set(value) = definedExternally\n var preserveDrawingBuffer: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var preferLowPowerToHighPerformance: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var failIfMajorPerformanceCaveat: Boolean? /* = false */\n get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun WebGLContextAttributes(alpha: Boolean? = true, depth: Boolean? = true, stencil: Boolean? = false, antialias: Boolean? = true, premultipliedAlpha: Boolean? = true, preserveDrawingBuffer: Boolean? = false, preferLowPowerToHighPerformance: Boolean? = false, failIfMajorPerformanceCaveat: Boolean? = false): WebGLContextAttributes {\n val o = js(\"({})\")\n o[\"alpha\"] = alpha\n o[\"depth\"] = depth\n o[\"stencil\"] = stencil\n o[\"antialias\"] = antialias\n o[\"premultipliedAlpha\"] = premultipliedAlpha\n o[\"preserveDrawingBuffer\"] = preserveDrawingBuffer\n o[\"preferLowPowerToHighPerformance\"] = preferLowPowerToHighPerformance\n

o[\"failIfMajorPerformanceCaveat\"] = failIfMajorPerformanceCaveat\n   return o\n}\n\npublic external abstract class WebGLObject\n\n/**\n * Exposes the JavaScript [WebGLBuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLBuffer) to Kotlin\n */\npublic external abstract class WebGLBuffer : WebGLObject\n\n/**\n * Exposes the JavaScript [WebGLFramebuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLFramebuffer) to Kotlin\n */\npublic external abstract class WebGLFramebuffer : WebGLObject\n\n/**\n * Exposes the JavaScript [WebGLProgram](https://developer.mozilla.org/en/docs/Web/API/WebGLProgram) to Kotlin\n */\npublic external abstract class WebGLProgram : WebGLObject\n\n/**\n * Exposes the JavaScript [WebGLRenderbuffer](https://developer.mozilla.org/en/docs/Web/API/WebGLRenderbuffer) to Kotlin\n */\npublic external abstract class WebGLRenderbuffer : WebGLObject\n\n/**\n * Exposes the JavaScript [WebGLShader](https://developer.mozilla.org/en/docs/Web/API/WebGLShader) to Kotlin\n */\npublic external abstract class WebGLShader : WebGLObject\n\n/**\n * Exposes the JavaScript [WebGLTexture](https://developer.mozilla.org/en/docs/Web/API/WebGLTexture) to Kotlin\n */\npublic external abstract class WebGLTexture : WebGLObject\n\n/**\n * Exposes the JavaScript [WebGLUniformLocation](https://developer.mozilla.org/en/docs/Web/API/WebGLUniformLocation) to Kotlin\n */\npublic external abstract class WebGLUniformLocation\n\n/**\n * Exposes the JavaScript [WebGLActiveInfo](https://developer.mozilla.org/en/docs/Web/API/WebGLActiveInfo) to Kotlin\n */\npublic external abstract class WebGLActiveInfo {\n   open val size: Int\n   open val type: Int\n   open val name: String\n}\n\n/**\n * Exposes the JavaScript [WebGLShaderPrecisionFormat](https://developer.mozilla.org/en/docs/Web/API/WebGLShaderPrecisionFormat) to Kotlin\n */\npublic external abstract class WebGLShaderPrecisionFormat {\n   open val rangeMin: Int\n   open val rangeMax: Int\n   open val precision: Int\n}\n\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface WebGLRenderingContextBase {\n   val canvas: HTMLCanvasElement\n   val drawingBufferWidth: Int\n   val drawingBufferHeight: Int\n   fun getContextAttributes(): WebGLContextAttributes?\n   fun isContextLost(): Boolean\n   fun getSupportedExtensions(): Array<String>?\n   fun getExtension(name: String): dynamic\n   fun activeTexture(texture: Int)\n   fun attachShader(program: WebGLProgram?, shader: WebGLShader?)\n   fun bindAttribLocation(program: WebGLProgram?, index: Int, name: String)\n   fun bindBuffer(target: Int, buffer: WebGLBuffer?)\n   fun bindFramebuffer(target: Int, framebuffer: WebGLFramebuffer?)\n   fun bindRenderbuffer(target: Int, renderbuffer: WebGLRenderbuffer?)\n   fun bindTexture(target: Int, texture: WebGLTexture?)\n   fun blendColor(red: Float, green: Float, blue: Float, alpha: Float)\n   fun blendEquation(mode: Int)\n   fun blendEquationSeparate(modeRGB: Int, modeAlpha: Int)\n   fun blendFunc(sfactor: Int, dfactor: Int)\n   fun blendFuncSeparate(srcRGB: Int, dstRGB: Int, srcAlpha: Int, dstAlpha: Int)\n   fun bufferData(target: Int, size: Int, usage: Int)\n   fun bufferData(target: Int, data: BufferDataSource?, usage: Int)\n   fun bufferSubData(target: Int, offset: Int, data: BufferDataSource?)\n   fun checkFramebufferStatus(target: Int): Int\n   fun clear(mask: Int)\n   fun clearColor(red: Float, green: Float, blue: Float, alpha: Float)\n   fun clearDepth(depth: Float)\n   fun clearStencil(s: Int)\n   fun colorMask(red: Boolean, green: Boolean, blue: Boolean, alpha: Boolean)\n   fun compileShader(shader: WebGLShader?)\n   fun compressedTexImage2D(target: Int, level: Int, internalformat: Int, width: Int, height: Int, border: Int, data: ArrayBufferView)\n   fun compressedTexSubImage2D(target: Int, level: Int, xoffset: Int, yoffset: Int, width: Int, height: Int, format: Int, data: ArrayBufferView)\n   fun copyTexImage2D(target: Int, level: Int, internalformat: Int, x: Int, y: Int, width: Int, height: Int, border: Int)\n   fun copyTexSubImage2D(target: Int, level: Int, xoffset: Int, yoffset: Int, x: Int, y: Int, width: Int, height: Int)\n   fun createBuffer(): WebGLBuffer?\n   fun createFramebuffer(): WebGLFramebuffer?\n   fun createProgram(): WebGLProgram?\n   fun createRenderbuffer(): WebGLRenderbuffer?\n   fun createShader(type: Int): WebGLShader?\n   fun createTexture(): WebGLTexture?\n   fun cullFace(mode: Int)\n   fun deleteBuffer(buffer: WebGLBuffer?)\n   fun deleteFramebuffer(framebuffer: WebGLFramebuffer?)\n   fun

deleteProgram(program: WebGLProgram?)\n fun deleteRenderbuffer(renderbuffer: WebGLRenderbuffer?)\n fun deleteShader(shader: WebGLShader?)\n fun deleteTexture(texture: WebGLTexture?)\n fun depthFunc(func: Int)\n fun depthMask(flag: Boolean)\n fun depthRange(zNear: Float, zFar: Float)\n fun detachShader(program: WebGLProgram?, shader: WebGLShader?)\n fun disable(cap: Int)\n fun disableVertexAttribArray(index: Int)\n fun drawArrays(mode: Int, first: Int, count: Int)\n fun drawElements(mode: Int, count: Int, type: Int, offset: Int)\n fun enable(cap:
Int)\n fun enableVertexAttribArray(index: Int)\n fun finish()\n fun flush()\n fun framebufferRenderbuffer(target: Int, attachment: Int, renderbuffertarget: Int, renderbuffer: WebGLRenderbuffer?)\n fun framebufferTexture2D(target: Int, attachment: Int, textarget: Int, texture: WebGLTexture?, level: Int)\n fun frontFace(mode: Int)\n fun generateMipmap(target: Int)\n fun getActiveAttrib(program: WebGLProgram?, index: Int): WebGLActiveInfo?\n fun getActiveUniform(program: WebGLProgram?, index: Int): WebGLActiveInfo?\n fun getAttachedShaders(program: WebGLProgram?): Array<WebGLShader>?\n fun getAttribLocation(program: WebGLProgram?, name: String): Int\n fun getBufferParameter(target: Int, pname: Int): Any?\n fun getParameter(pname: Int): Any?\n fun getError(): Int\n fun getFramebufferAttachmentParameter(target: Int, attachment: Int, pname: Int): Any?\n fun getProgramParameter(program: WebGLProgram?, pname: Int): Any?\n fun getProgramInfoLog(program: WebGLProgram?): String?\n fun getRenderbufferParameter(target: Int, pname: Int): Any?\n fun getShaderParameter(shader: WebGLShader?, pname: Int): Any?\n fun getShaderPrecisionFormat(shadertype: Int, precisiontype: Int): WebGLShaderPrecisionFormat?\n fun getShaderInfoLog(shader: WebGLShader?): String?\n fun getShaderSource(shader: WebGLShader?): String?\n fun getTexParameter(target: Int, pname: Int): Any?\n fun getUniform(program: WebGLProgram?, location: WebGLUniformLocation?): Any?\n fun getUniformLocation(program: WebGLProgram?, name: String): WebGLUniformLocation?\n fun getVertexAttrib(index: Int, pname: Int): Any?\n fun getVertexAttribOffset(index: Int, pname: Int): Int\n fun hint(target: Int, mode: Int)\n fun isBuffer(buffer: WebGLBuffer?): Boolean\n fun isEnabled(cap: Int): Boolean\n fun isFramebuffer(framebuffer: WebGLFramebuffer?): Boolean\n fun isProgram(program: WebGLProgram?): Boolean\n fun isRenderbuffer(renderbuffer:
WebGLRenderbuffer?): Boolean\n fun isShader(shader: WebGLShader?): Boolean\n fun isTexture(texture: WebGLTexture?): Boolean\n fun lineWidth(width: Float)\n fun linkProgram(program: WebGLProgram?)\n fun pixelStorei(pname: Int, param: Int)\n fun polygonOffset(factor: Float, units: Float)\n fun readPixels(x: Int, y: Int, width: Int, height: Int, format: Int, type: Int, pixels: ArrayBufferView?)\n fun renderbufferStorage(target: Int, internalformat: Int, width: Int, height: Int)\n fun sampleCoverage(value: Float, invert: Boolean)\n fun scissor(x: Int, y: Int, width: Int, height: Int)\n fun shaderSource(shader: WebGLShader?, source: String)\n fun stencilFunc(func: Int, ref: Int, mask: Int)\n fun stencilFuncSeparate(face: Int, func: Int, ref: Int, mask: Int)\n fun stencilMask(mask: Int)\n fun stencilMaskSeparate(face: Int, mask: Int)\n fun stencilOp(fail: Int, zfail: Int, zpass: Int)\n fun stencilOpSeparate(face: Int, fail: Int, zfail: Int,
zpass: Int)\n fun texImage2D(target: Int, level: Int, internalformat: Int, width: Int, height: Int, border: Int, format: Int, type: Int, pixels: ArrayBufferView?)\n fun texImage2D(target: Int, level: Int, internalformat: Int, format: Int, type: Int, source: TexImageSource?)\n fun texParameterf(target: Int, pname: Int, param: Float)\n fun texParameteri(target: Int, pname: Int, param: Int)\n fun texSubImage2D(target: Int, level: Int, xoffset: Int, yoffset: Int, width: Int, height: Int, format: Int, type: Int, pixels: ArrayBufferView?)\n fun texSubImage2D(target: Int, level: Int, xoffset: Int, yoffset: Int, format: Int, type: Int, source: TexImageSource?)\n fun uniform1f(location: WebGLUniformLocation?, x: Float)\n fun uniform1fv(location: WebGLUniformLocation?, v: Float32Array)\n fun uniform1fv(location: WebGLUniformLocation?, v: Array<Float>)\n fun uniform1i(location: WebGLUniformLocation?, x: Int)\n fun uniform1iv(location: WebGLUniformLocation?, v:
Int32Array)\n fun uniform1iv(location: WebGLUniformLocation?, v: Array<Int>)\n fun uniform2f(location: WebGLUniformLocation?, x: Float, y: Float)\n fun uniform2fv(location: WebGLUniformLocation?, v: Float32Array)\n fun uniform2fv(location: WebGLUniformLocation?, v: Array<Float>)\n fun uniform2i(location: WebGLUniformLocation?, x: Int, y: Int)\n fun uniform2iv(location:

WebGLUniformLocation?, v: Int32Array)\n fun uniform2iv(location: WebGLUniformLocation?, v: Array<Int>)\n fun uniform3f(location: WebGLUniformLocation?, x: Float, y: Float, z: Float)\n fun uniform3fv(location: WebGLUniformLocation?, v: Float32Array)\n fun uniform3fv(location: WebGLUniformLocation?, v: Array<Float>)\n fun uniform3i(location: WebGLUniformLocation?, x: Int, y: Int, z: Int)\n fun uniform3iv(location: WebGLUniformLocation?, v: Int32Array)\n fun uniform3iv(location: WebGLUniformLocation?, v: Array<Int>)\n fun uniform4f(location: WebGLUniformLocation?, x: Float, y: Float, z: Float, w: Float)\n fun uniform4fv(location: WebGLUniformLocation?, v: Float32Array)\n fun uniform4fv(location: WebGLUniformLocation?, v: Array<Float>)\n fun uniform4i(location: WebGLUniformLocation?, x: Int, y: Int, z: Int, w: Int)\n fun uniform4iv(location: WebGLUniformLocation?, v: Int32Array)\n fun uniform4iv(location: WebGLUniformLocation?, v: Array<Int>)\n fun uniformMatrix2fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)\n fun uniformMatrix2fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)\n fun uniformMatrix3fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)\n fun uniformMatrix3fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)\n fun uniformMatrix4fv(location: WebGLUniformLocation?, transpose: Boolean, value: Float32Array)\n fun uniformMatrix4fv(location: WebGLUniformLocation?, transpose: Boolean, value: Array<Float>)\n fun useProgram(program: WebGLProgram?)\n fun validateProgram(program: WebGLProgram?)\n fun vertexAttrib1f(index: Int, x: Float)\n fun vertexAttrib1fv(index: Int, values: dynamic)\n fun vertexAttrib2f(index: Int, x: Float, y: Float)\n fun vertexAttrib2fv(index: Int, values: dynamic)\n fun vertexAttrib3f(index: Int, x: Float, y: Float, z: Float)\n fun vertexAttrib3fv(index: Int, values: dynamic)\n fun vertexAttrib4f(index: Int, x: Float, y: Float, z: Float, w: Float)\n fun vertexAttrib4fv(index: Int, values: dynamic)\n fun vertexAttribPointer(index: Int, size: Int, type: Int, normalized: Boolean, stride: Int, offset: Int)\n fun viewport(x: Int, y: Int, width: Int, height: Int)\n\n companion object {\n val DEPTH_BUFFER_BIT: Int\n val STENCIL_BUFFER_BIT: Int\n val COLOR_BUFFER_BIT: Int\n val POINTS: Int\n val LINES: Int\n val LINE_LOOP: Int\n val LINE_STRIP: Int\n val TRIANGLES: Int\n val TRIANGLE_STRIP: Int\n val TRIANGLE_FAN: Int\n val ZERO: Int\n val ONE: Int\n val SRC_COLOR: Int\n val ONE_MINUS_SRC_COLOR: Int\n val SRC_ALPHA: Int\n val ONE_MINUS_SRC_ALPHA: Int\n val DST_ALPHA: Int\n val ONE_MINUS_DST_ALPHA: Int\n val DST_COLOR: Int\n val ONE_MINUS_DST_COLOR: Int\n val SRC_ALPHA_SATURATE: Int\n val FUNC_ADD: Int\n val BLEND_EQUATION: Int\n val BLEND_EQUATION_RGB: Int\n val BLEND_EQUATION_ALPHA: Int\n val FUNC_SUBTRACT: Int\n val FUNC_REVERSE_SUBTRACT: Int\n val BLEND_DST_RGB: Int\n val BLEND_SRC_RGB: Int\n val BLEND_DST_ALPHA: Int\n val BLEND_SRC_ALPHA: Int\n val CONSTANT_COLOR: Int\n val ONE_MINUS_CONSTANT_COLOR: Int\n val CONSTANT_ALPHA: Int\n val ONE_MINUS_CONSTANT_ALPHA: Int\n val BLEND_COLOR: Int\n val ARRAY_BUFFER: Int\n val ELEMENT_ARRAY_BUFFER: Int\n val ARRAY_BUFFER_BINDING: Int\n val ELEMENT_ARRAY_BUFFER_BINDING: Int\n val STREAM_DRAW: Int\n val STATIC_DRAW: Int\n val DYNAMIC_DRAW: Int\n val BUFFER_SIZE: Int\n val BUFFER_USAGE: Int\n val CURRENT_VERTEX_ATTRIB: Int\n val FRONT: Int\n val BACK: Int\n val FRONT_AND_BACK: Int\n val CULL_FACE: Int\n val BLEND: Int\n val DITHER: Int\n val STENCIL_TEST: Int\n val DEPTH_TEST: Int\n val SCISSOR_TEST: Int\n val POLYGON_OFFSET_FILL: Int\n val SAMPLE_ALPHA_TO_COVERAGE: Int\n val SAMPLE_COVERAGE: Int\n val NO_ERROR: Int\n val INVALID_ENUM: Int\n val INVALID_VALUE: Int\n val INVALID_OPERATION: Int\n val OUT_OF_MEMORY: Int\n val CW: Int\n val CCW: Int\n val LINE_WIDTH: Int\n val ALIASED_POINT_SIZE_RANGE: Int\n val ALIASED_LINE_WIDTH_RANGE: Int\n val CULL_FACE_MODE: Int\n val FRONT_FACE: Int\n

val DEPTH_RANGE: Int\n val DEPTH_WRITEMASK: Int\n val DEPTH_CLEAR_VALUE: Int\n val DEPTH_FUNC: Int\n val STENCIL_CLEAR_VALUE: Int\n val STENCIL_FUNC: Int\n val STENCIL_FAIL: Int\n val STENCIL_PASS_DEPTH_FAIL: Int\n val STENCIL_PASS_DEPTH_PASS: Int\n val STENCIL_REF: Int\n val STENCIL_VALUE_MASK: Int\n val STENCIL_WRITEMASK: Int\n val STENCIL_BACK_FUNC: Int\n val STENCIL_BACK_FAIL: Int\n val STENCIL_BACK_PASS_DEPTH_FAIL: Int\n val STENCIL_BACK_PASS_DEPTH_PASS: Int\n val STENCIL_BACK_REF: Int\n val STENCIL_BACK_VALUE_MASK: Int\n val STENCIL_BACK_WRITEMASK: Int\n val VIEWPORT: Int\n val SCISSOR_BOX: Int\n val COLOR_CLEAR_VALUE: Int\n val COLOR_WRITEMASK: Int\n val UNPACK_ALIGNMENT: Int\n val PACK_ALIGNMENT: Int\n val MAX_TEXTURE_SIZE: Int\n val MAX_VIEWPORT_DIMS: Int\n val SUBPIXEL_BITS: Int\n
val RED_BITS: Int\n val GREEN_BITS: Int\n val BLUE_BITS: Int\n val ALPHA_BITS: Int\n val DEPTH_BITS: Int\n val STENCIL_BITS: Int\n val POLYGON_OFFSET_UNITS: Int\n val POLYGON_OFFSET_FACTOR: Int\n val TEXTURE_BINDING_2D: Int\n val SAMPLE_BUFFERS: Int\n val SAMPLES: Int\n val SAMPLE_COVERAGE_VALUE: Int\n val SAMPLE_COVERAGE_INVERT: Int\n val COMPRESSED_TEXTURE_FORMATS: Int\n val DONT_CARE: Int\n val FASTEST: Int\n val NICEST: Int\n val GENERATE_MIPMAP_HINT: Int\n val BYTE: Int\n val UNSIGNED_BYTE: Int\n val SHORT: Int\n val UNSIGNED_SHORT: Int\n val INT: Int\n val UNSIGNED_INT: Int\n val FLOAT: Int\n val DEPTH_COMPONENT: Int\n val ALPHA: Int\n val RGB: Int\n val RGBA: Int\n val LUMINANCE: Int\n val LUMINANCE_ALPHA: Int\n val UNSIGNED_SHORT_4_4_4_4: Int\n val UNSIGNED_SHORT_5_5_5_1:
Int\n val UNSIGNED_SHORT_5_6_5: Int\n val FRAGMENT_SHADER: Int\n val VERTEX_SHADER: Int\n val MAX_VERTEX_ATTRIBS: Int\n val MAX_VERTEX_UNIFORM_VECTORS: Int\n val MAX_VARYING_VECTORS: Int\n val MAX_COMBINED_TEXTURE_IMAGE_UNITS: Int\n val MAX_VERTEX_TEXTURE_IMAGE_UNITS: Int\n val MAX_TEXTURE_IMAGE_UNITS: Int\n val MAX_FRAGMENT_UNIFORM_VECTORS: Int\n val SHADER_TYPE: Int\n val DELETE_STATUS: Int\n val LINK_STATUS: Int\n val VALIDATE_STATUS: Int\n val ATTACHED_SHADERS: Int\n val ACTIVE_UNIFORMS: Int\n val ACTIVE_ATTRIBUTES: Int\n val SHADING_LANGUAGE_VERSION: Int\n val CURRENT_PROGRAM: Int\n val NEVER: Int\n val LESS: Int\n val EQUAL: Int\n val LEQUAL: Int\n val GREATER: Int\n val NOTEQUAL: Int\n val GEQUAL: Int\n val ALWAYS: Int\n val KEEP: Int\n val REPLACE: Int\n val
INCR: Int\n val DECR: Int\n val INVERT: Int\n val INCR_WRAP: Int\n val DECR_WRAP: Int\n val VENDOR: Int\n val RENDERER: Int\n val VERSION: Int\n val NEAREST: Int\n val LINEAR: Int\n val NEAREST_MIPMAP_NEAREST: Int\n val LINEAR_MIPMAP_NEAREST: Int\n val NEAREST_MIPMAP_LINEAR: Int\n val LINEAR_MIPMAP_LINEAR: Int\n val TEXTURE_MAG_FILTER: Int\n val TEXTURE_MIN_FILTER: Int\n val TEXTURE_WRAP_S: Int\n val TEXTURE_WRAP_T: Int\n val TEXTURE_2D: Int\n val TEXTURE: Int\n val TEXTURE_CUBE_MAP: Int\n val TEXTURE_BINDING_CUBE_MAP: Int\n val TEXTURE_CUBE_MAP_POSITIVE_X: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_X: Int\n val TEXTURE_CUBE_MAP_POSITIVE_Y: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_Y: Int\n val TEXTURE_CUBE_MAP_POSITIVE_Z: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_Z: Int\n val MAX_CUBE_MAP_TEXTURE_SIZE: Int\n
val TEXTURE0: Int\n val TEXTURE1: Int\n val TEXTURE2: Int\n val TEXTURE3: Int\n val TEXTURE4: Int\n val TEXTURE5: Int\n val TEXTURE6: Int\n val TEXTURE7: Int\n val TEXTURE8: Int\n val TEXTURE9: Int\n val TEXTURE10: Int\n val TEXTURE11: Int\n val TEXTURE12: Int\n val TEXTURE13: Int\n val TEXTURE14: Int\n val TEXTURE15: Int\n val

TEXTURE16: Int\n val TEXTURE17: Int\n val TEXTURE18: Int\n val TEXTURE19: Int\n val TEXTURE20: Int\n val TEXTURE21: Int\n val TEXTURE22: Int\n val TEXTURE23: Int\n val TEXTURE24: Int\n val TEXTURE25: Int\n val TEXTURE26: Int\n val TEXTURE27: Int\n val TEXTURE28: Int\n val TEXTURE29: Int\n val TEXTURE30: Int\n val TEXTURE31: Int\n val ACTIVE_TEXTURE: Int\n val REPEAT: Int\n val CLAMP_TO_EDGE: Int\n val MIRRORED_REPEAT:
Int\n val FLOAT_VEC2: Int\n val FLOAT_VEC3: Int\n val FLOAT_VEC4: Int\n val INT_VEC2: Int\n val INT_VEC3: Int\n val INT_VEC4: Int\n val BOOL: Int\n val BOOL_VEC2: Int\n val BOOL_VEC3: Int\n val BOOL_VEC4: Int\n val FLOAT_MAT2: Int\n val FLOAT_MAT3: Int\n val FLOAT_MAT4: Int\n val SAMPLER_2D: Int\n val SAMPLER_CUBE: Int\n val VERTEX_ATTRIB_ARRAY_ENABLED: Int\n val VERTEX_ATTRIB_ARRAY_SIZE: Int\n val VERTEX_ATTRIB_ARRAY_STRIDE: Int\n val VERTEX_ATTRIB_ARRAY_TYPE: Int\n val VERTEX_ATTRIB_ARRAY_NORMALIZED: Int\n val VERTEX_ATTRIB_ARRAY_POINTER: Int\n val VERTEX_ATTRIB_ARRAY_BUFFER_BINDING: Int\n val IMPLEMENTATION_COLOR_READ_TYPE: Int\n val IMPLEMENTATION_COLOR_READ_FORMAT: Int\n val COMPILE_STATUS: Int\n val LOW_FLOAT: Int\n val MEDIUM_FLOAT: Int\n val HIGH_FLOAT: Int\n val LOW_INT: Int\n
val MEDIUM_INT: Int\n val HIGH_INT: Int\n val FRAMEBUFFER: Int\n val RENDERBUFFER: Int\n val RGBA4: Int\n val RGB5_A1: Int\n val RGB565: Int\n val DEPTH_COMPONENT16: Int\n val STENCIL_INDEX: Int\n val STENCIL_INDEX8: Int\n val DEPTH_STENCIL: Int\n val RENDERBUFFER_WIDTH: Int\n val RENDERBUFFER_HEIGHT: Int\n val RENDERBUFFER_INTERNAL_FORMAT: Int\n val RENDERBUFFER_RED_SIZE: Int\n val RENDERBUFFER_GREEN_SIZE: Int\n val RENDERBUFFER_BLUE_SIZE: Int\n val RENDERBUFFER_ALPHA_SIZE: Int\n val RENDERBUFFER_DEPTH_SIZE: Int\n val RENDERBUFFER_STENCIL_SIZE: Int\n val FRAMEBUFFER_ATTACHMENT_OBJECT_TYPE: Int\n val FRAMEBUFFER_ATTACHMENT_OBJECT_NAME: Int\n val FRAMEBUFFER_ATTACHMENT_TEXTURE_LEVEL: Int\n val FRAMEBUFFER_ATTACHMENT_TEXTURE_CUBE_MAP_FACE: Int\n val COLOR_ATTACHMENT0: Int\n val DEPTH_ATTACHMENT: Int\n
val STENCIL_ATTACHMENT: Int\n val DEPTH_STENCIL_ATTACHMENT: Int\n val NONE: Int\n
val FRAMEBUFFER_COMPLETE: Int\n val FRAMEBUFFER_INCOMPLETE_ATTACHMENT: Int\n val FRAMEBUFFER_INCOMPLETE_MISSING_ATTACHMENT: Int\n val FRAMEBUFFER_INCOMPLETE_DIMENSIONS: Int\n val FRAMEBUFFER_UNSUPPORTED: Int\n val FRAMEBUFFER_BINDING: Int\n val RENDERBUFFER_BINDING: Int\n val MAX_RENDERBUFFER_SIZE: Int\n val INVALID_FRAMEBUFFER_OPERATION: Int\n val UNPACK_FLIP_Y_WEBGL: Int\n val UNPACK_PREMULTIPLY_ALPHA_WEBGL: Int\n val CONTEXT_LOST_WEBGL: Int\n val UNPACK_COLORSPACE_CONVERSION_WEBGL: Int\n val BROWSER_DEFAULT_WEBGL: Int\n }\n}\n\n/**\n * Exposes the JavaScript [WebGLRenderingContext](https://developer.mozilla.org/en/docs/Web/API/WebGLRenderingContext) to Kotlin\n */\npublic external abstract class WebGLRenderingContext : WebGLRenderingContextBase, RenderingContext {\n companion object
{\n val DEPTH_BUFFER_BIT: Int\n val STENCIL_BUFFER_BIT: Int\n val COLOR_BUFFER_BIT: Int\n val POINTS: Int\n val LINES: Int\n val LINE_LOOP: Int\n val LINE_STRIP: Int\n val TRIANGLES: Int\n val TRIANGLE_STRIP: Int\n val TRIANGLE_FAN: Int\n val ZERO: Int\n val ONE: Int\n val SRC_COLOR: Int\n val ONE_MINUS_SRC_COLOR: Int\n val SRC_ALPHA: Int\n val ONE_MINUS_SRC_ALPHA: Int\n val DST_ALPHA: Int\n val ONE_MINUS_DST_ALPHA: Int\n val DST_COLOR: Int\n val ONE_MINUS_DST_COLOR: Int\n val SRC_ALPHA_SATURATE: Int\n val FUNC_ADD: Int\n val BLEND_EQUATION: Int\n val

BLEND_EQUATION_RGB: Int\n val BLEND_EQUATION_ALPHA: Int\n val FUNC_SUBTRACT: Int\n val FUNC_REVERSE_SUBTRACT: Int\n val BLEND_DST_RGB: Int\n val BLEND_SRC_RGB: Int\n val BLEND_DST_ALPHA: Int\n val BLEND_SRC_ALPHA: Int\n val CONSTANT_COLOR: Int\n val ONE_MINUS_CONSTANT_COLOR: Int\n val CONSTANT_ALPHA: Int\n val ONE_MINUS_CONSTANT_ALPHA: Int\n val BLEND_COLOR: Int\n val ARRAY_BUFFER: Int\n val ELEMENT_ARRAY_BUFFER: Int\n val ARRAY_BUFFER_BINDING: Int\n val ELEMENT_ARRAY_BUFFER_BINDING: Int\n val STREAM_DRAW: Int\n val STATIC_DRAW: Int\n val DYNAMIC_DRAW: Int\n val BUFFER_SIZE: Int\n val BUFFER_USAGE: Int\n val CURRENT_VERTEX_ATTRIB: Int\n val FRONT: Int\n val BACK: Int\n val FRONT_AND_BACK: Int\n val CULL_FACE: Int\n val BLEND: Int\n val DITHER: Int\n val STENCIL_TEST: Int\n val DEPTH_TEST: Int\n val SCISSOR_TEST: Int\n val POLYGON_OFFSET_FILL: Int\n val SAMPLE_ALPHA_TO_COVERAGE: Int\n val SAMPLE_COVERAGE: Int\n val NO_ERROR: Int\n val INVALID_ENUM: Int\n val INVALID_VALUE: Int\n val INVALID_OPERATION: Int\n val OUT_OF_MEMORY: Int\n val CW: Int\n val CCW: Int\n val LINE_WIDTH: Int\n val ALIASED_POINT_SIZE_RANGE: Int\n val ALIASED_LINE_WIDTH_RANGE: Int\n val CULL_FACE_MODE: Int\n val FRONT_FACE: Int\n val DEPTH_RANGE: Int\n val DEPTH_WRITEMASK: Int\n val DEPTH_CLEAR_VALUE: Int\n val DEPTH_FUNC: Int\n val STENCIL_CLEAR_VALUE: Int\n val STENCIL_FUNC: Int\n val STENCIL_FAIL: Int\n val STENCIL_PASS_DEPTH_FAIL: Int\n val STENCIL_PASS_DEPTH_PASS: Int\n val STENCIL_REF: Int\n val STENCIL_VALUE_MASK: Int\n val STENCIL_WRITEMASK: Int\n val STENCIL_BACK_FUNC: Int\n val STENCIL_BACK_FAIL: Int\n val STENCIL_BACK_PASS_DEPTH_FAIL: Int\n val STENCIL_BACK_PASS_DEPTH_PASS: Int\n val STENCIL_BACK_REF: Int\n val STENCIL_BACK_VALUE_MASK: Int\n val STENCIL_BACK_WRITEMASK: Int\n val VIEWPORT: Int\n val SCISSOR_BOX: Int\n val COLOR_CLEAR_VALUE: Int\n val COLOR_WRITEMASK: Int\n val UNPACK_ALIGNMENT: Int\n val PACK_ALIGNMENT: Int\n val MAX_TEXTURE_SIZE: Int\n val MAX_VIEWPORT_DIMS: Int\n val SUBPIXEL_BITS: Int\n val RED_BITS: Int\n val GREEN_BITS: Int\n val BLUE_BITS: Int\n val ALPHA_BITS: Int\n val DEPTH_BITS: Int\n val STENCIL_BITS: Int\n val POLYGON_OFFSET_UNITS: Int\n val POLYGON_OFFSET_FACTOR: Int\n val TEXTURE_BINDING_2D: Int\n val SAMPLE_BUFFERS: Int\n val SAMPLES: Int\n val SAMPLE_COVERAGE_VALUE: Int\n val SAMPLE_COVERAGE_INVERT: Int\n val COMPRESSED_TEXTURE_FORMATS: Int\n val DONT_CARE: Int\n val FASTEST: Int\n val NICEST: Int\n val GENERATE_MIPMAP_HINT: Int\n val BYTE: Int\n val UNSIGNED_BYTE: Int\n val SHORT: Int\n val UNSIGNED_SHORT: Int\n val INT: Int\n val UNSIGNED_INT: Int\n val FLOAT: Int\n val DEPTH_COMPONENT: Int\n val ALPHA: Int\n val RGB: Int\n val RGBA: Int\n val LUMINANCE: Int\n val LUMINANCE_ALPHA: Int\n val UNSIGNED_SHORT_4_4_4_4: Int\n val UNSIGNED_SHORT_5_5_5_1: Int\n val UNSIGNED_SHORT_5_6_5: Int\n val FRAGMENT_SHADER: Int\n val VERTEX_SHADER: Int\n val MAX_VERTEX_ATTRIBS: Int\n val MAX_VERTEX_UNIFORM_VECTORS: Int\n val MAX_VARYING_VECTORS: Int\n val MAX_COMBINED_TEXTURE_IMAGE_UNITS: Int\n val MAX_VERTEX_TEXTURE_IMAGE_UNITS: Int\n val MAX_TEXTURE_IMAGE_UNITS: Int\n val MAX_FRAGMENT_UNIFORM_VECTORS: Int\n val SHADER_TYPE: Int\n val DELETE_STATUS: Int\n val LINK_STATUS: Int\n val VALIDATE_STATUS: Int\n val ATTACHED_SHADERS: Int\n val ACTIVE_UNIFORMS: Int\n val ACTIVE_ATTRIBUTES: Int\n val SHADING_LANGUAGE_VERSION: Int\n val CURRENT_PROGRAM: Int\n val NEVER: Int\n val LESS: Int\n val EQUAL: Int\n val LEQUAL: Int\n val GREATER: Int\n val NOTEQUAL: Int\n val GEQUAL: Int\n val ALWAYS: Int\n val KEEP:

Int\n val REPLACE: Int\n val INCR: Int\n val DECR: Int\n val INVERT: Int\n val INCR_WRAP: Int\n val DECR_WRAP: Int\n val VENDOR: Int\n val RENDERER: Int\n val VERSION: Int\n val NEAREST: Int\n val LINEAR: Int\n val NEAREST_MIPMAP_NEAREST: Int\n val LINEAR_MIPMAP_NEAREST: Int\n val NEAREST_MIPMAP_LINEAR: Int\n val LINEAR_MIPMAP_LINEAR: Int\n val TEXTURE_MAG_FILTER: Int\n val TEXTURE_MIN_FILTER: Int\n val TEXTURE_WRAP_S: Int\n val TEXTURE_WRAP_T: Int\n val TEXTURE_2D: Int\n val TEXTURE: Int\n val TEXTURE_CUBE_MAP: Int\n val TEXTURE_BINDING_CUBE_MAP: Int\n val TEXTURE_CUBE_MAP_POSITIVE_X: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_X: Int\n val TEXTURE_CUBE_MAP_POSITIVE_Y: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_Y: Int\n val TEXTURE_CUBE_MAP_POSITIVE_Z: Int\n val TEXTURE_CUBE_MAP_NEGATIVE_Z: Int\n val MAX_CUBE_MAP_TEXTURE_SIZE: Int\n val TEXTURE0: Int\n val TEXTURE1: Int\n val TEXTURE2: Int\n val TEXTURE3: Int\n val TEXTURE4: Int\n val TEXTURE5: Int\n val TEXTURE6: Int\n val TEXTURE7: Int\n val TEXTURE8: Int\n val TEXTURE9: Int\n val TEXTURE10: Int\n val TEXTURE11: Int\n val TEXTURE12: Int\n val TEXTURE13: Int\n val TEXTURE14: Int\n val TEXTURE15: Int\n val TEXTURE16: Int\n val TEXTURE17: Int\n val TEXTURE18: Int\n val TEXTURE19: Int\n val TEXTURE20: Int\n val TEXTURE21: Int\n val TEXTURE22: Int\n val TEXTURE23: Int\n val TEXTURE24: Int\n val TEXTURE25: Int\n val TEXTURE26: Int\n val TEXTURE27: Int\n val TEXTURE28: Int\n val TEXTURE29: Int\n val TEXTURE30: Int\n val TEXTURE31: Int\n val ACTIVE_TEXTURE: Int\n val REPEAT: Int\n val CLAMP_TO_EDGE: Int\n val MIRRORED_REPEAT: Int\n val FLOAT_VEC2: Int\n val FLOAT_VEC3: Int\n val FLOAT_VEC4: Int\n val INT_VEC2: Int\n val INT_VEC3: Int\n val INT_VEC4: Int\n val BOOL: Int\n val BOOL_VEC2: Int\n val BOOL_VEC3: Int\n val BOOL_VEC4: Int\n val FLOAT_MAT2: Int\n val FLOAT_MAT3: Int\n val FLOAT_MAT4: Int\n val SAMPLER_2D: Int\n val SAMPLER_CUBE: Int\n val VERTEX_ATTRIB_ARRAY_ENABLED: Int\n val VERTEX_ATTRIB_ARRAY_SIZE: Int\n val VERTEX_ATTRIB_ARRAY_STRIDE: Int\n val VERTEX_ATTRIB_ARRAY_TYPE: Int\n val VERTEX_ATTRIB_ARRAY_NORMALIZED: Int\n val VERTEX_ATTRIB_ARRAY_POINTER: Int\n val VERTEX_ATTRIB_ARRAY_BUFFER_BINDING: Int\n val IMPLEMENTATION_COLOR_READ_TYPE: Int\n val IMPLEMENTATION_COLOR_READ_FORMAT: Int\n val COMPILE_STATUS: Int\n val LOW_FLOAT: Int\n val MEDIUM_FLOAT: Int\n val HIGH_FLOAT: Int\n val LOW_INT: Int\n val MEDIUM_INT: Int\n val HIGH_INT: Int\n val FRAMEBUFFER: Int\n val RENDERBUFFER: Int\n val RGBA4: Int\n val RGB5_A1: Int\n val RGB565: Int\n val DEPTH_COMPONENT16: Int\n val STENCIL_INDEX: Int\n val STENCIL_INDEX8: Int\n val DEPTH_STENCIL: Int\n val RENDERBUFFER_WIDTH: Int\n val RENDERBUFFER_HEIGHT: Int\n val RENDERBUFFER_INTERNAL_FORMAT: Int\n val RENDERBUFFER_RED_SIZE: Int\n val RENDERBUFFER_GREEN_SIZE: Int\n val RENDERBUFFER_BLUE_SIZE: Int\n val RENDERBUFFER_ALPHA_SIZE: Int\n val RENDERBUFFER_DEPTH_SIZE: Int\n val RENDERBUFFER_STENCIL_SIZE: Int\n val FRAMEBUFFER_ATTACHMENT_OBJECT_TYPE: Int\n val FRAMEBUFFER_ATTACHMENT_OBJECT_NAME: Int\n val FRAMEBUFFER_ATTACHMENT_TEXTURE_LEVEL: Int\n val FRAMEBUFFER_ATTACHMENT_TEXTURE_CUBE_MAP_FACE: Int\n val COLOR_ATTACHMENT0: Int\n val DEPTH_ATTACHMENT: Int\n val STENCIL_ATTACHMENT: Int\n val DEPTH_STENCIL_ATTACHMENT: Int\n val NONE: Int\n val FRAMEBUFFER_COMPLETE: Int\n val FRAMEBUFFER_INCOMPLETE_ATTACHMENT: Int\n val FRAMEBUFFER_INCOMPLETE_MISSING_ATTACHMENT: Int\n val FRAMEBUFFER_INCOMPLETE_DIMENSIONS: Int\n val FRAMEBUFFER_UNSUPPORTED: Int\n val FRAMEBUFFER_BINDING: Int\n val RENDERBUFFER_BINDING: Int\n val

```
MAX_RENDERBUFFER_SIZE: Int\n        val INVALID_FRAMEBUFFER_OPERATION: Int\n        val
UNPACK_FLIP_Y_WEBGL: Int\n        val UNPACK_PREMULTIPLY_ALPHA_WEBGL: Int\n        val
CONTEXT_LOST_WEBGL: Int\n        val UNPACK_COLORSPACE_CONVERSION_WEBGL: Int\n        val
BROWSER_DEFAULT_WEBGL: Int\n
  }\n}\n\n/**\n * Exposes the JavaScript
[WebGLContextEvent](https://developer.mozilla.org/en/docs/Web/API/WebGLContextEvent) to Kotlin\n
*/\npublic external open class WebGLContextEvent(type: String, eventInit: WebGLContextEventInit =
definedExternally) : Event {\n    open val statusMessage: String\n\n    companion object {\n        val NONE: Short\n
    val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n
}\n}\n\npublic external interface WebGLContextEventInit : EventInit {\n    var statusMessage: String? /* = \"\" */\n
    get() = definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun
WebGLContextEventInit(statusMessage: String? = \"\", bubbles: Boolean? = false, cancelable: Boolean? = false,
composed: Boolean? = false): WebGLContextEventInit {\n    val o = js(\"({})\")\n    o[\"statusMessage\"] =
statusMessage\n    o[\"bubbles\"]
 = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\n/**\n * Exposes
the JavaScript [ArrayBuffer](https://developer.mozilla.org/en/docs/Web/API/ArrayBuffer) to Kotlin\n */\npublic
external open class ArrayBuffer(length: Int) : BufferDataSource {\n    open val byteLength: Int\n    fun slice(begin:
Int, end: Int = definedExternally): ArrayBuffer\n\n    companion object {\n        fun isView(value: Any?): Boolean\n
 }\n}\n\n/**\n * Exposes the JavaScript
[ArrayBufferView](https://developer.mozilla.org/en/docs/Web/API/ArrayBufferView) to Kotlin\n */\npublic
external interface ArrayBufferView : BufferDataSource {\n    val buffer: ArrayBuffer\n    val byteOffset: Int\n    val
byteLength: Int\n}\n\n/**\n * Exposes the JavaScript
[Int8Array](https://developer.mozilla.org/en/docs/Web/API/Int8Array) to Kotlin\n */\npublic external open class
Int8Array : ArrayBufferView {\n    constructor(length: Int)\n    constructor(array: Int8Array)\n
  constructor(array: Array<Byte>)\n    constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length:
Int = definedExternally)\n    open val length: Int\n    override val buffer: ArrayBuffer\n    override val byteOffset:
Int\n    override val byteLength: Int\n    fun set(array: Int8Array, offset: Int = definedExternally)\n    fun set(array:
Array<Byte>, offset: Int = definedExternally)\n    fun subarray(start: Int, end: Int): Int8Array\n\n    companion
object {\n        val BYTES_PER_ELEMENT: Int\n    }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int8Array.get(index: Int):
Byte = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int8Array.set(index: Int,
value: Byte) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript
[Uint8Array](https://developer.mozilla.org/en/docs/Web/API/Uint8Array) to Kotlin\n
 */\npublic external open class Uint8Array : ArrayBufferView {\n    constructor(length: Int)\n    constructor(array:
Uint8Array)\n    constructor(array: Array<Byte>)\n    constructor(buffer: ArrayBuffer, byteOffset: Int =
definedExternally, length: Int = definedExternally)\n    open val length: Int\n    override val buffer: ArrayBuffer\n
override val byteOffset: Int\n    override val byteLength: Int\n    fun set(array: Uint8Array, offset: Int =
definedExternally)\n    fun set(array: Array<Byte>, offset: Int = definedExternally)\n    fun subarray(start: Int, end:
Int): Uint8Array\n\n    companion object {\n        val BYTES_PER_ELEMENT: Int\n
}\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint8Array.get(index: Int):
Byte = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint8Array.set(index: Int,
value: Byte) { asDynamic()[index]
 = value }\n\n/**\n * Exposes the JavaScript
[Uint8ClampedArray](https://developer.mozilla.org/en/docs/Web/API/Uint8ClampedArray) to Kotlin\n */\npublic
```

external open class Uint8ClampedArray : ArrayBufferView {\n constructor(length: Int)\n constructor(array: Uint8ClampedArray)\n constructor(array: Array<Byte>)\n constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n override val byteOffset: Int\n override val byteLength: Int\n fun set(array: Uint8ClampedArray, offset: Int = definedExternally)\n fun set(array: Array<Byte>, offset: Int = definedExternally)\n fun subarray(start: Int, end: Int): Uint8ClampedArray\n\n companion object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint8ClampedArray.get(index: Int): Byte = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint8ClampedArray.set(index: Int, value: Byte) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript [Int16Array](https://developer.mozilla.org/en/docs/Web/API/Int16Array) to Kotlin\n */\npublic external open class Int16Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array: Int16Array)\n constructor(array: Array<Short>)\n constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n override val byteOffset: Int\n override val byteLength: Int\n fun set(array: Int16Array, offset: Int = definedExternally)\n fun set(array: Array<Short>, offset: Int = definedExternally)\n fun subarray(start: Int, end: Int): Int16Array\n\n companion object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int16Array.get(index: Int): Short = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int16Array.set(index: Int, value: Short) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript [Uint16Array](https://developer.mozilla.org/en/docs/Web/API/Uint16Array) to Kotlin\n */\npublic external open class Uint16Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array: Uint16Array)\n constructor(array: Array<Short>)\n constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n override val byteOffset: Int\n override val byteLength: Int\n fun set(array: Uint16Array, offset: Int = definedExternally)\n fun set(array: Array<Short>, offset: Int = definedExternally)\n fun subarray(start: Int, end: Int): Uint16Array\n\n companion object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint16Array.get(index: Int): Short = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint16Array.set(index: Int, value: Short) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript [Int32Array](https://developer.mozilla.org/en/docs/Web/API/Int32Array) to Kotlin\n */\npublic external open class Int32Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array: Int32Array)\n constructor(array: Array<Int>)\n constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)\n open val length: Int\n override val buffer: ArrayBuffer\n override val byteOffset: Int\n override val byteLength: Int\n fun set(array: Int32Array, offset: Int = definedExternally)\n fun set(array: Array<Int>, offset: Int = definedExternally)\n fun subarray(start: Int, end: Int): Int32Array\n\n companion object {\n val BYTES_PER_ELEMENT: Int\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int32Array.get(index: Int): Int = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Int32Array.set(index: Int, value: Int) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript [Uint32Array](https://developer.mozilla.org/en/docs/Web/API/Uint32Array) to Kotlin\n */\npublic external open class Uint32Array : ArrayBufferView {\n constructor(length: Int)\n constructor(array: Uint32Array)\n

constructor(array: Array<Int>)\n   constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)\n   open val length: Int\n   override val buffer: ArrayBuffer\n   override val byteOffset: Int\n   override val byteLength: Int\n   fun set(array: Uint32Array, offset: Int = definedExternally)\n   fun set(array: Array<Int>, offset: Int = definedExternally)\n   fun subarray(start: Int, end: Int): Uint32Array\n\n   companion object {\n   val BYTES_PER_ELEMENT: Int\n   }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint32Array.get(index: Int): Int = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Uint32Array.set(index: Int, value: Int) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript [Float32Array](https://developer.mozilla.org/en/docs/Web/API/Float32Array) to Kotlin\n */\npublic external open class Float32Array : ArrayBufferView {\n   constructor(length: Int)\n   constructor(array: Float32Array)\n   constructor(array: Array<Float>)\n   constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)\n   open val length: Int\n   override val buffer: ArrayBuffer\n   override val byteOffset: Int\n   override val byteLength: Int\n   fun set(array: Float32Array, offset: Int = definedExternally)\n   fun set(array: Array<Float>, offset: Int = definedExternally)\n   fun subarray(start: Int, end: Int): Float32Array\n\n   companion object {\n   val BYTES_PER_ELEMENT: Int\n   }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Float32Array.get(index: Int): Float = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Float32Array.set(index: Int, value: Float) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript [Float64Array](https://developer.mozilla.org/en/docs/Web/API/Float64Array) to Kotlin\n */\npublic external open class Float64Array : ArrayBufferView {\n   constructor(length: Int)\n   constructor(array: Float64Array)\n   constructor(array: Array<Double>)\n   constructor(buffer: ArrayBuffer, byteOffset: Int = definedExternally, length: Int = definedExternally)\n   open val length: Int\n   override val buffer: ArrayBuffer\n   override val byteOffset: Int\n   override val byteLength: Int\n   fun set(array: Float64Array, offset: Int = definedExternally)\n   fun set(array: Array<Double>, offset: Int = definedExternally)\n   fun subarray(start: Int, end: Int): Float64Array\n\n   companion object {\n   val BYTES_PER_ELEMENT: Int\n   }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Float64Array.get(index: Int): Double = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Float64Array.set(index: Int, value: Double) { asDynamic()[index] = value }\n\n/**\n * Exposes the JavaScript [DataView](https://developer.mozilla.org/en/docs/Web/API/DataView) to Kotlin\n */\npublic external open class DataView(buffer: ArrayBuffer, byteOffset: Int = definedExternally, byteLength: Int = definedExternally) : ArrayBufferView {\n   override val buffer: ArrayBuffer\n   override val byteOffset: Int\n   override val byteLength: Int\n   fun getInt8(byteOffset: Int): Byte\n   fun getUint8(byteOffset: Int): Byte\n   fun getInt16(byteOffset: Int, littleEndian: Boolean = definedExternally): Short\n   fun getUint16(byteOffset: Int, littleEndian: Boolean = definedExternally): Short\n   fun getInt32(byteOffset: Int, littleEndian: Boolean = definedExternally): Int\n   fun getUint32(byteOffset: Int, littleEndian: Boolean = definedExternally): Int\n   fun getFloat32(byteOffset: Int, littleEndian: Boolean = definedExternally): Float\n   fun getFloat64(byteOffset: Int, littleEndian: Boolean = definedExternally): Double\n   fun setInt8(byteOffset: Int, value: Byte)\n   fun setUint8(byteOffset: Int, value: Byte)\n   fun setInt16(byteOffset: Int, value: Short, littleEndian: Boolean = definedExternally)\n   fun setUint16(byteOffset: Int, value: Short, littleEndian: Boolean = definedExternally)\n   fun setInt32(byteOffset: Int, value: Int, littleEndian: Boolean = definedExternally)\n   fun setUint32(byteOffset: Int, value: Int, littleEndian: Boolean = definedExternally)\n   fun setFloat32(byteOffset: Int, value: Float, littleEndian: Boolean = definedExternally)\n   fun setFloat64(byteOffset: Int, value: Double, littleEndian: Boolean = definedExternally)\n}\n\npublic external interface BufferDataSource\n\npublic external interface

TexImageSource","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.clipboard\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\npublic external interface ClipboardEventInit : EventInit {\n   var clipboardData: DataTransfer? /* = null */\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ClipboardEventInit(clipboardData: DataTransfer? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ClipboardEventInit {\n   val o = js(\"({})\")\n   o[\"clipboardData\"] = clipboardData\n   o[\"bubbles\"] = bubbles\n   o[\"cancelable\"] = cancelable\n   o[\"composed\"] = composed\n   return o\n}\n\n/**\n * Exposes the JavaScript [ClipboardEvent](https://developer.mozilla.org/en/docs/Web/API/ClipboardEvent) to Kotlin\n */\npublic external open class ClipboardEvent(type: String, eventInitDict: ClipboardEventInit = definedExternally) : Event {\n   open val clipboardData: DataTransfer?\n\n   companion object {\n   val NONE: Short\n   val CAPTURING_PHASE: Short\n   val AT_TARGET: Short\n   val BUBBLING_PHASE: Short\n   }\n}\n\n/**\n * Exposes the JavaScript [Clipboard](https://developer.mozilla.org/en/docs/Web/API/Clipboard) to Kotlin\n */\npublic external abstract class Clipboard : EventTarget {\n   fun read(): Promise<DataTransfer>\n   fun readText(): Promise<String>\n   fun write(data: DataTransfer): Promise<Unit>\n   fun writeText(data: String): Promise<Unit>\n}\n\npublic external interface ClipboardPermissionDescriptor {\n   var allowWithoutGesture: Boolean? /* = false */\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ClipboardPermissionDescriptor(allowWithoutGesture: Boolean? = false): ClipboardPermissionDescriptor {\n   val o = js(\"({})\")\n   o[\"allowWithoutGesture\"] = allowWithoutGesture\n   return o\n}","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.css\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\n\npublic external abstract class MediaList : ItemArrayLike<String> {\n   open var mediaText: String\n   fun appendMedium(medium: String)\n   fun deleteMedium(medium: String)\n   override fun item(index: Int): String?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun MediaList.get(index: Int): String? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript [StyleSheet](https://developer.mozilla.org/en/docs/Web/API/StyleSheet) to Kotlin\n */\npublic external abstract class StyleSheet {\n   open val type: String\n   open val href: String?\n   open val ownerNode: UnionElementOrProcessingInstruction?\n   open val parentStyleSheet: StyleSheet?\n   open val title: String?\n   open val media: MediaList\n   open var disabled: Boolean\n}\n\n/**\n * Exposes the JavaScript [CSSStyleSheet](https://developer.mozilla.org/en/docs/Web/API/CSSStyleSheet) to Kotlin\n */\npublic external abstract class CSSStyleSheet : StyleSheet {\n   open val ownerRule: CSSRule?\n   open val cssRules: CSSRuleList\n   fun insertRule(rule: String, index: Int): Int\n   fun deleteRule(index: Int)\n}\n\n/**\n * Exposes the JavaScript [StyleSheetList](https://developer.mozilla.org/en/docs/Web/API/StyleSheetList) to Kotlin\n */\npublic external abstract class StyleSheetList : ItemArrayLike<StyleSheet> {\n   override fun item(index: Int): StyleSheet?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun StyleSheetList.get(index: Int): StyleSheet? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript [LinkStyle](https://developer.mozilla.org/en/docs/Web/API/LinkStyle) to Kotlin\n */\npublic external interface LinkStyle {\n   val sheet: StyleSheet?\n   get() = definedExternally\n}\n\n/**\n * Exposes the JavaScript [CSSRuleList](https://developer.mozilla.org/en/docs/Web/API/CSSRuleList) to Kotlin\n */\npublic external abstract

```
class CSSRuleList : ItemArrayLike<CSSRule> {\n    override fun item(index: Int): CSSRule?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun CSSRuleList.get(index: Int): CSSRule? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript [CSSRule](https://developer.mozilla.org/en/docs/Web/API/CSSRule) to Kotlin\n */\npublic external abstract class CSSRule {\n    open val type: Short\n    open var cssText: String\n    open val parentRule: CSSRule?\n    open val parentStyleSheet: CSSStyleSheet?\n\n    companion object {\n        val STYLE_RULE: Short\n        val CHARSET_RULE: Short\n        val IMPORT_RULE: Short\n        val MEDIA_RULE: Short\n        val FONT_FACE_RULE: Short\n        val PAGE_RULE: Short\n        val MARGIN_RULE: Short\n        val NAMESPACE_RULE: Short\n    }\n}\n\n/**\n * Exposes the JavaScript [CSSStyleRule](https://developer.mozilla.org/en/docs/Web/API/CSSStyleRule) to Kotlin\n */\npublic external abstract class CSSStyleRule : CSSRule {\n    open var selectorText: String\n    open val style: CSSStyleDeclaration\n\n    companion object {\n        val STYLE_RULE: Short\n        val CHARSET_RULE: Short\n        val IMPORT_RULE: Short\n        val MEDIA_RULE: Short\n        val FONT_FACE_RULE: Short\n        val PAGE_RULE: Short\n        val MARGIN_RULE: Short\n        val NAMESPACE_RULE: Short\n    }\n}\n\npublic external abstract class CSSImportRule : CSSRule {\n    open val href: String\n    open val media: MediaList\n    open val styleSheet: CSSStyleSheet\n\n    companion object {\n        val STYLE_RULE: Short\n        val CHARSET_RULE: Short\n        val IMPORT_RULE: Short\n        val MEDIA_RULE: Short\n        val FONT_FACE_RULE: Short\n        val PAGE_RULE: Short\n        val MARGIN_RULE: Short\n        val NAMESPACE_RULE: Short\n    }\n}\n\n/**\n * Exposes the JavaScript [CSSGroupingRule](https://developer.mozilla.org/en/docs/Web/API/CSSGroupingRule) to Kotlin\n */\npublic external abstract class CSSGroupingRule : CSSRule {\n    open val cssRules: CSSRuleList\n    fun insertRule(rule: String, index: Int): Int\n    fun deleteRule(index: Int)\n\n    companion object {\n        val STYLE_RULE: Short\n        val CHARSET_RULE: Short\n        val IMPORT_RULE: Short\n        val MEDIA_RULE: Short\n        val FONT_FACE_RULE: Short\n        val PAGE_RULE: Short\n        val MARGIN_RULE: Short\n        val NAMESPACE_RULE: Short\n    }\n}\n\n/**\n * Exposes the JavaScript [CSSMediaRule](https://developer.mozilla.org/en/docs/Web/API/CSSMediaRule) to Kotlin\n */\npublic external abstract class CSSMediaRule : CSSGroupingRule {\n    open val media: MediaList\n\n    companion object {\n        val STYLE_RULE: Short\n        val CHARSET_RULE: Short\n        val IMPORT_RULE: Short\n        val MEDIA_RULE: Short\n        val FONT_FACE_RULE: Short\n        val PAGE_RULE: Short\n        val MARGIN_RULE: Short\n        val NAMESPACE_RULE: Short\n    }\n}\n\n/**\n * Exposes the JavaScript [CSSPageRule](https://developer.mozilla.org/en/docs/Web/API/CSSPageRule) to Kotlin\n */\npublic external abstract class CSSPageRule : CSSGroupingRule {\n    open var selectorText: String\n    open val style: CSSStyleDeclaration\n\n    companion object {\n        val STYLE_RULE: Short\n        val CHARSET_RULE: Short\n        val IMPORT_RULE: Short\n        val MEDIA_RULE: Short\n        val FONT_FACE_RULE: Short\n        val PAGE_RULE: Short\n        val MARGIN_RULE: Short\n        val NAMESPACE_RULE: Short\n    }\n}\n\npublic external abstract class CSSMarginRule : CSSRule {\n    open val name: String\n    open val style: CSSStyleDeclaration\n\n    companion object {\n        val STYLE_RULE: Short\n        val CHARSET_RULE: Short\n        val IMPORT_RULE: Short\n        val MEDIA_RULE: Short\n        val FONT_FACE_RULE: Short\n        val PAGE_RULE: Short\n        val MARGIN_RULE: Short\n        val NAMESPACE_RULE: Short\n    }\n}\n\n/**\n * Exposes the JavaScript [CSSNamespaceRule](https://developer.mozilla.org/en/docs/Web/API/CSSNamespaceRule) to Kotlin\n */\npublic external abstract class CSSNamespaceRule : CSSRule {\n    open val namespaceURI: String\n    open val prefix: String\n\n    companion object {\n        val STYLE_RULE: Short\n        val CHARSET_RULE: Short\n        val IMPORT_RULE: Short\n        val MEDIA_RULE: Short\n        val FONT_FACE_RULE: Short\n        val PAGE_RULE:
```

Short\n val MARGIN_RULE: Short\n val NAMESPACE_RULE: Short\n }\n}\n\n/**\n * Exposes the JavaScript [CSSStyleDeclaration](https://developer.mozilla.org/en/docs/Web/API/CSSStyleDeclaration) to Kotlin\n */\npublic external abstract class CSSStyleDeclaration : ItemArrayLike<String> {\n open var cssText: String\n open val parentRule: CSSRule?\n open var cssFloat: String\n open var alignContent: String\n open var alignItems: String\n open var alignSelf: String\n open var animation: String\n open var animationDelay: String\n open var animationDirection: String\n open var animationDuration: String\n open var animationFillMode: String\n open var animationIterationCount: String\n open var animationName: String\n open var animationPlayState: String\n open var animationTimingFunction: String\n open var backfaceVisibility: String\n open var background: String\n open var backgroundAttachment: String\n open var backgroundClip: String\n open var backgroundColor: String\n open var backgroundImage: String\n open var backgroundOrigin: String\n open var backgroundPosition: String\n open var backgroundRepeat: String\n open var backgroundSize: String\n open var border: String\n open var borderBottom: String\n open var borderBottomColor: String\n open var borderBottomLeftRadius: String\n open var borderBottomRightRadius: String\n open var borderBottomStyle: String\n open var borderBottomWidth: String\n open var borderCollapse: String\n open var borderColor: String\n open var borderImage: String\n open var borderImageOutset: String\n open var borderImageRepeat: String\n open var borderImageSlice: String\n open var borderImageSource: String\n open var borderImageWidth: String\n open var borderLeft: String\n open var borderLeftColor: String\n open var borderLeftStyle: String\n open var borderLeftWidth: String\n open var borderRadius:
String\n open var borderRight: String\n open var borderRightColor: String\n open var borderRightStyle: String\n open var borderRightWidth: String\n open var borderSpacing: String\n open var borderStyle: String\n open var borderTop: String\n open var borderTopColor: String\n open var borderTopLeftRadius: String\n open var borderTopRightRadius: String\n open var borderTopStyle: String\n open var borderTopWidth: String\n open var borderWidth: String\n open var bottom: String\n open var boxDecorationBreak: String\n open var boxShadow: String\n open var boxSizing: String\n open var breakAfter: String\n open var breakBefore: String\n open var breakInside: String\n open var captionSide: String\n open var clear: String\n open var clip: String\n open var color: String\n open var columnCount: String\n open var columnFill: String\n open var columnGap: String\n open var columnRule: String\n open var columnRuleColor:
String\n open var columnRuleStyle: String\n open var columnRuleWidth: String\n open var columnSpan: String\n open var columnWidth: String\n open var columns: String\n open var content: String\n open var counterIncrement: String\n open var counterReset: String\n open var cursor: String\n open var direction: String\n open var display: String\n open var emptyCells: String\n open var filter: String\n open var flex: String\n open var flexBasis: String\n open var flexDirection: String\n open var flexFlow: String\n open var flexGrow: String\n open var flexShrink: String\n open var flexWrap: String\n open var font: String\n open var fontFamily: String\n open var fontFeatureSettings: String\n open var fontKerning: String\n open var fontLanguageOverride: String\n open var fontSize: String\n open var fontSizeAdjust: String\n open var fontStretch: String\n open var fontStyle: String\n open var fontSynthesis:
String\n open var fontVariant: String\n open var fontVariantAlternates: String\n open var fontVariantCaps: String\n open var fontVariantEastAsian: String\n open var fontVariantLigatures: String\n open var fontVariantNumeric: String\n open var fontVariantPosition: String\n open var fontWeight: String\n open var hangingPunctuation: String\n open var height: String\n open var hyphens: String\n open var imageOrientation: String\n open var imageRendering: String\n open var imageResolution: String\n open var imeMode: String\n open var justifyContent: String\n open var left: String\n open var letterSpacing: String\n open var lineBreak: String\n open var lineHeight: String\n open var listStyle: String\n open var listStyleImage: String\n open var listStylePosition: String\n open var listStyleType: String\n open var margin: String\n open var marginBottom: String\n open var marginLeft: String\n open var marginRight:
String\n open var marginTop: String\n open var mark: String\n open var markAfter: String\n open var markBefore: String\n open var marks: String\n open var marqueeDirection: String\n open var

marqueePlayCount: String\n   open var marqueeSpeed: String\n   open var marqueeStyle: String\n   open var mask: String\n   open var maskType: String\n   open var maxHeight: String\n   open var maxWidth: String\n   open var minHeight: String\n   open var minWidth: String\n   open var navDown: String\n   open var navIndex: String\n   open var navLeft: String\n   open var navRight: String\n   open var navUp: String\n   open var objectFit: String\n   open var objectPosition: String\n   open var opacity: String\n   open var order: String\n   open var orphans: String\n   open var outline: String\n   open var outlineColor: String\n   open var outlineOffset: String\n   open var outlineStyle: String\n   open var outlineWidth: String\n   open var overflowWrap: String\n   open var overflowX: String\n   open var overflowY: String\n   open var padding: String\n   open var paddingBottom: String\n   open var paddingLeft: String\n   open var paddingRight: String\n   open var paddingTop: String\n   open var pageBreakAfter: String\n   open var pageBreakBefore: String\n   open var pageBreakInside: String\n   open var perspective: String\n   open var perspectiveOrigin: String\n   open var phonemes: String\n   open var position: String\n   open var quotes: String\n   open var resize: String\n   open var rest: String\n   open var restAfter: String\n   open var restBefore: String\n   open var right: String\n   open var tabSize: String\n   open var tableLayout: String\n   open var textAlign: String\n   open var textAlignLast: String\n   open var textCombineUpright: String\n   open var textDecoration: String\n   open var textDecorationColor: String\n   open var textDecorationLine: String\n   open var textDecorationStyle: String\n   open var textIndent: String\n   open var textJustify: String\n   open var textOrientation: String\n   open var textOverflow: String\n   open var textShadow: String\n   open var textTransform: String\n   open var textUnderlinePosition: String\n   open var top: String\n   open var transform: String\n   open var transformOrigin: String\n   open var transformStyle: String\n   open var transition: String\n   open var transitionDelay: String\n   open var transitionDuration: String\n   open var transitionProperty: String\n   open var transitionTimingFunction: String\n   open var unicodeBidi: String\n   open var verticalAlign: String\n   open var visibility: String\n   open var voiceBalance: String\n   open var voiceDuration: String\n   open var voicePitch: String\n   open var voicePitchRange: String\n   open var voiceRate: String\n   open var voiceStress: String\n   open var voiceVolume: String\n   open var whiteSpace: String\n   open var widows: String\n   open var width: String\n   open var wordBreak: String\n   open var wordSpacing: String\n   open var wordWrap: String\n   open var writingMode: String\n   open var zIndex: String\n   open var _dashed_attribute: String\n   open var _camel_cased_attribute: String\n   open var _webkit_cased_attribute: String\n   fun getPropertyValue(property: String): String\n   fun getPropertyPriority(property: String): String\n   fun setProperty(property: String, value: String, priority: String = definedExternally)\n   fun setPropertyValue(property: String, value: String)\n   fun setPropertyPriority(property: String, priority: String)\n   fun removeProperty(property: String): String\n   override fun item(index: Int): String\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun CSSStyleDeclaration.get(index: Int): String? = asDynamic()[index]\n\npublic external interface ElementCSSInlineStyle {\n   val style: CSSStyleDeclaration\n}\n\n/**\n * Exposes the JavaScript [CSS](https://developer.mozilla.org/en/docs/Web/API/CSS) to Kotlin\n */\npublic external abstract class CSS {\n   companion object {\n   fun escape(ident: String): String\n   }\n}\n\npublic external interface UnionElementOrProcessingInstruction","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.encryptedmedia\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\n/**\n * Exposes the JavaScript [MediaKeySystemConfiguration](https://developer.mozilla.org/en/docs/Web/API/MediaKeySystemConfiguration) to Kotlin\n */\npublic external interface MediaKeySystemConfiguration {\n   var label: String? /* = \"\" */\n   get() = definedExternally\n   set(value) = definedExternally\n   var initDataTypes: Array<String>? /* = arrayOf() */\n   get() = definedExternally\n   set(value) = definedExternally\n   var audioCapabilities: Array<MediaKeySystemMediaCapability>? /* = arrayOf() */\n   get() = definedExternally\n   set(value) = definedExternally\n   var videoCapabilities: Array<MediaKeySystemMediaCapability>? /* = arrayOf() */\n

get() = definedExternally\n set(value) = definedExternally\n var distinctiveIdentifier: MediaKeysRequirement? /* = MediaKeysRequirement.OPTIONAL */\n get() = definedExternally\n set(value) = definedExternally\n var persistentState: MediaKeysRequirement? /* = MediaKeysRequirement.OPTIONAL */\n get() = definedExternally\n set(value) = definedExternally\n var sessionTypes: Array<String>?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaKeySystemConfiguration(label: String? = \"\", initDataTypes: Array<String>? = arrayOf(), audioCapabilities: Array<MediaKeySystemMediaCapability>? = arrayOf(), videoCapabilities: Array<MediaKeySystemMediaCapability>? = arrayOf(), distinctiveIdentifier: MediaKeysRequirement? = MediaKeysRequirement.OPTIONAL, persistentState: MediaKeysRequirement? = MediaKeysRequirement.OPTIONAL, sessionTypes: Array<String>? = undefined): MediaKeySystemConfiguration {\n val o = js(\"({})\")\n o[\"label\"] = label\n o[\"initDataTypes\"] = initDataTypes\n o[\"audioCapabilities\"] = audioCapabilities\n o[\"videoCapabilities\"] = videoCapabilities\n o[\"distinctiveIdentifier\"] = distinctiveIdentifier\n o[\"persistentState\"] = persistentState\n o[\"sessionTypes\"] = sessionTypes\n return o\n}\n\npublic external interface MediaKeySystemMediaCapability {\n var contentType: String? /* = \"\" */\n get() = definedExternally\n set(value) = definedExternally\n var robustness: String? /* = \"\" */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaKeySystemMediaCapability(contentType: String? = \"\", robustness: String? = \"\"): MediaKeySystemMediaCapability {\n val o = js(\"({})\")\n o[\"contentType\"] = contentType\n o[\"robustness\"] = robustness\n return o\n}\n\n/**\n * Exposes the JavaScript [MediaKeySystemAccess](https://developer.mozilla.org/en/docs/Web/API/MediaKeySystemAccess) to Kotlin\n */\npublic external abstract class MediaKeySystemAccess {\n open val keySystem: String\n fun getConfiguration(): MediaKeySystemConfiguration\n fun createMediaKeys(): Promise<MediaKeys>\n}\n\n/**\n * Exposes the JavaScript [MediaKeys](https://developer.mozilla.org/en/docs/Web/API/MediaKeys) to Kotlin\n */\npublic external abstract class MediaKeys {\n fun createSession(sessionType: MediaKeySessionType = definedExternally): MediaKeySession\n fun setServerCertificate(serverCertificate: dynamic): Promise<Boolean>\n}\n\n/**\n * Exposes the JavaScript [MediaKeySession](https://developer.mozilla.org/en/docs/Web/API/MediaKeySession) to Kotlin\n */\npublic external abstract class MediaKeySession : EventTarget {\n open val sessionId: String\n open val expiration: Double\n open val closed: Promise<Unit>\n open val keyStatuses: MediaKeyStatusMap\n open var onkeystatuseschange: ((Event) -> dynamic)?\n open var onmessage: ((MessageEvent) -> dynamic)?\n fun generateRequest(initDataType: String, initData: dynamic): Promise<Unit>\n fun load(sessionId: String): Promise<Boolean>\n fun update(response: dynamic): Promise<Unit>\n fun close(): Promise<Unit>\n fun remove(): Promise<Unit>\n}\n\n/**\n * Exposes the JavaScript [MediaKeyStatusMap](https://developer.mozilla.org/en/docs/Web/API/MediaKeyStatusMap) to Kotlin\n */\npublic external abstract class MediaKeyStatusMap {\n open val size: Int\n fun has(keyId: dynamic): Boolean\n fun get(keyId: dynamic): Any?\n}\n\n/**\n * Exposes the JavaScript [MediaKeyMessageEvent](https://developer.mozilla.org/en/docs/Web/API/MediaKeyMessageEvent) to Kotlin\n */\npublic external open class MediaKeyMessageEvent(type: String, eventInitDict: MediaKeyMessageEventInit) : Event {\n open val messageType: MediaKeyMessageType\n open val message: ArrayBuffer\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface MediaKeyMessageEventInit : EventInit {\n var messageType: MediaKeyMessageType?\n var message: ArrayBuffer?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun

MediaKeyMessageEventInit(messageType: MediaKeyMessageType?, message:
ArrayBuffer?, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
MediaKeyMessageEventInit {\n val o = js(\"({})\")\n o[\"messageType\"] = messageType\n o[\"message\"] =
message\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return
o\n}\n\npublic external open class MediaEncryptedEvent(type: String, eventInitDict: MediaEncryptedEventInit =
definedExternally) : Event {\n open val initDataType: String\n open val initData: ArrayBuffer?\n\n companion
object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val
BUBBLING_PHASE: Short\n }\n}\n\npublic external interface MediaEncryptedEventInit : EventInit {\n var
initDataType: String? /* = \"\" */\n get() = definedExternally\n set(value) = definedExternally\n var
initData: ArrayBuffer? /* = null */\n get() = definedExternally\n set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun
MediaEncryptedEventInit(initDataType: String? = \"\", initData: ArrayBuffer? = null, bubbles: Boolean? = false,
cancelable: Boolean? = false, composed: Boolean? = false): MediaEncryptedEventInit {\n val o = js(\"({})\")\n
o[\"initDataType\"] = initDataType\n o[\"initData\"] = initData\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"]
= cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/* please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface MediaKeysRequirement {\n companion object\n}\n\npublic inline val
MediaKeysRequirement.Companion.REQUIRED: MediaKeysRequirement get() =
\"required\".asDynamic().unsafeCast<MediaKeysRequirement>()\n\npublic inline val
MediaKeysRequirement.Companion.OPTIONAL: MediaKeysRequirement get() =
\"optional\".asDynamic().unsafeCast<MediaKeysRequirement>()\n\npublic
inline val MediaKeysRequirement.Companion.NOT_ALLOWED: MediaKeysRequirement get() = \"not-
allowed\".asDynamic().unsafeCast<MediaKeysRequirement>()\n\n/* please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface MediaKeySessionType {\n companion object\n}\n\npublic inline val
MediaKeySessionType.Companion.TEMPORARY: MediaKeySessionType get() =
\"temporary\".asDynamic().unsafeCast<MediaKeySessionType>()\n\npublic inline val
MediaKeySessionType.Companion.PERSISTENT_LICENSE: MediaKeySessionType get() = \"persistent-
license\".asDynamic().unsafeCast<MediaKeySessionType>()\n\n/* please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface MediaKeyStatus {\n companion object\n}\n\npublic inline val MediaKeyStatus.Companion.USABLE:
MediaKeyStatus get() = \"usable\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic
inline val MediaKeyStatus.Companion.EXPIRED: MediaKeyStatus get() =
\"expired\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.RELEASED: MediaKeyStatus get() =
\"released\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.OUTPUT_RESTRICTED: MediaKeyStatus get() = \"output-
restricted\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.OUTPUT_DOWNSCALED: MediaKeyStatus get() = \"output-
downscaled\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.STATUS_PENDING: MediaKeyStatus get() = \"status-
pending\".asDynamic().unsafeCast<MediaKeyStatus>()\n\npublic inline val
MediaKeyStatus.Companion.INTERNAL_ERROR: MediaKeyStatus get() = \"internal-
error\".asDynamic().unsafeCast<MediaKeyStatus>()\n\n/* please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic
external interface MediaKeyMessageType {\n companion object\n}\n\npublic inline val
MediaKeyMessageType.Companion.LICENSE_REQUEST: MediaKeyMessageType get() = \"license-

```
request\".asDynamic().unsafeCast<MediaKeyMessageType>()\n\npublic inline val
MediaKeyMessageType.Companion.LICENSE_RENEWAL: MediaKeyMessageType get() = \"license-
renewal\".asDynamic().unsafeCast<MediaKeyMessageType>()\n\npublic inline val
MediaKeyMessageType.Companion.LICENSE_RELEASE: MediaKeyMessageType get() = \"license-
release\".asDynamic().unsafeCast<MediaKeyMessageType>()\n\npublic inline val
MediaKeyMessageType.Companion.INDIVIDUALIZATION_REQUEST: MediaKeyMessageType get() =
\"individualization-request\".asDynamic().unsafeCast<MediaKeyMessageType>()","/*\n * Copyright 2010-2021
JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the
Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-
GENERATED, DO NOT EDIT!\n//
 See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.events\n\nimport kotlin.js.*\nimport
org.khronos.webgl.*\nimport org.w3c.dom.*\n\n/**\n * Exposes the JavaScript
[UIEvent](https://developer.mozilla.org/en/docs/Web/API/UIEvent) to Kotlin\n */\npublic external open class
UIEvent(type: String, eventInitDict: UIEventInit = definedExternally) : Event {\n    open val view: Window?\n
open val detail: Int\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n
val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface UIEventInit :
EventInit {\n    var view: Window? /* = null */\n        get() = definedExternally\n        set(value) =
definedExternally\n    var detail: Int? /* = 0 */\n        get() = definedExternally\n        set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun UIEventInit(view: Window? =
 null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false):
UIEventInit {\n    val o = js(\"({})\")\n    o[\"view\"] = view\n    o[\"detail\"] = detail\n    o[\"bubbles\"] = bubbles\n
o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\n/**\n * Exposes the JavaScript
[FocusEvent](https://developer.mozilla.org/en/docs/Web/API/FocusEvent) to Kotlin\n */\npublic external open class
FocusEvent(type: String, eventInitDict: FocusEventInit = definedExternally) : UIEvent {\n    open val relatedTarget:
EventTarget?\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val
AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface FocusEventInit :
UIEventInit {\n    var relatedTarget: EventTarget? /* = null */\n        get() = definedExternally\n        set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic
 inline fun FocusEventInit(relatedTarget: EventTarget? = null, view: Window? = null, detail: Int? = 0, bubbles:
Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): FocusEventInit {\n    val o =
js(\"({})\")\n    o[\"relatedTarget\"] = relatedTarget\n    o[\"view\"] = view\n    o[\"detail\"] = detail\n
o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return
o\n}\n\n/**\n * Exposes the JavaScript [MouseEvent](https://developer.mozilla.org/en/docs/Web/API/MouseEvent)
to Kotlin\n */\npublic external open class MouseEvent(type: String, eventInitDict: MouseEventInit =
definedExternally) : UIEvent, UnionElementOrMouseEvent {\n    open val screenX: Int\n    open val screenY: Int\n
 open val clientX: Int\n    open val clientY: Int\n    open val ctrlKey: Boolean\n    open val shiftKey: Boolean\n
open val altKey: Boolean\n    open val metaKey: Boolean\n    open val button: Short\n
    open val buttons: Short\n    open val relatedTarget: EventTarget?\n    open val region: String?\n    open val pageX:
Double\n    open val pageY: Double\n    open val x: Double\n    open val y: Double\n    open val offsetX: Double\n
open val offsetY: Double\n    fun getModifierState(keyArg: String): Boolean\n\n    companion object {\n        val
NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val
BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface MouseEventInit : EventModifierInit {\n    var
screenX: Int? /* = 0 */\n        get() = definedExternally\n        set(value) = definedExternally\n    var screenY: Int? /*
= 0 */\n        get() = definedExternally\n        set(value) = definedExternally\n    var clientX: Int? /* = 0 */\n
get() = definedExternally\n        set(value) = definedExternally\n    var clientY: Int? /* = 0 */\n        get() =
definedExternally\n        set(value) = definedExternally\n    var button: Short?
```

/* = 0 */\n get() = definedExternally\n set(value) = definedExternally\n var buttons: Short? /* = 0 */\n get() = definedExternally\n set(value) = definedExternally\n var relatedTarget: EventTarget? /* = null */\n get() = definedExternally\n set(value) = definedExternally\n var region: String? /* = null */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MouseEventInit(screenX: Int? = 0, screenY: Int? = 0, clientX: Int? = 0, clientY: Int? = 0, button: Short? = 0, buttons: Short? = 0, relatedTarget: EventTarget? = null, region: String? = null, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): MouseEventInit {\n val o = js(\"({})\")\n o[\"screenX\"] = screenX\n o[\"screenY\"] = screenY\n o[\"clientX\"] = clientX\n o[\"clientY\"] = clientY\n o[\"button\"] = button\n o[\"buttons\"] = buttons\n o[\"relatedTarget\"] = relatedTarget\n o[\"region\"] = region\n o[\"ctrlKey\"] = ctrlKey\n o[\"shiftKey\"] = shiftKey\n o[\"altKey\"] = altKey\n o[\"metaKey\"] = metaKey\n o[\"modifierAltGraph\"] = modifierAltGraph\n o[\"modifierCapsLock\"] = modifierCapsLock\n o[\"modifierFn\"] = modifierFn\n o[\"modifierFnLock\"] = modifierFnLock\n o[\"modifierHyper\"] = modifierHyper\n o[\"modifierNumLock\"] = modifierNumLock\n o[\"modifierScrollLock\"] = modifierScrollLock\n o[\"modifierSuper\"] = modifierSuper\n o[\"modifierSymbol\"] = modifierSymbol\n o[\"modifierSymbolLock\"] = modifierSymbolLock\n o[\"view\"] = view\n o[\"detail\"] = detail\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\npublic external interface EventModifierInit : UIEventInit {\n var ctrlKey: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var shiftKey: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var altKey: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var metaKey: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var modifierAltGraph: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var modifierCapsLock: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var modifierFn: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var modifierFnLock: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var modifierHyper: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var modifierNumLock: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var modifierScrollLock: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var modifierSuper: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var modifierSymbol: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var modifierSymbolLock: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun EventModifierInit(ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): EventModifierInit {\n val o = js(\"({})\")\n o[\"ctrlKey\"] = ctrlKey\n o[\"shiftKey\"] = shiftKey\n o[\"altKey\"] = altKey\n o[\"metaKey\"] = metaKey\n o[\"modifierAltGraph\"]

= modifierAltGraph\n o[\"modifierCapsLock\"] = modifierCapsLock\n o[\"modifierFn\"] = modifierFn\n o[\"modifierFnLock\"] = modifierFnLock\n o[\"modifierHyper\"] = modifierHyper\n o[\"modifierNumLock\"] = modifierNumLock\n o[\"modifierScrollLock\"] = modifierScrollLock\n o[\"modifierSuper\"] = modifierSuper\n o[\"modifierSymbol\"] = modifierSymbol\n o[\"modifierSymbolLock\"] = modifierSymbolLock\n o[\"view\"] = view\n o[\"detail\"] = detail\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript [WheelEvent](https://developer.mozilla.org/en/docs/Web/API/WheelEvent) to Kotlin\n */\npublic external open class WheelEvent(type: String, eventInitDict: WheelEventInit = definedExternally) : MouseEvent {\n open val deltaX: Double\n open val deltaY: Double\n open val deltaZ: Double\n open val deltaMode: Int\n\n companion object {\n val DOM_DELTA_PIXEL: Int\n val DOM_DELTA_LINE: Int\n val DOM_DELTA_PAGE: Int\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface WheelEventInit : MouseEventInit {\n var deltaX: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var deltaY: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var deltaZ: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var deltaMode: Int? /* = 0 */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun WheelEventInit(deltaX: Double? = 0.0, deltaY: Double? = 0.0, deltaZ: Double? = 0.0, deltaMode: Int? = 0, screenX: Int? = 0, screenY: Int? = 0, clientX: Int? = 0, clientY: Int? = 0, button: Short? = 0, buttons: Short? = 0, relatedTarget: EventTarget? = null, region: String? = null, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): WheelEventInit {\n val o = js(\"({})\")\n o[\"deltaX\"] = deltaX\n o[\"deltaY\"] = deltaY\n o[\"deltaZ\"] = deltaZ\n o[\"deltaMode\"] = deltaMode\n o[\"screenX\"] = screenX\n o[\"screenY\"] = screenY\n o[\"clientX\"] = clientX\n o[\"clientY\"] = clientY\n o[\"button\"] = button\n o[\"buttons\"] = buttons\n o[\"relatedTarget\"] = relatedTarget\n o[\"region\"] = region\n o[\"ctrlKey\"] = ctrlKey\n o[\"shiftKey\"] = shiftKey\n o[\"altKey\"] = altKey\n o[\"metaKey\"] = metaKey\n o[\"modifierAltGraph\"] = modifierAltGraph\n o[\"modifierCapsLock\"] = modifierCapsLock\n o[\"modifierFn\"] = modifierFn\n o[\"modifierFnLock\"] = modifierFnLock\n o[\"modifierHyper\"] = modifierHyper\n o[\"modifierNumLock\"] = modifierNumLock\n o[\"modifierScrollLock\"] = modifierScrollLock\n o[\"modifierSuper\"] = modifierSuper\n o[\"modifierSymbol\"] = modifierSymbol\n o[\"modifierSymbolLock\"] = modifierSymbolLock\n o[\"view\"] = view\n o[\"detail\"] = detail\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript [InputEvent](https://developer.mozilla.org/en/docs/Web/API/InputEvent) to Kotlin\n */\npublic external open class InputEvent(type: String, eventInitDict: InputEventInit = definedExternally) : UIEvent {\n open val data: String\n open val isComposing: Boolean\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface InputEventInit : UIEventInit {\n var data: String? /* = \"\" */\n get() = definedExternally\n set(value) = definedExternally\n var isComposing: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun InputEventInit(data: String? = \"\", isComposing: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): InputEventInit {\n val o = js(\"({})\")\n o[\"data\"] = data\n o[\"isComposing\"]

= isComposing\n o[\"view\"] = view\n o[\"detail\"] = detail\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript [KeyboardEvent](https://developer.mozilla.org/en/docs/Web/API/KeyboardEvent) to Kotlin\n */\npublic external open class KeyboardEvent(type: String, eventInitDict: KeyboardEventInit = definedExternally) : UIEvent {\n open val key: String\n open val code: String\n open val location: Int\n open val ctrlKey: Boolean\n open val shiftKey: Boolean\n open val altKey: Boolean\n open val metaKey: Boolean\n open val repeat: Boolean\n open val isComposing: Boolean\n open val charCode: Int\n open val keyCode: Int\n open val which: Int\n fun getModifierState(keyArg: String): Boolean\n\n companion object {\n val DOM_KEY_LOCATION_STANDARD: Int\n val DOM_KEY_LOCATION_LEFT: Int\n val DOM_KEY_LOCATION_RIGHT: Int\n
val DOM_KEY_LOCATION_NUMPAD: Int\n val NONE: Short\n val CAPTURING_PHASE: Short\n
val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface KeyboardEventInit : EventModifierInit {\n var key: String? /* = \"\" */\n get() = definedExternally\n set(value) = definedExternally\n var code: String? /* = \"\" */\n get() = definedExternally\n set(value) = definedExternally\n var location: Int? /* = 0 */\n get() = definedExternally\n set(value) = definedExternally\n var repeat: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var isComposing: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun KeyboardEventInit(key: String? = \"\", code: String? = \"\", location: Int? = 0, repeat:
Boolean? = false, isComposing: Boolean? = false, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): KeyboardEventInit {\n val o = js(\"({})\")\n o[\"key\"] = key\n o[\"code\"] = code\n o[\"location\"] = location\n o[\"repeat\"] = repeat\n o[\"isComposing\"] = isComposing\n o[\"ctrlKey\"] = ctrlKey\n o[\"shiftKey\"] = shiftKey\n o[\"altKey\"] = altKey\n o[\"metaKey\"] = metaKey\n o[\"modifierAltGraph\"] = modifierAltGraph\n o[\"modifierCapsLock\"]
= modifierCapsLock\n o[\"modifierFn\"] = modifierFn\n o[\"modifierFnLock\"] = modifierFnLock\n o[\"modifierHyper\"] = modifierHyper\n o[\"modifierNumLock\"] = modifierNumLock\n o[\"modifierScrollLock\"] = modifierScrollLock\n o[\"modifierSuper\"] = modifierSuper\n o[\"modifierSymbol\"] = modifierSymbol\n o[\"modifierSymbolLock\"] = modifierSymbolLock\n o[\"view\"] = view\n o[\"detail\"] = detail\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript [CompositionEvent](https://developer.mozilla.org/en/docs/Web/API/CompositionEvent) to Kotlin\n */\npublic external open class CompositionEvent(type: String, eventInitDict: CompositionEventInit = definedExternally) : UIEvent {\n open val data: String\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface CompositionEventInit : UIEventInit {\n var data: String? /* = \"\" */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun CompositionEventInit(data: String? = \"\", view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): CompositionEventInit {\n val o = js(\"({})\")\n o[\"data\"] = data\n o[\"view\"] = view\n o[\"detail\"] = detail\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript [Event](https://developer.mozilla.org/en/docs/Web/API/Event) to Kotlin\n */\npublic external open class

Event(type: String, eventInitDict: EventInit = definedExternally) {\n   open val type: String\n   open val target: EventTarget?\n   open val currentTarget:
EventTarget?\n   open val eventPhase: Short\n   open val bubbles: Boolean\n   open val cancelable: Boolean\n   open val defaultPrevented: Boolean\n   open val composed: Boolean\n   open val isTrusted: Boolean\n   open val timeStamp: Number\n   fun composedPath(): Array<EventTarget>\n   fun stopPropagation()\n   fun stopImmediatePropagation()\n   fun preventDefault()\n   fun initEvent(type: String, bubbles: Boolean, cancelable: Boolean)\n\n   companion object {\n   val NONE: Short\n   val CAPTURING_PHASE: Short\n   val AT_TARGET: Short\n   val BUBBLING_PHASE: Short\n   }\n}\n\n/**\n * Exposes the JavaScript [EventTarget](https://developer.mozilla.org/en/docs/Web/API/EventTarget) to Kotlin\n */\npublic external abstract class EventTarget {\n   fun addEventListener(type: String, callback: EventListener?, options: dynamic = definedExternally)\n   fun addEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic = definedExternally)\n
    fun removeEventListener(type: String, callback: EventListener?, options: dynamic = definedExternally)\n   fun removeEventListener(type: String, callback: ((Event) -> Unit)?, options: dynamic = definedExternally)\n   fun dispatchEvent(event: Event): Boolean\n}\n\n/**\n * Exposes the JavaScript [EventListener](https://developer.mozilla.org/en/docs/Web/API/EventListener) to Kotlin\n */\npublic external interface EventListener {\n   fun handleEvent(event: Event)\n}","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.dom\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.clipboard.*\nimport org.w3c.dom.css.*\nimport org.w3c.dom.encryptedmedia.*\nimport org.w3c.dom.events.*\nimport org.w3c.dom.mediacapture.*\nimport org.w3c.dom.mediasource.*\nimport org.w3c.dom.pointerevents.*\nimport org.w3c.dom.svg.*\nimport org.w3c.fetch.*\nimport org.w3c.files.*\nimport org.w3c.performance.*\nimport org.w3c.workers.*\nimport org.w3c.xhr.*\n\npublic external abstract class HTMLAllCollection {\n   open val length: Int\n   fun item(nameOrIndex: String = definedExternally): UnionElementOrHTMLCollection?\n   fun namedItem(name: String): UnionElementOrHTMLCollection?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun HTMLAllCollection.get(index: Int): Element? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun HTMLAllCollection.get(name: String): UnionElementOrHTMLCollection? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript [HTMLFormControlsCollection](https://developer.mozilla.org/en/docs/Web/API/HTMLFormControlsCollection) to
Kotlin\n */\npublic external abstract class HTMLFormControlsCollection : HTMLCollection\n\n/**\n * Exposes the JavaScript [RadioNodeList](https://developer.mozilla.org/en/docs/Web/API/RadioNodeList) to Kotlin\n */\npublic external abstract class RadioNodeList : NodeList, UnionElementOrRadioNodeList {\n   open var value: String\n}\n\n/**\n * Exposes the JavaScript [HTMLOptionsCollection](https://developer.mozilla.org/en/docs/Web/API/HTMLOptionsCollection) to Kotlin\n */\npublic external abstract class HTMLOptionsCollection : HTMLCollection {\n   override var length: Int\n   open var selectedIndex: Int\n   fun add(element: UnionHTMLOptGroupElementOrHTMLOptionElement, before: dynamic = definedExternally)\n   fun remove(index: Int)\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun HTMLOptionsCollection.set(index: Int, option: HTMLOptionElement?) { asDynamic()[index] = option }\n\n/**\n * Exposes the JavaScript [HTMLElement](https://developer.mozilla.org/en/docs/Web/API/HTMLElement)
to Kotlin\n */\npublic external abstract class HTMLElement : Element, GlobalEventHandlers, DocumentAndElementEventHandlers, ElementContentEditable, ElementCSSInlineStyle {\n   open var title:

String\n   open var lang: String\n   open var translate: Boolean\n   open var dir: String\n   open val dataset: DOMStringMap\n   open var hidden: Boolean\n   open var tabIndex: Int\n   open var accessKey: String\n   open val accessKeyLabel: String\n   open var draggable: Boolean\n   open val dropzone: DOMTokenList\n   open var contextMenu: HTMLMenuElement?\n   open var spellcheck: Boolean\n   open var innerText: String\n   open val offsetParent: Element?\n   open val offsetTop: Int\n   open val offsetLeft: Int\n   open val offsetWidth: Int\n   open val offsetHeight: Int\n   fun click()\n   fun focus()\n   fun blur()\n   fun forceSpellCheck()\n\n   companion object {\n   val ELEMENT_NODE: Short\n
    val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n
    val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n   val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n   val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n   }\n}\n\n/**\n * Exposes the JavaScript [HTMLUnknownElement](https://developer.mozilla.org/en/docs/Web/API/HTMLUnknownElement) to Kotlin\n */\npublic external abstract class HTMLUnknownElement : HTMLElement {\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE:
 Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n   val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n   val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n   val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n   }\n}\n\n/**\n * Exposes the JavaScript [DOMStringMap](https://developer.mozilla.org/en/docs/Web/API/DOMStringMap) to Kotlin\n */\npublic external abstract class DOMStringMap\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun DOMStringMap.get(name: String): String? = asDynamic()[name]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun DOMStringMap.set(name: String, value: String) { asDynamic()[name] = value }\n\n/**\n * Exposes the JavaScript [HTMLHtmlElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHtmlElement) to Kotlin\n */\npublic external abstract class HTMLHtmlElement : HTMLElement {\n   open var version: String\n\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n   val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n   val DOCUMENT_POSITION_FOLLOWING: Short\n   val DOCUMENT_POSITION_CONTAINS: Short\n   val DOCUMENT_POSITION_CONTAINED_BY: Short\n   val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n   }\n}\n\n/**\n * Exposes the JavaScript [HTMLHeadElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHeadElement) to Kotlin\n */\npublic external abstract class HTMLHeadElement : HTMLElement {\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val

CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLTitleElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTitleElement) to Kotlin\n */\npublic external abstract class HTMLTitleElement : HTMLElement {\n open var text: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLBaseElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBaseElement) to Kotlin\n */\npublic external abstract class HTMLBaseElement : HTMLElement {\n open var href: String\n open var target: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLLinkElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLinkElement) to Kotlin\n */\npublic external abstract class HTMLLinkElement : HTMLElement, LinkStyle {\n open var href: String\n open var crossOrigin: String?\n open var rel: String\n open var `as`: RequestDestination\n open val relList: DOMTokenList\n open var media: String\n open var nonce: String\n open var hreflang: String\n open var type: String\n open val sizes: DOMTokenList\n open var referrerPolicy: String\n open var charset: String\n open var rev: String\n open var target: String\n open var scope: String\n open var workerType: WorkerType\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript

[HTMLMetaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMetaElement) to Kotlin\n */\npublic external abstract class HTMLMetaElement : HTMLElement {\n open var name: String\n open var httpEquiv: String\n open var content: String\n open var scheme: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLStyleElement](https://developer.mozilla.org/en/docs/Web/API/HTMLStyleElement) to Kotlin\n */\npublic external abstract class HTMLStyleElement : HTMLElement, LinkStyle {\n open var media: String\n open var nonce: String\n open var type: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLBodyElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBodyElement) to Kotlin\n */\npublic external abstract class HTMLBodyElement : HTMLElement, WindowEventHandlers {\n open var text: String\n open var link: String\n open var vLink: String\n open var aLink: String\n open var bgColor: String\n open var background: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLHeadingElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHeadingElement) to Kotlin\n */\npublic external abstract class HTMLHeadingElement : HTMLElement {\n open var align: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript

[HTMLParagraphElement](https://developer.mozilla.org/en/docs/Web/API/HTMLParagraphElement) to Kotlin\n */\npublic external abstract class HTMLParagraphElement : HTMLElement {\n open var align: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLHRElement](https://developer.mozilla.org/en/docs/Web/API/HTMLHRElement) to Kotlin\n */\npublic external abstract class HTMLHRElement : HTMLElement {\n open var align: String\n open var color: String\n open var noShade: Boolean\n open var size: String\n open var width: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLPreElement](https://developer.mozilla.org/en/docs/Web/API/HTMLPreElement) to Kotlin\n */\npublic external abstract class HTMLPreElement : HTMLElement {\n open var width: Int\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLQuoteElement](https://developer.mozilla.org/en/docs/Web/API/HTMLQuoteElement) to Kotlin\n */\npublic external abstract class HTMLQuoteElement : HTMLElement {\n open var cite: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript

[HTMLOListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOListElement) to Kotlin\n */\npublic external abstract class HTMLOListElement : HTMLElement {\n open var reversed: Boolean\n open var start: Int\n open var type: String\n open var compact: Boolean\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE:
Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLUListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLUListElement) to Kotlin\n */\npublic external abstract class HTMLUListElement : HTMLElement {\n open var compact: Boolean\n open var type: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE:
Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLLIElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLIElement) to Kotlin\n */\npublic external abstract class HTMLLIElement : HTMLElement {\n open var value: Int\n open var type: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n
val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLDListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDListElement) to Kotlin\n */\npublic external abstract class HTMLDListElement : HTMLElement {\n open var compact: Boolean\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE:
Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLDivElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDivElement) to Kotlin\n */\npublic external abstract class HTMLDivElement : HTMLElement {\n open var align: String\n\n companion

object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE:
Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLAnchorElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAnchorElement) to Kotlin\n */\npublic external abstract class HTMLAnchorElement : HTMLElement, HTMLHyperlinkElementUtils {\n open var target: String\n open var download: String\n open var ping: String\n open var rel: String\n open val relList: DOMTokenList\n open var hreflang: String\n open var type: String\n open var text: String\n open var referrerPolicy: String\n open var coords: String\n open var charset:
String\n open var name: String\n open var rev: String\n open var shape: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLDataElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDataElement) to Kotlin\n */\npublic external abstract class HTMLDataElement : HTMLElement {\n open var value: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLTimeElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTimeElement) to Kotlin\n */\npublic external abstract class HTMLTimeElement : HTMLElement {\n open var dateTime: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLSpanElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSpanElement)

to Kotlin\n */\npublic external abstract class HTMLSpanElement : HTMLElement {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLBRElement](https://developer.mozilla.org/en/docs/Web/API/HTMLBRElement) to Kotlin\n */\npublic external abstract class HTMLBRElement : HTMLElement {\n open var clear: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLHyperlinkElementUtils](https://developer.mozilla.org/en/docs/Web/API/HTMLHyperlinkElementUtils) to Kotlin\n */\npublic external interface HTMLHyperlinkElementUtils {\n var href: String\n val origin: String\n var protocol: String\n var username: String\n var password: String\n var host: String\n var hostname: String\n var port: String\n var pathname: String\n var search: String\n var hash: String\n}\n\n/**\n * Exposes the JavaScript [HTMLModElement](https://developer.mozilla.org/en/docs/Web/API/HTMLModElement) to Kotlin\n */\npublic external abstract class HTMLModElement : HTMLElement {\n open var cite: String\n open var dateTime: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE:
Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLPictureElement](https://developer.mozilla.org/en/docs/Web/API/HTMLPictureElement) to Kotlin\n */\npublic external abstract class HTMLPictureElement : HTMLElement {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE:
Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes

the JavaScript [HTMLSourceElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSourceElement) to Kotlin\n */\npublic external abstract class HTMLSourceElement : HTMLElement {\n open var src: String\n open var type: String\n open var srcset: String\n open var sizes: String\n open var media: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE:
Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLImageElement](https://developer.mozilla.org/en/docs/Web/API/HTMLImageElement) to Kotlin\n */\npublic external abstract class HTMLImageElement : HTMLElement, HTMLOrSVGImageElement, TexImageSource {\n open var alt: String\n open var src: String\n open var srcset: String\n open var sizes: String\n open var crossOrigin: String?\n open var useMap: String\n open var isMap: Boolean\n open var width: Int\n open var height: Int\n open val naturalWidth:
Int\n open val naturalHeight: Int\n open val complete: Boolean\n open val currentSrc: String\n open var referrerPolicy: String\n open var name: String\n open var lowsrc: String\n open var align: String\n open var hspace: Int\n open var vspace: Int\n open var longDesc: String\n open var border: String\n open val x: Int\n open val y: Int\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLIFrameElement](https://developer.mozilla.org/en/docs/Web/API/HTMLIFrameElement) to Kotlin\n */\npublic external abstract class HTMLIFrameElement : HTMLElement {\n open var src: String\n open var srcdoc: String\n open var name: String\n open val sandbox: DOMTokenList\n open var allowFullscreen: Boolean\n open var allowUserMedia: Boolean\n open var width: String\n open var height: String\n open var referrerPolicy: String\n open val contentDocument: Document?\n open val contentWindow: Window?\n open var align: String\n open var scrolling: String\n open var frameBorder: String\n open var longDesc: String\n open var marginHeight: String\n open var marginWidth: String\n fun getSVGDocument(): Document?\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript

[HTMLEmbedElement](https://developer.mozilla.org/en/docs/Web/API/HTMLEmbedElement) to Kotlin\n */\npublic external abstract class HTMLEmbedElement : HTMLElement {\n open var src: String\n open var type: String\n
open var width: String\n open var height: String\n open var align: String\n open var name: String\n fun getSVGDocument(): Document?\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLObjectElement](https://developer.mozilla.org/en/docs/Web/API/HTMLObjectElement)
to Kotlin\n */\npublic external abstract class HTMLObjectElement : HTMLElement {\n open var data: String\n open var type: String\n open var typeMustMatch: Boolean\n open var name: String\n open var useMap: String\n open val form: HTMLFormElement?\n open var width: String\n open var height: String\n open val contentDocument: Document?\n open val contentWindow: Window?\n open val willValidate: Boolean\n open val validity: ValidityState\n open val validationMessage: String\n open var align: String\n open var archive: String\n open var code: String\n open var declare: Boolean\n open var hspace: Int\n open var standby: String\n open var vspace: Int\n open var codeBase: String\n open var codeType: String\n open var border: String\n fun getSVGDocument(): Document?\n fun checkValidity(): Boolean\n fun reportValidity(): Boolean\n fun setCustomValidity(error: String)\n\n
companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLParamElement](https://developer.mozilla.org/en/docs/Web/API/HTMLParamElement) to Kotlin\n */\npublic external abstract class HTMLParamElement : HTMLElement {\n open var name:
String\n open var value: String\n open var type: String\n open var valueType: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLVideoElement](https://developer.mozilla.org/en/docs/Web/API/HTMLVideoElement)
to Kotlin\n */\npublic external abstract class HTMLVideoElement : HTMLMediaElement, CanvasImageSource, TexImageSource {\n open var width: Int\n open var height: Int\n open val videoWidth: Int\n open val

videoHeight: Int\n open var poster: String\n open var playsInline: Boolean\n\n companion object {\n val NETWORK_EMPTY: Short\n val NETWORK_IDLE: Short\n val NETWORK_LOADING: Short\n val NETWORK_NO_SOURCE: Short\n val HAVE_NOTHING: Short\n val HAVE_METADATA: Short\n val HAVE_CURRENT_DATA: Short\n val HAVE_FUTURE_DATA: Short\n val HAVE_ENOUGH_DATA: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLAudioElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAudioElement) to Kotlin\n */\npublic external abstract class HTMLAudioElement : HTMLMediaElement {\n companion object {\n val NETWORK_EMPTY: Short\n val NETWORK_IDLE: Short\n val NETWORK_LOADING: Short\n val NETWORK_NO_SOURCE: Short\n val HAVE_NOTHING: Short\n val HAVE_METADATA: Short\n val HAVE_CURRENT_DATA: Short\n val HAVE_FUTURE_DATA: Short\n val HAVE_ENOUGH_DATA: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLTrackElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTrackElement) to Kotlin\n */\npublic external abstract class HTMLTrackElement : HTMLElement {\n open var kind: String\n open var src: String\n open var srclang: String\n open var label: String\n open var default: Boolean\n open val readyState: Short\n open val track: TextTrack\n\n companion object {\n val NONE: Short\n val LOADING: Short\n val LOADED: Short\n val ERROR: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLMediaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMediaElement) to Kotlin\n */\npublic external abstract class HTMLMediaElement : HTMLElement {\n open val error: MediaError?\n open var src: String\n open var srcObject: MediaProvider?\n open val currentSrc: String\n open var crossOrigin: String?\n open val networkState: Short\n open var preload: String\n open val buffered: TimeRanges\n open val readyState: Short\n open val seeking: Boolean\n open var currentTime: Double\n open val duration:

```
Double\n    open val paused: Boolean\n    open var defaultPlaybackRate: Double\n    open var playbackRate:
Double\n    open val played: TimeRanges\n    open val seekable: TimeRanges\n    open val ended: Boolean\n    open
var autoplay: Boolean\n    open var loop: Boolean\n    open var controls: Boolean\n    open var volume: Double\n
open var muted: Boolean\n    open var defaultMuted: Boolean\n    open val audioTracks:
 AudioTrackList\n    open val videoTracks: VideoTrackList\n    open val textTracks: TextTrackList\n    open val
mediaKeys: MediaKeys?\n    open var onencrypted: ((Event) -> dynamic)?\n    open var onwaitingforkey: ((Event) -
> dynamic)?\n    fun load()\n    fun canPlayType(type: String): CanPlayTypeResult\n    fun fastSeek(time: Double)\n
  fun getStartDate(): dynamic\n    fun play(): Promise<Unit>\n    fun pause()\n    fun addTextTrack(kind:
TextTrackKind, label: String = definedExternally, language: String = definedExternally): TextTrack\n    fun
setMediaKeys(mediaKeys: MediaKeys?): Promise<Unit>\n\n    companion object {\n        val
NETWORK_EMPTY: Short\n        val NETWORK_IDLE: Short\n        val NETWORK_LOADING: Short\n
val NETWORK_NO_SOURCE: Short\n        val HAVE_NOTHING: Short\n        val HAVE_METADATA:
Short\n        val HAVE_CURRENT_DATA: Short\n        val HAVE_FUTURE_DATA: Short\n        val
HAVE_ENOUGH_DATA: Short\n        val ELEMENT_NODE: Short\n
 val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
     val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[MediaError](https://developer.mozilla.org/en/docs/Web/API/MediaError) to Kotlin\n */\npublic external abstract
class MediaError {\n    open val code: Short\n\n    companion object {\n        val MEDIA_ERR_ABORTED: Short\n
     val MEDIA_ERR_NETWORK:
 Short\n        val MEDIA_ERR_DECODE: Short\n        val MEDIA_ERR_SRC_NOT_SUPPORTED: Short\n
}\n}\n\n/**\n * Exposes the JavaScript
[AudioTrackList](https://developer.mozilla.org/en/docs/Web/API/AudioTrackList) to Kotlin\n */\npublic external
abstract class AudioTrackList : EventTarget {\n    open val length: Int\n    open var onchange: ((Event) ->
dynamic)?\n    open var onaddtrack: ((TrackEvent) -> dynamic)?\n    open var onremovetrack: ((TrackEvent) ->
dynamic)?\n    fun getTrackById(id: String): AudioTrack?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun AudioTrackList.get(index:
Int): AudioTrack? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[AudioTrack](https://developer.mozilla.org/en/docs/Web/API/AudioTrack) to Kotlin\n */\npublic external abstract
class AudioTrack : UnionAudioTrackOrTextTrackOrVideoTrack {\n    open val id: String\n    open val kind:
String\n    open val label: String\n
 open val language: String\n    open var enabled: Boolean\n    open val sourceBuffer: SourceBuffer?\n}\n\n/**\n *
Exposes the JavaScript [VideoTrackList](https://developer.mozilla.org/en/docs/Web/API/VideoTrackList) to
Kotlin\n */\npublic external abstract class VideoTrackList : EventTarget {\n    open val length: Int\n    open val
selectedIndex: Int\n    open var onchange: ((Event) -> dynamic)?\n    open var onaddtrack: ((TrackEvent) ->
dynamic)?\n    open var onremovetrack: ((TrackEvent) -> dynamic)?\n    fun getTrackById(id: String):
VideoTrack?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun VideoTrackList.get(index:
Int): VideoTrack? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[VideoTrack](https://developer.mozilla.org/en/docs/Web/API/VideoTrack) to Kotlin\n */\npublic external abstract
class VideoTrack : UnionAudioTrackOrTextTrackOrVideoTrack {\n    open val id: String\n    open val kind:
String\n
```

```
  open val label: String\n    open val language: String\n    open var selected: Boolean\n    open val sourceBuffer:
SourceBuffer?\n}\n\npublic external abstract class TextTrackList : EventTarget {\n    open val length: Int\n    open
var onchange: ((Event) -> dynamic)?\n    open var onaddtrack: ((TrackEvent) -> dynamic)?\n    open var
onremovetrack: ((TrackEvent) -> dynamic)?\n    fun getTrackById(id: String):
TextTrack?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun TextTrackList.get(index: Int):
TextTrack? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[TextTrack](https://developer.mozilla.org/en/docs/Web/API/TextTrack) to Kotlin\n */\npublic external abstract
class TextTrack : EventTarget, UnionAudioTrackOrTextTrackOrVideoTrack {\n    open val kind: TextTrackKind\n
 open val label: String\n    open val language: String\n    open val id: String\n    open val
inBandMetadataTrackDispatchType: String\n
 open var mode: TextTrackMode\n    open val cues: TextTrackCueList?\n    open val activeCues:
TextTrackCueList?\n    open var oncuechange: ((Event) -> dynamic)?\n    open val sourceBuffer: SourceBuffer?\n
fun addCue(cue: TextTrackCue)\n    fun removeCue(cue: TextTrackCue)\n}\n\npublic external abstract class
TextTrackCueList {\n    open val length: Int\n    fun getCueById(id: String):
TextTrackCue?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun TextTrackCueList.get(index:
Int): TextTrackCue? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript
[TextTrackCue](https://developer.mozilla.org/en/docs/Web/API/TextTrackCue) to Kotlin\n */\npublic external
abstract class TextTrackCue : EventTarget {\n    open val track: TextTrack?\n    open var id: String\n    open var
startTime: Double\n    open var endTime: Double\n    open var pauseOnExit: Boolean\n    open var onenter: ((Event)
-> dynamic)?\n    open var onexit: ((Event)
 -> dynamic)?\n}\n\n/**\n * Exposes the JavaScript
[TimeRanges](https://developer.mozilla.org/en/docs/Web/API/TimeRanges) to Kotlin\n */\npublic external abstract
class TimeRanges {\n    open val length: Int\n    fun start(index: Int): Double\n    fun end(index: Int):
Double\n}\n\n/**\n * Exposes the JavaScript
[TrackEvent](https://developer.mozilla.org/en/docs/Web/API/TrackEvent) to Kotlin\n */\npublic external open class
TrackEvent(type: String, eventInitDict: TrackEventInit = definedExternally) : Event {\n    open val track:
UnionAudioTrackOrTextTrackOrVideoTrack?\n\n    companion object {\n        val NONE: Short\n        val
CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n
    }\n}\n\npublic external interface TrackEventInit : EventInit {\n    var track:
UnionAudioTrackOrTextTrackOrVideoTrack? /* = null */\n        get() = definedExternally\n        set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic
 inline fun TrackEventInit(track: UnionAudioTrackOrTextTrackOrVideoTrack? = null, bubbles: Boolean? = false,
cancelable: Boolean? = false, composed: Boolean? = false): TrackEventInit {\n    val o = js(\"({})\")\n    o[\"track\"]
= track\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return
o\n}\n\n/**\n * Exposes the JavaScript
[HTMLMapElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMapElement) to Kotlin\n */\npublic
external abstract class HTMLMapElement : HTMLElement {\n    open var name: String\n    open val areas:
HTMLCollection\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE:
Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n
        val DOCUMENT_TYPE_NODE: Short\n        val DOCUMENT_FRAGMENT_NODE: Short\n        val
NOTATION_NODE: Short\n        val DOCUMENT_POSITION_DISCONNECTED: Short\n        val
DOCUMENT_POSITION_PRECEDING: Short\n        val DOCUMENT_POSITION_FOLLOWING: Short\n
```

val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLAreaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLAreaElement) to Kotlin\n */\npublic external abstract class HTMLAreaElement : HTMLElement, HTMLHyperlinkElementUtils {\n open var alt: String\n open var coords: String\n open var shape: String\n open var target: String\n open var download: String\n open var ping: String\n open var rel: String\n open val relList: DOMTokenList\n open var referrerPolicy: String\n open var noHref: Boolean\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLTableElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableElement) to Kotlin\n */\npublic external abstract class HTMLTableElement : HTMLElement {\n open var caption: HTMLTableCaptionElement?\n open var tHead: HTMLTableSectionElement?\n open var tFoot: HTMLTableSectionElement?\n open val tBodies: HTMLCollection\n open val rows: HTMLCollection\n open var align: String\n open var border: String\n open var frame: String\n open var rules: String\n open var summary: String\n open var width: String\n open var bgColor: String\n open var cellPadding: String\n open var cellSpacing: String\n fun createCaption(): HTMLTableCaptionElement\n fun deleteCaption()\n fun createTHead(): HTMLTableSectionElement\n fun deleteTHead()\n fun createTFoot(): HTMLTableSectionElement\n fun deleteTFoot()\n fun createTBody(): HTMLTableSectionElement\n fun insertRow(index: Int = definedExternally): HTMLTableRowElement\n fun deleteRow(index: Int)\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLTableCaptionElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableCaptionElement) to Kotlin\n */\npublic external abstract class HTMLTableCaptionElement : HTMLElement {\n open var align: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLTableColElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableColElement) to Kotlin\n

*/\npublic external abstract class HTMLTableColElement : HTMLElement {\n   open var span: Int\n   open var align: String\n   open var ch: String\n   open var chOff: String\n   open var vAlign: String\n   open var width: String\n\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n   val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n   val DOCUMENT_POSITION_FOLLOWING: Short\n   val DOCUMENT_POSITION_CONTAINS: Short\n   val DOCUMENT_POSITION_CONTAINED_BY: Short\n   val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n   }\n}\n\n/**\n * Exposes the JavaScript [HTMLTableSectionElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableSectionElement) to Kotlin\n */\npublic external abstract class HTMLTableSectionElement : HTMLElement {\n   open val rows: HTMLCollection\n   open var align: String\n   open var ch: String\n   open var chOff: String\n   open var vAlign: String\n   fun insertRow(index: Int = definedExternally): HTMLElement\n   fun deleteRow(index: Int)\n\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n   val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n   val DOCUMENT_POSITION_FOLLOWING: Short\n   val DOCUMENT_POSITION_CONTAINS: Short\n   val DOCUMENT_POSITION_CONTAINED_BY: Short\n   val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n   }\n}\n\n/**\n * Exposes the JavaScript [HTMLTableRowElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableRowElement) to Kotlin\n */\npublic external abstract class HTMLTableRowElement : HTMLElement {\n   open val rowIndex: Int\n   open val sectionRowIndex: Int\n   open val cells: HTMLCollection\n   open var align: String\n   open var ch: String\n   open var chOff: String\n   open var vAlign: String\n   open var bgColor: String\n   fun insertCell(index: Int = definedExternally): HTMLElement\n   fun deleteCell(index: Int)\n\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n   val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n   val DOCUMENT_POSITION_FOLLOWING: Short\n   val DOCUMENT_POSITION_CONTAINS: Short\n   val DOCUMENT_POSITION_CONTAINED_BY: Short\n   val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n   }\n}\n\n/**\n * Exposes the JavaScript [HTMLTableCellElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTableCellElement) to Kotlin\n */\npublic external abstract class HTMLTableCellElement : HTMLElement {\n   open var colSpan: Int\n   open var rowSpan: Int\n   open var headers: String\n   open val cellIndex: Int\n   open var scope: String\n   open var abbr: String\n   open var align: String\n   open var axis: String\n   open var height: String\n   open var width: String\n   open var ch: String\n   open var chOff: String\n   open var noWrap: Boolean\n   open var vAlign: String\n   open var bgColor: String\n\n   companion object {\n   val ELEMENT_NODE:

Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLFormElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFormElement) to Kotlin\n */\npublic external abstract class HTMLFormElement : HTMLElement {\n open var acceptCharset: String\n open var action: String\n open var autocomplete: String\n open var enctype: String\n open var encoding: String\n open var method: String\n open var name: String\n open var noValidate: Boolean\n open var target: String\n open val elements: HTMLFormControlsCollection\n open val length: Int\n fun submit()\n fun reset()\n fun checkValidity(): Boolean\n fun reportValidity(): Boolean\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun HTMLFormElement.get(index: Int): Element? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun HTMLFormElement.get(name: String): UnionElementOrRadioNodeList? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript [HTMLLabelElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLabelElement) to Kotlin\n */\npublic external abstract class HTMLLabelElement : HTMLElement {\n open val form: HTMLFormElement?\n open var htmlFor: String\n open val control: HTMLElement?\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLInputElement](https://developer.mozilla.org/en/docs/Web/API/HTMLInputElement) to Kotlin\n */\npublic external abstract class HTMLInputElement : HTMLElement {\n open var accept: String\n open var alt: String\n open var autocomplete: String\n open var autofocus: Boolean\n open var defaultChecked: Boolean\n

open var checked: Boolean\n open var dirName: String\n open var disabled: Boolean\n open val form: HTMLFormElement?\n open val files: FileList?\n open var formAction: String\n open var formEnctype: String\n open var formMethod: String\n open var formNoValidate: Boolean\n open var formTarget: String\n open var height: Int\n open var indeterminate: Boolean\n open var inputMode: String\n open val list: HTMLElement?\n open var max: String\n open var maxLength: Int\n open var min: String\n open var minLength: Int\n open var multiple: Boolean\n open var name: String\n open var pattern: String\n open var placeholder: String\n open var readOnly: Boolean\n open var required: Boolean\n open var size: Int\n open var src: String\n open var step: String\n open var type: String\n open var defaultValue: String\n open var value: String\n open var valueAsDate: dynamic\n open var valueAsNumber: Double\n open var width: Int\n open val willValidate: Boolean\n open val validity: ValidityState\n open val validationMessage: String\n open val labels: NodeList\n open var selectionStart: Int?\n open var selectionEnd: Int?\n open var selectionDirection: String?\n open var align: String\n open var useMap: String\n fun stepUp(n: Int = definedExternally)\n fun stepDown(n: Int = definedExternally)\n fun checkValidity(): Boolean\n fun reportValidity(): Boolean\n fun setCustomValidity(error: String)\n fun select()\n fun setRangeText(replacement: String)\n fun setRangeText(replacement: String, start: Int, end: Int, selectionMode: SelectionMode = definedExternally)\n fun setSelectionRange(start: Int, end: Int, direction: String = definedExternally)\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE:
Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLButtonElement](https://developer.mozilla.org/en/docs/Web/API/HTMLButtonElement) to Kotlin\n */\npublic external abstract class HTMLButtonElement : HTMLElement {\n open var autofocus: Boolean\n open var disabled: Boolean\n open val form: HTMLFormElement?\n open var formAction: String\n open var formEnctype: String\n open var formMethod: String\n open var formNoValidate: Boolean\n open var formTarget: String\n open var name: String\n open var type: String\n open var value: String\n open var menu: HTMLMenuElement?\n open val willValidate: Boolean\n open val validity: ValidityState\n open val validationMessage: String\n open val labels: NodeList\n fun checkValidity(): Boolean\n fun reportValidity(): Boolean\n fun setCustomValidity(error: String)\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLSelectElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSelectElement) to Kotlin\n */\npublic external abstract class HTMLSelectElement : HTMLElement, ItemArrayLike<Element> {\n open var autocomplete: String\n open var autofocus: Boolean\n open var disabled: Boolean\n open val form: HTMLFormElement?\n open var multiple: Boolean\n open var name: String\n open var required: Boolean\n open var size: Int\n open val type: String\n open val options: HTMLOptionsCollection\n override var length:

```
Int\n    open val selectedOptions: HTMLCollection\n    open var selectedIndex: Int\n    open var value: String\n
open val willValidate: Boolean\n    open val validity: ValidityState\n    open val validationMessage:
 String\n    open val labels: NodeList\n    fun namedItem(name: String): HTMLOptionElement?\n    fun
add(element: UnionHTMLOptGroupElementOrHTMLOptionElement, before: dynamic = definedExternally)\n
fun remove(index: Int)\n    fun checkValidity(): Boolean\n    fun reportValidity(): Boolean\n    fun
setCustomValidity(error: String)\n    override fun item(index: Int): Element?\n\n    companion object {\n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING:
 Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n        val
DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n
}\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLSelectElement.get(index: Int): Element? =
asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun
HTMLSelectElement.set(index: Int, option: HTMLOptionElement?) { asDynamic()[index] = option }\n\n/**\n *
Exposes the JavaScript
[HTMLDataListElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDataListElement) to Kotlin\n
*/\npublic external abstract class HTMLDataListElement : HTMLElement {\n    open val options:
HTMLCollection\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE:
Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE:
 Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
    val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[HTMLOptGroupElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOptGroupElement) to Kotlin\n
*/\npublic external abstract class HTMLOptGroupElement : HTMLElement,
UnionHTMLOptGroupElementOrHTMLOptionElement {\n    open var disabled: Boolean\n    open var label:
String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n
    val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
```

[HTMLOptionElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOptionElement) to Kotlin\n */\npublic external abstract class HTMLOptionElement : HTMLElement, UnionHTMLOptGroupElementOrHTMLOptionElement {\n open var disabled: Boolean\n open val form: HTMLFormElement?\n open var label: String\n open var defaultSelected: Boolean\n open var selected: Boolean\n open var value: String\n open var text: String\n open val index: Int\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLTextAreaElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTextAreaElement) to Kotlin\n */\npublic external abstract class HTMLTextAreaElement : HTMLElement {\n open var autocomplete: String\n open var autofocus: Boolean\n open var cols: Int\n open var dirName: String\n open var disabled: Boolean\n open val form: HTMLFormElement?\n open var inputMode: String\n open var maxLength: Int\n open var minLength: Int\n open var name: String\n open var placeholder: String\n open var readOnly: Boolean\n open var required: Boolean\n open var rows: Int\n open var wrap: String\n open val type: String\n open var defaultValue: String\n open var value: String\n open val textLength: Int\n open val willValidate: Boolean\n open val validity: ValidityState\n open val validationMessage: String\n open val labels: NodeList\n open var selectionStart: Int?\n open var selectionEnd: Int?\n open var selectionDirection: String?\n fun checkValidity(): Boolean\n fun reportValidity(): Boolean\n fun setCustomValidity(error: String)\n fun select()\n fun setRangeText(replacement: String)\n fun setRangeText(replacement: String, start: Int, end: Int, selectionMode: SelectionMode = definedExternally)\n fun setSelectionRange(start: Int, end: Int, direction: String = definedExternally)\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLKeygenElement](https://developer.mozilla.org/en/docs/Web/API/HTMLKeygenElement) to Kotlin\n */\npublic external abstract class HTMLKeygenElement : HTMLElement {\n open var autofocus: Boolean\n open var challenge: String\n open var disabled: Boolean\n open val form: HTMLFormElement?\n open var keytype: String\n open var name: String\n open val type: String\n open val willValidate: Boolean\n open val validity: ValidityState\n open val validationMessage: String\n open val labels: NodeList\n fun checkValidity(): Boolean\n fun reportValidity(): Boolean\n fun setCustomValidity(error: String)\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val

DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLOutputElement](https://developer.mozilla.org/en/docs/Web/API/HTMLOutputElement) to Kotlin\n */\npublic external abstract class HTMLOutputElement : HTMLElement {\n open val htmlFor: DOMTokenList\n open val form: HTMLFormElement?\n open var name: String\n open val type: String\n open var defaultValue: String\n open var value: String\n open val willValidate: Boolean\n open val validity: ValidityState\n open val validationMessage: String\n open val labels: NodeList\n fun checkValidity(): Boolean\n fun reportValidity(): Boolean\n fun setCustomValidity(error: String)\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLProgressElement](https://developer.mozilla.org/en/docs/Web/API/HTMLProgressElement) to Kotlin\n */\npublic external abstract class HTMLProgressElement : HTMLElement {\n open var value: Double\n open var max: Double\n open val position: Double\n open val labels: NodeList\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLMeterElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMeterElement) to Kotlin\n */\npublic external abstract class HTMLMeterElement : HTMLElement {\n open var value: Double\n open var min: Double\n open var max: Double\n open var low: Double\n open var high: Double\n open var optimum: Double\n open val labels: NodeList\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLFieldSetElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFieldSetElement) to Kotlin\n */\npublic external abstract class HTMLFieldSetElement : HTMLElement {\n open var disabled:

```
Boolean\n    open val form: HTMLFormElement?\n    open var name: String\n    open val type: String\n    open val elements: HTMLCollection\n    open val willValidate: Boolean\n    open val validity: ValidityState\n    open val validationMessage: String\n    fun checkValidity(): Boolean\n    fun reportValidity(): Boolean\n    fun setCustomValidity(error: String)\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript [HTMLLegendElement](https://developer.mozilla.org/en/docs/Web/API/HTMLLegendElement) to Kotlin\n */\npublic external abstract class HTMLLegendElement : HTMLElement {\n    open val form: HTMLFormElement?\n    open var align: String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript [ValidityState](https://developer.mozilla.org/en/docs/Web/API/ValidityState) to Kotlin\n */\npublic external abstract class ValidityState {\n    open val valueMissing: Boolean\n    open val typeMismatch: Boolean\n    open val patternMismatch: Boolean\n    open val tooLong: Boolean\n    open val tooShort: Boolean\n    open val rangeUnderflow: Boolean\n    open val rangeOverflow: Boolean\n    open val stepMismatch: Boolean\n    open val badInput: Boolean\n    open val customError: Boolean\n    open val valid: Boolean\n}\n\n/**\n * Exposes the JavaScript [HTMLDetailsElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDetailsElement) to Kotlin\n */\npublic external abstract class HTMLDetailsElement : HTMLElement {\n    open var open: Boolean\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external abstract class HTMLMenuElement : HTMLElement {\n    open var type: String\n    open var label: String\n    open var compact: Boolean\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
```

DOCUMENT_POSITION_DISCONNECTED:
Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val
DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class
HTMLMenuItemElement : HTMLElement {\n open var type: String\n open var label: String\n open var icon:
String\n open var disabled: Boolean\n open var checked: Boolean\n open var radiogroup: String\n open var
default: Boolean\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val
PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n
val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val
DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n
val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY:
Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external
open class RelatedEvent(type: String, eventInitDict: RelatedEventInit = definedExternally) : Event {\n open val
relatedTarget: EventTarget?\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE:
Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface
RelatedEventInit : EventInit {\n var relatedTarget: EventTarget? /* = null */\n get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun RelatedEventInit(relatedTarget:
EventTarget?
= null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): RelatedEventInit {\n
val o = js(\"({})\")\n o[\"relatedTarget\"] = relatedTarget\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] =
cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[HTMLDialogElement](https://developer.mozilla.org/en/docs/Web/API/HTMLDialogElement) to Kotlin\n
*/\npublic external abstract class HTMLDialogElement : HTMLElement {\n open var open: Boolean\n open var
returnValue: String\n fun show(anchor: UnionElementOrMouseEvent = definedExternally)\n fun
showModal(anchor: UnionElementOrMouseEvent = definedExternally)\n fun close(returnValue: String =
definedExternally)\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE:
Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val
ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n
val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val
DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val
DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val
DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n
val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n
val DOCUMENT_POSITION_CONTAINED_BY: Short\n val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript
[HTMLScriptElement](https://developer.mozilla.org/en/docs/Web/API/HTMLScriptElement) to Kotlin\n */\npublic
external abstract class HTMLScriptElement : HTMLElement, HTMLOrSVGScriptElement {\n open var src:
String\n open var type: String\n open var charset: String\n open var async: Boolean\n open var defer:
Boolean\n open var crossOrigin: String?\n open var text: String\n open var nonce:
String\n open var event: String\n open var htmlFor: String\n\n companion object {\n val
ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val
CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE:

Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLTemplateElement](https://developer.mozilla.org/en/docs/Web/API/HTMLTemplateElement) to Kotlin\n */\npublic external abstract class HTMLTemplateElement : HTMLElement {\n open val content: DocumentFragment\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [HTMLSlotElement](https://developer.mozilla.org/en/docs/Web/API/HTMLSlotElement) to Kotlin\n */\npublic external abstract class HTMLSlotElement : HTMLElement {\n open var name: String\n fun assignedNodes(options: AssignedNodesOptions = definedExternally): Array<Node>\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external interface AssignedNodesOptions {\n var flatten: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun AssignedNodesOptions(flatten: Boolean? = false): AssignedNodesOptions {\n val o = js(\"({})\")\n o[\"flatten\"] = flatten\n return o\n}\n\n/**\n * Exposes the JavaScript [HTMLCanvasElement](https://developer.mozilla.org/en/docs/Web/API/HTMLCanvasElement) to Kotlin\n */\npublic external abstract class HTMLCanvasElement : HTMLElement, CanvasImageSource, TexImageSource {\n open var width: Int\n open var height: Int\n fun getContext(contextId: String, vararg arguments: Any?): RenderingContext?\n fun toDataURL(type: String = definedExternally, quality: Any? = definedExternally): String\n fun toBlob(_callback: (Blob?) -> Unit, type: String = definedExternally, quality: Any? = definedExternally)\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n

```
    val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external interface
CanvasRenderingContext2DSettings {\n    var alpha: Boolean? /* = true */\n        get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic
 inline fun CanvasRenderingContext2DSettings(alpha: Boolean? = true): CanvasRenderingContext2DSettings {\n
val o = js(\"({})\")\n    o[\"alpha\"] = alpha\n    return o\n}\n\n/**\n * Exposes the JavaScript
[CanvasRenderingContext2D](https://developer.mozilla.org/en/docs/Web/API/CanvasRenderingContext2D) to
Kotlin\n */\npublic external abstract class CanvasRenderingContext2D : CanvasState, CanvasTransform,
CanvasCompositing, CanvasImageSmoothing, CanvasFillStrokeStyles, CanvasShadowStyles, CanvasFilters,
CanvasRect, CanvasDrawPath, CanvasUserInterface, CanvasText, CanvasDrawImage, CanvasHitRegion,
CanvasImageData, CanvasPathDrawingStyles, CanvasTextDrawingStyles, CanvasPath, RenderingContext {\n
open val canvas: HTMLCanvasElement\n}\n\npublic external interface CanvasState {\n    fun save()\n    fun
restore()\n}\n\npublic external interface CanvasTransform {\n    fun scale(x: Double, y: Double)\n    fun
rotate(angle: Double)\n    fun translate(x:
 Double, y: Double)\n    fun transform(a: Double, b: Double, c: Double, d: Double, e: Double, f: Double)\n    fun
getTransform(): DOMMatrix\n    fun setTransform(a: Double, b: Double, c: Double, d: Double, e: Double, f:
Double)\n    fun setTransform(transform: dynamic = definedExternally)\n    fun resetTransform()\n}\n\npublic
external interface CanvasCompositing {\n    var globalAlpha: Double\n    var globalCompositeOperation:
String\n}\n\npublic external interface CanvasImageSmoothing {\n    var imageSmoothingEnabled: Boolean\n    var
imageSmoothingQuality: ImageSmoothingQuality\n}\n\npublic external interface CanvasFillStrokeStyles {\n    var
strokeStyle: dynamic\n        get() = definedExternally\n        set(value) = definedExternally\n    var fillStyle:
dynamic\n        get() = definedExternally\n        set(value) = definedExternally\n    fun createLinearGradient(x0:
Double, y0: Double, x1: Double, y1: Double): CanvasGradient\n    fun createRadialGradient(x0: Double, y0:
Double,
 r0: Double, x1: Double, y1: Double, r1: Double): CanvasGradient\n    fun createPattern(image:
CanvasImageSource, repetition: String): CanvasPattern?\n}\n\npublic external interface CanvasShadowStyles {\n
var shadowOffsetX: Double\n    var shadowOffsetY: Double\n    var shadowBlur: Double\n    var shadowColor:
String\n}\n\npublic external interface CanvasFilters {\n    var filter: String\n}\n\npublic external interface
CanvasRect {\n    fun clearRect(x: Double, y: Double, w: Double, h: Double)\n    fun fillRect(x: Double, y: Double,
w: Double, h: Double)\n    fun strokeRect(x: Double, y: Double, w: Double, h: Double)\n}\n\npublic external
interface CanvasDrawPath {\n    fun beginPath()\n    fun fill(fillRule: CanvasFillRule = definedExternally)\n    fun
fill(path: Path2D, fillRule: CanvasFillRule = definedExternally)\n    fun stroke()\n    fun stroke(path: Path2D)\n
fun clip(fillRule: CanvasFillRule = definedExternally)\n    fun clip(path: Path2D, fillRule: CanvasFillRule =
definedExternally)\n
    fun resetClip()\n    fun isPointInPath(x: Double, y: Double, fillRule: CanvasFillRule = definedExternally):
Boolean\n    fun isPointInPath(path: Path2D, x: Double, y: Double, fillRule: CanvasFillRule = definedExternally):
Boolean\n    fun isPointInStroke(x: Double, y: Double): Boolean\n    fun isPointInStroke(path: Path2D, x: Double,
y: Double): Boolean\n}\n\npublic external interface CanvasUserInterface {\n    fun drawFocusIfNeeded(element:
Element)\n    fun drawFocusIfNeeded(path: Path2D, element: Element)\n    fun scrollPathIntoView()\n    fun
scrollPathIntoView(path: Path2D)\n}\n\npublic external interface CanvasText {\n    fun fillText(text: String, x:
Double, y: Double, maxWidth: Double = definedExternally)\n    fun strokeText(text: String, x: Double, y: Double,
maxWidth: Double = definedExternally)\n    fun measureText(text: String): TextMetrics\n}\n\npublic external
interface CanvasDrawImage {\n    fun drawImage(image: CanvasImageSource, dx: Double, dy: Double)\n
    fun drawImage(image: CanvasImageSource, dx: Double, dy: Double, dw: Double, dh: Double)\n    fun
drawImage(image: CanvasImageSource, sx: Double, sy: Double, sw: Double, sh: Double, dx: Double, dy: Double,
dw: Double, dh: Double)\n}\n\npublic external interface CanvasHitRegion {\n    fun addHitRegion(options:
```

HitRegionOptions = definedExternally)\n   fun removeHitRegion(id: String)\n   fun clearHitRegions()\n}\n\npublic external interface CanvasImageData {\n   fun createImageData(sw: Double, sh: Double): ImageData\n   fun createImageData(imagedata: ImageData): ImageData\n   fun getImageData(sx: Double, sy: Double, sw: Double, sh: Double): ImageData\n   fun putImageData(imagedata: ImageData, dx: Double, dy: Double)\n   fun putImageData(imagedata: ImageData, dx: Double, dy: Double, dirtyX: Double, dirtyY: Double, dirtyWidth: Double, dirtyHeight: Double)\n}\n\npublic external interface CanvasPathDrawingStyles {\n   var lineWidth: Double\n   var lineCap: CanvasLineCap\n
 var lineJoin: CanvasLineJoin\n   var miterLimit: Double\n   var lineDashOffset: Double\n   fun setLineDash(segments: Array<Double>)\n   fun getLineDash(): Array<Double>\n}\n\npublic external interface CanvasTextDrawingStyles {\n   var font: String\n   var textAlign: CanvasTextAlign\n   var textBaseline: CanvasTextBaseline\n   var direction: CanvasDirection\n}\n\npublic external interface CanvasPath {\n   fun closePath()\n   fun moveTo(x: Double, y: Double)\n   fun lineTo(x: Double, y: Double)\n   fun quadraticCurveTo(cpx: Double, cpy: Double, x: Double, y: Double)\n   fun bezierCurveTo(cp1x: Double, cp1y: Double, cp2x: Double, cp2y: Double, x: Double, y: Double)\n   fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radius: Double)\n   fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radiusX: Double, radiusY: Double, rotation: Double)\n   fun rect(x: Double, y: Double, w: Double, h: Double)\n   fun arc(x: Double, y: Double, radius: Double, startAngle: Double,
 endAngle: Double, anticlockwise: Boolean = definedExternally)\n   fun ellipse(x: Double, y: Double, radiusX: Double, radiusY: Double, rotation: Double, startAngle: Double, endAngle: Double, anticlockwise: Boolean = definedExternally)\n}\n\n/**\n * Exposes the JavaScript [CanvasGradient](https://developer.mozilla.org/en/docs/Web/API/CanvasGradient) to Kotlin\n */\npublic external abstract class CanvasGradient {\n   fun addColorStop(offset: Double, color: String)\n}\n\n/**\n * Exposes the JavaScript [CanvasPattern](https://developer.mozilla.org/en/docs/Web/API/CanvasPattern) to Kotlin\n */\npublic external abstract class CanvasPattern {\n   fun setTransform(transform: dynamic = definedExternally)\n}\n\n/**\n * Exposes the JavaScript [TextMetrics](https://developer.mozilla.org/en/docs/Web/API/TextMetrics) to Kotlin\n */\npublic external abstract class TextMetrics {\n   open val width: Double\n   open val actualBoundingBoxLeft: Double\n   open val actualBoundingBoxRight: Double\n
    open val fontBoundingBoxAscent: Double\n   open val fontBoundingBoxDescent: Double\n   open val actualBoundingBoxAscent: Double\n   open val actualBoundingBoxDescent: Double\n   open val emHeightAscent: Double\n   open val emHeightDescent: Double\n   open val hangingBaseline: Double\n   open val alphabeticBaseline: Double\n   open val ideographicBaseline: Double\n}\n\npublic external interface HitRegionOptions {\n   var path: Path2D? /* = null */\n   get() = definedExternally\n   set(value) = definedExternally\n   var fillRule: CanvasFillRule? /* = CanvasFillRule.NONZERO */\n   get() = definedExternally\n   set(value) = definedExternally\n   var id: String? /* = \"\" */\n   get() = definedExternally\n   set(value) = definedExternally\n   var parentID: String? /* = null */\n   get() = definedExternally\n   set(value) = definedExternally\n   var cursor: String? /* = \"inherit\" */\n   get() = definedExternally\n   set(value)
 = definedExternally\n   var control: Element? /* = null */\n   get() = definedExternally\n   set(value) = definedExternally\n   var label: String? /* = null */\n   get() = definedExternally\n   set(value) = definedExternally\n   var role: String? /* = null */\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun HitRegionOptions(path: Path2D? = null, fillRule: CanvasFillRule? = CanvasFillRule.NONZERO, id: String? = \"\", parentID: String? = null, cursor: String? = \"inherit\", control: Element? = null, label: String? = null, role: String? = null): HitRegionOptions {\n   val o = js(\"({})\")\n   o[\"path\"] = path\n   o[\"fillRule\"] = fillRule\n   o[\"id\"] = id\n   o[\"parentID\"] = parentID\n
 o[\"cursor\"] = cursor\n   o[\"control\"] = control\n   o[\"label\"] = label\n   o[\"role\"] = role\n   return o\n}\n\n/**\n

```
 * Exposes the JavaScript [ImageData](https://developer.mozilla.org/en/docs/Web/API/ImageData) to Kotlin\n */\npublic external open class ImageData : ImageBitmapSource, TexImageSource {\n    constructor(sw: Int, sh: Int)\n    constructor(data: Uint8ClampedArray, sw: Int, sh: Int = definedExternally)\n    open val width: Int\n    open val height: Int\n    open val data: Uint8ClampedArray\n}\n\n/**\n * Exposes the JavaScript [Path2D](https://developer.mozilla.org/en/docs/Web/API/Path2D) to Kotlin\n */\npublic external open class Path2D() : CanvasPath {\n    constructor(path: Path2D)\n    constructor(paths: Array<Path2D>, fillRule: CanvasFillRule = definedExternally)\n    constructor(d: String)\n    fun addPath(path: Path2D, transform: dynamic = definedExternally)\n    override fun closePath()\n    override fun moveTo(x: Double, y: Double)\n    override fun lineTo(x: Double, y: Double)\n    override fun quadraticCurveTo(cpx: Double, cpy: Double, x: Double, y: Double)\n    override fun bezierCurveTo(cp1x: Double, cp1y: Double, cp2x: Double, cp2y: Double, x: Double, y: Double)\n    override fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radius: Double)\n    override fun arcTo(x1: Double, y1: Double, x2: Double, y2: Double, radiusX: Double, radiusY: Double, rotation: Double)\n    override fun rect(x: Double, y: Double, w: Double, h: Double)\n    override fun arc(x: Double, y: Double, radius: Double, startAngle: Double, endAngle: Double, anticlockwise: Boolean /* = definedExternally */)\n    override fun ellipse(x: Double, y: Double, radiusX: Double, radiusY: Double, rotation: Double, startAngle: Double, endAngle: Double, anticlockwise: Boolean /* = definedExternally */)\n}\n\n/**\n * Exposes the JavaScript [ImageBitmapRenderingContext](https://developer.mozilla.org/en/docs/Web/API/ImageBitmapRenderingContext) to Kotlin\n */\npublic external abstract class ImageBitmapRenderingContext {\n    open val canvas: HTMLCanvasElement\n    fun transferFromImageBitmap(bitmap: ImageBitmap?)\n}\n\npublic external interface ImageBitmapRenderingContextSettings {\n    var alpha: Boolean? /* = true */\n        get() = definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ImageBitmapRenderingContextSettings(alpha: Boolean? = true): ImageBitmapRenderingContextSettings {\n    val o = js(\"({})\")\n    o[\"alpha\"] = alpha\n    return o\n}\n\n/**\n * Exposes the JavaScript [CustomElementRegistry](https://developer.mozilla.org/en/docs/Web/API/CustomElementRegistry) to Kotlin\n */\npublic external abstract class CustomElementRegistry {\n    fun define(name: String, constructor: () -> dynamic, options: ElementDefinitionOptions = definedExternally)\n    fun get(name: String): Any?\n    fun whenDefined(name: String): Promise<Unit>\n}\n\npublic external interface ElementDefinitionOptions {\n    var extends: String?\n        get() = definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ElementDefinitionOptions(extends: String? = undefined): ElementDefinitionOptions {\n    val o = js(\"({})\")\n    o[\"extends\"] = extends\n    return o\n}\n\npublic external interface ElementContentEditable {\n    var contentEditable: String\n    val isContentEditable: Boolean\n}\n\n/**\n * Exposes the JavaScript [DataTransfer](https://developer.mozilla.org/en/docs/Web/API/DataTransfer) to Kotlin\n */\npublic external abstract class DataTransfer {\n    open var dropEffect: String\n    open var effectAllowed: String\n    open val items: DataTransferItemList\n    open val types: Array<out String>\n    open val files: FileList\n    fun setDragImage(image: Element, x: Int, y: Int)\n    fun getData(format: String): String\n    fun setData(format: String, data: String)\n    fun clearData(format: String = definedExternally)\n}\n\n/**\n * Exposes the JavaScript [DataTransferItemList](https://developer.mozilla.org/en/docs/Web/API/DataTransferItemList) to Kotlin\n */\npublic external abstract class DataTransferItemList {\n    open val length: Int\n    fun add(data: String, type: String): DataTransferItem?\n    fun add(data: File): DataTransferItem?\n    fun remove(index: Int)\n    fun clear()\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun DataTransferItemList.get(index: Int): DataTransferItem? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript [DataTransferItem](https://developer.mozilla.org/en/docs/Web/API/DataTransferItem) to Kotlin\n */\npublic
```

external abstract class DataTransferItem {\n   open val kind: String\n   open val type: String\n   fun getAsString(_callback: ((String) -> Unit)?)\n   fun getAsFile(): File?\n}\n\n/**\n * Exposes the JavaScript [DragEvent](https://developer.mozilla.org/en/docs/Web/API/DragEvent) to Kotlin\n */\npublic external open class DragEvent(type: String, eventInitDict: DragEventInit = definedExternally) : MouseEvent {\n   open val dataTransfer: DataTransfer?\n\n   companion object {\n   val NONE: Short\n   val CAPTURING_PHASE: Short\n   val AT_TARGET: Short\n   val BUBBLING_PHASE: Short\n   }\n}\n\npublic external interface DragEventInit : MouseEventInit {\n   var dataTransfer: DataTransfer? /* = null */\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun DragEventInit(dataTransfer: DataTransfer? = null, screenX: Int? = 0, screenY: Int? = 0, clientX: Int? = 0, clientY: Int? = 0, button: Short? = 0, buttons: Short? = 0, relatedTarget: EventTarget? = null, region: String? = null, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean?
 = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): DragEventInit {\n   val o = js(\"({})\")\n   o[\"dataTransfer\"] = dataTransfer\n   o[\"screenX\"] = screenX\n   o[\"screenY\"] = screenY\n   o[\"clientX\"] = clientX\n   o[\"clientY\"] = clientY\n   o[\"button\"] = button\n   o[\"buttons\"] = buttons\n   o[\"relatedTarget\"] = relatedTarget\n   o[\"region\"] = region\n   o[\"ctrlKey\"] = ctrlKey\n   o[\"shiftKey\"] = shiftKey\n   o[\"altKey\"] = altKey\n   o[\"metaKey\"] = metaKey\n   o[\"modifierAltGraph\"] = modifierAltGraph\n   o[\"modifierCapsLock\"] = modifierCapsLock\n   o[\"modifierFn\"] = modifierFn\n   o[\"modifierFnLock\"] = modifierFnLock\n   o[\"modifierHyper\"] = modifierHyper\n   o[\"modifierNumLock\"] = modifierNumLock\n   o[\"modifierScrollLock\"] = modifierScrollLock\n   o[\"modifierSuper\"] = modifierSuper\n   o[\"modifierSymbol\"] = modifierSymbol\n   o[\"modifierSymbolLock\"] = modifierSymbolLock\n   o[\"view\"] = view\n   o[\"detail\"] = detail\n   o[\"bubbles\"] = bubbles\n   o[\"cancelable\"] = cancelable\n   o[\"composed\"] = composed\n   return o\n}\n\n/**\n * Exposes the JavaScript [Window](https://developer.mozilla.org/en/docs/Web/API/Window) to Kotlin\n */\npublic external abstract class Window : EventTarget, GlobalEventHandlers, WindowEventHandlers, WindowOrWorkerGlobalScope, WindowSessionStorage, WindowLocalStorage, GlobalPerformance, UnionMessagePortOrWindowProxy {\n   open val window: Window\n   open val self: Window\n   open val document: Document\n   open var name: String\n   open val location: Location\n   open val history: History\n   open
 val customElements: CustomElementRegistry\n   open val locationbar: BarProp\n   open val menubar: BarProp\n   open val personalbar: BarProp\n   open val scrollbars: BarProp\n   open val statusbar: BarProp\n   open val toolbar: BarProp\n   open var status: String\n   open val closed: Boolean\n   open val frames: Window\n   open val length: Int\n   open val top: Window\n   open var opener: Any?\n   open val parent: Window\n   open val frameElement: Element?\n   open val navigator: Navigator\n   open val applicationCache: ApplicationCache\n   open val external: External\n   open val screen: Screen\n   open val innerWidth: Int\n   open val innerHeight: Int\n   open val scrollX: Double\n   open val pageXOffset: Double\n   open val scrollY: Double\n   open val pageYOffset: Double\n   open val screenX: Int\n   open val screenY: Int\n   open val outerWidth: Int\n   open val outerHeight: Int\n   open val devicePixelRatio: Double\n   fun close()\n   fun stop()\n   fun focus()\n   fun blur()\n   fun open(url: String = definedExternally, target: String = definedExternally, features: String = definedExternally): Window?\n   fun alert()\n   fun alert(message: String)\n   fun confirm(message: String = definedExternally): Boolean\n   fun prompt(message: String = definedExternally, default: String = definedExternally): String?\n   fun print()\n   fun requestAnimationFrame(callback: (Double) -> Unit): Int\n   fun cancelAnimationFrame(handle: Int)\n   fun postMessage(message: Any?, targetOrigin: String, transfer: Array<dynamic> = definedExternally)\n   fun captureEvents()\n   fun releaseEvents()\n   fun matchMedia(query: String): MediaQueryList\n   fun moveTo(x: Int, y: Int)\n   fun moveBy(x: Int, y: Int)\n   fun

```
resizeTo(x: Int, y: Int)\n    fun resizeBy(x: Int, y: Int)\n    fun scroll(options: ScrollToOptions = definedExternally)\n    fun scroll(x: Double, y: Double)\n    fun scrollTo(options: ScrollToOptions = definedExternally)\n    fun scrollTo(x: Double, y: Double)\n    fun scrollBy(options: ScrollToOptions = definedExternally)\n    fun scrollBy(x: Double, y: Double)\n    fun getComputedStyle(elt: Element, pseudoElt: String? = definedExternally): CSSStyleDeclaration\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Window.get(name: String): dynamic = asDynamic()[name]\n\npublic external abstract class BarProp {\n    open val visible: Boolean\n}\n\n/**\n * Exposes the JavaScript [History](https://developer.mozilla.org/en/docs/Web/API/History) to Kotlin\n */\npublic external abstract class History {\n    open val length: Int\n    open var scrollRestoration: ScrollRestoration\n    open val state: Any?\n    fun go(delta: Int = definedExternally)\n    fun back()\n    fun forward()\n    fun pushState(data: Any?, title: String, url: String? = definedExternally)\n    fun replaceState(data: Any?, title: String, url: String? = definedExternally)\n}\n\n/**\n * Exposes the JavaScript [Location](https://developer.mozilla.org/en/docs/Web/API/Location) to Kotlin\n */\npublic external abstract class Location {\n    open var href: String\n    open val origin: String\n    open var protocol: String\n    open var host: String\n    open var hostname: String\n    open var port: String\n    open var pathname: String\n    open var search: String\n    open var hash: String\n    open val ancestorOrigins: Array<out String>\n    fun assign(url: String)\n    fun replace(url: String)\n    fun reload()\n}\n\n/**\n * Exposes the JavaScript [PopStateEvent](https://developer.mozilla.org/en/docs/Web/API/PopStateEvent) to Kotlin\n */\npublic external open class PopStateEvent(type: String, eventInitDict: PopStateEventInit = definedExternally) : Event {\n    open val state: Any?\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface PopStateEventInit : EventInit {\n    var state: Any? /* = null */\n        get() = definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun PopStateEventInit(state: Any? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): PopStateEventInit {\n    val o = js(\"({})\")\n    o[\"state\"] = state\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\n/**\n * Exposes the JavaScript [HashChangeEvent](https://developer.mozilla.org/en/docs/Web/API/HashChangeEvent) to Kotlin\n */\npublic external open class HashChangeEvent(type: String, eventInitDict: HashChangeEventInit = definedExternally) : Event {\n    open val oldURL: String\n    open val newURL: String\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface HashChangeEventInit : EventInit {\n    var oldURL: String? /* = \"\" */\n        get() = definedExternally\n        set(value) = definedExternally\n    var newURL: String? /* = \"\" */\n        get() = definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun HashChangeEventInit(oldURL: String? = \"\", newURL: String? = \"\", bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): HashChangeEventInit {\n    val o = js(\"({})\")\n    o[\"oldURL\"] = oldURL\n    o[\"newURL\"] = newURL\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\n/**\n * Exposes the JavaScript [PageTransitionEvent](https://developer.mozilla.org/en/docs/Web/API/PageTransitionEvent) to Kotlin\n */\npublic external open class PageTransitionEvent(type: String, eventInitDict: PageTransitionEventInit = definedExternally) : Event {\n    open val persisted: Boolean\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface PageTransitionEventInit : EventInit {\n    var persisted: Boolean? /* = false */\n        get() = definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun PageTransitionEventInit(persisted:
```

```
Boolean? = false, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): PageTransitionEventInit {\n    val o = js(\"({})\")\n    o[\"persisted\"] = persisted\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\n/**\n * Exposes the JavaScript [BeforeUnloadEvent](https://developer.mozilla.org/en/docs/Web/API/BeforeUnloadEvent) to Kotlin\n */\npublic external open class BeforeUnloadEvent : Event {\n    var returnValue: String\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external abstract class ApplicationCache : EventTarget {\n    open val status: Short\n    open var onchecking: ((Event) -> dynamic)?\n    open var onerror: ((Event) -> dynamic)?\n    open var onnoupdate: ((Event) -> dynamic)?\n    open var ondownloading: ((Event) -> dynamic)?\n    open var onprogress: ((ProgressEvent) -> dynamic)?\n    open var onupdateready: ((Event) -> dynamic)?\n    open var oncached: ((Event) -> dynamic)?\n    open var onobsolete: ((Event) -> dynamic)?\n    fun update()\n    fun abort()\n    fun swapCache()\n\n    companion object {\n        val UNCACHED: Short\n        val IDLE: Short\n        val CHECKING: Short\n        val DOWNLOADING: Short\n        val UPDATEREADY: Short\n        val OBSOLETE: Short\n    }\n}\n\n/**\n * Exposes the JavaScript [NavigatorOnLine](https://developer.mozilla.org/en/docs/Web/API/NavigatorOnLine) to Kotlin\n */\npublic external interface NavigatorOnLine {\n    val onLine: Boolean\n}\n\n/**\n * Exposes the JavaScript [ErrorEvent](https://developer.mozilla.org/en/docs/Web/API/ErrorEvent) to Kotlin\n */\npublic external open class ErrorEvent(type: String, eventInitDict: ErrorEventInit = definedExternally) : Event {\n    open val message: String\n    open val filename: String\n    open val lineno: Int\n    open val colno: Int\n    open val error: Any?\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface ErrorEventInit : EventInit {\n    var message: String? /* = \"\" */\n        get() = definedExternally\n        set(value) = definedExternally\n    var filename: String? /* = \"\"\n */\n        get() = definedExternally\n        set(value) = definedExternally\n    var lineno: Int? /* = 0 */\n        get() = definedExternally\n        set(value) = definedExternally\n    var colno: Int? /* = 0 */\n        get() = definedExternally\n        set(value) = definedExternally\n    var error: Any? /* = null */\n        get() = definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ErrorEventInit(message: String? = \"\", filename: String? = \"\", lineno: Int? = 0, colno: Int? = 0, error: Any? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ErrorEventInit {\n    val o = js(\"({})\")\n    o[\"message\"] = message\n    o[\"filename\"] = filename\n    o[\"lineno\"] = lineno\n    o[\"colno\"] = colno\n    o[\"error\"] = error\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\n/**\n * Exposes the JavaScript [PromiseRejectionEvent](https://developer.mozilla.org/en/docs/Web/API/PromiseRejectionEvent) to Kotlin\n */\npublic external open class PromiseRejectionEvent(type: String, eventInitDict: PromiseRejectionEventInit) : Event {\n    open val promise: Promise<Any?>\n    open val reason: Any?\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface PromiseRejectionEventInit : EventInit {\n    var promise: Promise<Any?>?\n    var reason: Any?\n        get() = definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun PromiseRejectionEventInit(promise: Promise<Any?>?, reason: Any? = undefined, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): PromiseRejectionEventInit {\n    val o = js(\"({})\")\n    o[\"promise\"] = promise\n    o[\"reason\"] = reason\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\n/**\n * Exposes the JavaScript [GlobalEventHandlers](https://developer.mozilla.org/en/docs/Web/API/GlobalEventHandlers) to Kotlin\n */\npublic external interface GlobalEventHandlers {\n    var onabort: ((Event) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n    var onblur: ((FocusEvent) -> dynamic)?\n        get() =
```

definedExternally\n set(value) = definedExternally\n var oncancel: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var oncanplay: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var oncanplaythrough: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onchange: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onclick: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onclose: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var oncontextmenu: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var oncuechange: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var ondblclick: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var ondrag: ((DragEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var ondragend: ((DragEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var ondragenter: ((DragEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var ondragexit: ((DragEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var ondragleave: ((DragEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var ondragover: ((DragEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var ondragstart: ((DragEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var ondrop: ((DragEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var ondurationchange: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onemptied: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onended: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onfocus: ((FocusEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var oninput: ((InputEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var oninvalid: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onkeydown: ((KeyboardEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onkeypress: ((KeyboardEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onkeyup: ((KeyboardEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onload: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onloadeddata: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onloadedmetadata: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onloadend: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onloadstart: ((ProgressEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onmousedown: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onmouseenter: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onmouseleave: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onmousemove: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onmouseout: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onmouseover: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onmouseup: ((MouseEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onwheel: ((WheelEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onpause: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onplay: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onplaying: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onprogress: ((ProgressEvent) -> dynamic)?\n get() = definedExternally\n

```
    set(value) = definedExternally\n    var onratechange: ((Event) -> dynamic)?\n
      get() = definedExternally\n        set(value) = definedExternally\n    var onreset: ((Event) -> dynamic)?\n
get() = definedExternally\n        set(value) = definedExternally\n    var onresize: ((Event) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var onscroll: ((Event) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var onseeked: ((Event) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var onseeking: ((Event) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var onselect: ((Event) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var onshow: ((Event) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var onstalled: ((Event) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n
   var onsubmit: ((Event) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n    var
onsuspend: ((Event) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n    var
ontimeupdate: ((Event) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n    var
ontoggle: ((Event) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n    var
onvolumechange: ((Event) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n
var onwaiting: ((Event) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n    var
ongotpointercapture: ((PointerEvent) -> dynamic)?\n        get() = definedExternally\n        set(value) =
definedExternally\n    var onlostpointercapture: ((PointerEvent) -> dynamic)?\n        get() = definedExternally\n
set(value) = definedExternally\n    var onpointerdown:
 ((PointerEvent) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n    var
onpointermove: ((PointerEvent) -> dynamic)?\n        get() = definedExternally\n        set(value) =
definedExternally\n    var onpointerup: ((PointerEvent) -> dynamic)?\n        get() = definedExternally\n
set(value) = definedExternally\n    var onpointercancel: ((PointerEvent) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var onpointerover: ((PointerEvent) -> dynamic)?\n
get() = definedExternally\n        set(value) = definedExternally\n    var onpointerout: ((PointerEvent) -> dynamic)?\n
      get() = definedExternally\n        set(value) = definedExternally\n    var onpointerenter: ((PointerEvent) ->
dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n    var onpointerleave:
((PointerEvent) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n}\n\n/**\n
 * Exposes the JavaScript
[WindowEventHandlers](https://developer.mozilla.org/en/docs/Web/API/WindowEventHandlers) to Kotlin\n
*/\npublic external interface WindowEventHandlers {\n    var onafterprint: ((Event) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var onbeforeprint: ((Event) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var onbeforeunload: ((BeforeUnloadEvent) ->
String?)?\n        get() = definedExternally\n        set(value) = definedExternally\n    var onhashchange:
((HashChangeEvent) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n    var
onlanguagechange: ((Event) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n
var onmessage: ((MessageEvent) -> dynamic)?\n        get() = definedExternally\n        set(value) =
definedExternally\n    var onoffline: ((Event) -> dynamic)?\n
   get() = definedExternally\n        set(value) = definedExternally\n    var ononline: ((Event) -> dynamic)?\n
get() = definedExternally\n        set(value) = definedExternally\n    var onpagehide: ((PageTransitionEvent) ->
dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n    var onpageshow:
((PageTransitionEvent) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n    var
onpopstate: ((PopStateEvent) -> dynamic)?\n        get() = definedExternally\n        set(value) = definedExternally\n
var onrejectionhandled: ((Event) -> dynamic)?\n        get() = definedExternally\n        set(value) =
definedExternally\n    var onstorage: ((StorageEvent) -> dynamic)?\n        get() = definedExternally\n        set(value)
= definedExternally\n    var onunhandledrejection: ((PromiseRejectionEvent) -> dynamic)?\n        get() =
definedExternally\n        set(value) = definedExternally\n    var onunload: ((Event) ->
```

dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n}\n\npublic external interface DocumentAndElementEventHandlers {\n var oncopy: ((ClipboardEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var oncut: ((ClipboardEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n var onpaste: ((ClipboardEvent) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n/**\n * Exposes the JavaScript [WindowOrWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/WindowOrWorkerGlobalScope) to Kotlin\n */\npublic external interface WindowOrWorkerGlobalScope {\n val origin: String\n val caches: CacheStorage\n fun btoa(data: String): String\n fun atob(data: String): String\n fun setTimeout(handler: dynamic, timeout: Int = definedExternally, vararg arguments: Any?): Int\n fun clearTimeout(handle: Int = definedExternally)\n

fun setInterval(handler: dynamic, timeout: Int = definedExternally, vararg arguments: Any?): Int\n fun clearInterval(handle: Int = definedExternally)\n fun createImageBitmap(image: ImageBitmapSource, options: ImageBitmapOptions = definedExternally): Promise<ImageBitmap>\n fun createImageBitmap(image: ImageBitmapSource, sx: Int, sy: Int, sw: Int, sh: Int, options: ImageBitmapOptions = definedExternally): Promise<ImageBitmap>\n fun fetch(input: dynamic, init: RequestInit = definedExternally): Promise<Response>\n}\n\n/**\n * Exposes the JavaScript [Navigator](https://developer.mozilla.org/en/docs/Web/API/Navigator) to Kotlin\n */\npublic external abstract class Navigator : NavigatorID, NavigatorLanguage, NavigatorOnLine, NavigatorContentUtils, NavigatorCookies, NavigatorPlugins, NavigatorConcurrentHardware {\n open val clipboard: Clipboard\n open val mediaDevices: MediaDevices\n open val maxTouchPoints: Int\n open val serviceWorker: ServiceWorkerContainer\n

fun requestMediaKeySystemAccess(keySystem: String, supportedConfigurations: Array<MediaKeySystemConfiguration>): Promise<MediaKeySystemAccess>\n fun getUserMedia(constraints: MediaStreamConstraints, successCallback: (MediaStream) -> Unit, errorCallback: (dynamic) -> Unit)\n fun vibrate(pattern: dynamic): Boolean\n}\n\n/**\n * Exposes the JavaScript [NavigatorID](https://developer.mozilla.org/en/docs/Web/API/NavigatorID) to Kotlin\n */\npublic external interface NavigatorID {\n val appCodeName: String\n val appName: String\n val appVersion: String\n val platform: String\n val product: String\n val productSub: String\n val userAgent: String\n val vendor: String\n val vendorSub: String\n val oscpu: String\n fun taintEnabled(): Boolean\n}\n\n/**\n * Exposes the JavaScript [NavigatorLanguage](https://developer.mozilla.org/en/docs/Web/API/NavigatorLanguage) to Kotlin\n */\npublic external interface NavigatorLanguage {\n val language: String\n

val languages: Array<out String>\n}\n\npublic external interface NavigatorContentUtils {\n fun registerProtocolHandler(scheme: String, url: String, title: String)\n fun registerContentHandler(mimeType: String, url: String, title: String)\n fun isProtocolHandlerRegistered(scheme: String, url: String): String\n fun isContentHandlerRegistered(mimeType: String, url: String): String\n fun unregisterProtocolHandler(scheme: String, url: String)\n fun unregisterContentHandler(mimeType: String, url: String)\n}\n\npublic external interface NavigatorCookies {\n val cookieEnabled: Boolean\n}\n\n/**\n * Exposes the JavaScript [NavigatorPlugins](https://developer.mozilla.org/en/docs/Web/API/NavigatorPlugins) to Kotlin\n */\npublic external interface NavigatorPlugins {\n val plugins: PluginArray\n val mimeTypes: MimeTypeArray\n fun javaEnabled(): Boolean\n}\n\n/**\n * Exposes the JavaScript [PluginArray](https://developer.mozilla.org/en/docs/Web/API/PluginArray) to Kotlin\n */\npublic external abstract class PluginArray : ItemArrayLike<Plugin> {\n fun refresh(reload: Boolean = definedExternally)\n override fun item(index: Int): Plugin?\n fun namedItem(name: String): Plugin?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun PluginArray.get(index: Int): Plugin? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun PluginArray.get(name: String): Plugin? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript [MimeTypeArray](https://developer.mozilla.org/en/docs/Web/API/MimeTypeArray) to Kotlin\n */\npublic external

abstract class MimeTypeArray : ItemArrayLike<MimeType> {\n override fun item(index: Int): MimeType?\n fun namedItem(name: String): MimeType?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun MimeTypeArray.get(index: Int): MimeType? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun MimeTypeArray.get(name: String): MimeType? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript [Plugin](https://developer.mozilla.org/en/docs/Web/API/Plugin) to Kotlin\n */\npublic external abstract class Plugin : ItemArrayLike<MimeType> {\n open val name: String\n open val description: String\n open val filename: String\n override fun item(index: Int): MimeType?\n fun namedItem(name: String): MimeType?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Plugin.get(index: Int): MimeType? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Plugin.get(name: String): MimeType? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript [MimeType](https://developer.mozilla.org/en/docs/Web/API/MimeType) to Kotlin\n */\npublic external abstract class MimeType {\n open val type: String\n open val description: String\n open val suffixes: String\n open val enabledPlugin: Plugin\n}\n\n/**\n * Exposes the JavaScript [ImageBitmap](https://developer.mozilla.org/en/docs/Web/API/ImageBitmap) to Kotlin\n */\npublic external abstract class ImageBitmap : CanvasImageSource, TexImageSource {\n open val width: Int\n open val height: Int\n fun close()\n}\n\npublic external interface ImageBitmapOptions {\n var imageOrientation: ImageOrientation? /* = ImageOrientation.NONE */\n get() = definedExternally\n set(value) = definedExternally\n var premultiplyAlpha: PremultiplyAlpha? /* = PremultiplyAlpha.DEFAULT */\n get() = definedExternally\n set(value) = definedExternally\n var colorSpaceConversion: ColorSpaceConversion? /* = ColorSpaceConversion.DEFAULT */\n get() = definedExternally\n set(value) = definedExternally\n var resizeWidth: Int?\n get() = definedExternally\n set(value) = definedExternally\n var resizeHeight: Int?\n get() = definedExternally\n set(value) = definedExternally\n var resizeQuality: ResizeQuality? /* = ResizeQuality.LOW */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ImageBitmapOptions(imageOrientation: ImageOrientation? = ImageOrientation.NONE, premultiplyAlpha: PremultiplyAlpha? = PremultiplyAlpha.DEFAULT, colorSpaceConversion: ColorSpaceConversion? = ColorSpaceConversion.DEFAULT, resizeWidth: Int? = undefined, resizeHeight: Int? = undefined, resizeQuality: ResizeQuality? = ResizeQuality.LOW): ImageBitmapOptions {\n val o = js(\"({})\")\n o[\"imageOrientation\"] = imageOrientation\n o[\"premultiplyAlpha\"] = premultiplyAlpha\n o[\"colorSpaceConversion\"] = colorSpaceConversion\n o[\"resizeWidth\"] = resizeWidth\n o[\"resizeHeight\"] = resizeHeight\n o[\"resizeQuality\"] = resizeQuality\n return o\n}\n\n/**\n * Exposes the JavaScript [MessageEvent](https://developer.mozilla.org/en/docs/Web/API/MessageEvent) to Kotlin\n */\npublic external open class MessageEvent(type: String, eventInitDict: MessageEventInit = definedExternally) : Event {\n open val data: Any?\n open val origin: String\n open val lastEventId: String\n open val source: UnionMessagePortOrWindowProxy?\n open val ports: Array<out MessagePort>\n fun initMessageEvent(type: String, bubbles: Boolean, cancelable: Boolean, data: Any?, origin: String, lastEventId: String, source: UnionMessagePortOrWindowProxy?, ports: Array<MessagePort>)\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface MessageEventInit : EventInit {\n var data: Any? /* = null */\n get() = definedExternally\n set(value) = definedExternally\n var origin: String? /* = \"\" */\n get() = definedExternally\n set(value) = definedExternally\n var lastEventId: String? /* = \"\" */\n get() = definedExternally\n set(value) = definedExternally\n var source:

```
UnionMessagePortOrWindowProxy? /* = null */\n        get() = definedExternally\n        set(value) =
definedExternally\n    var ports: Array<MessagePort>? /* = arrayOf() */\n        get() = definedExternally\n
set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MessageEventInit(data: Any? = null,
origin: String? = \"\", lastEventId: String? = \"\", source: UnionMessagePortOrWindowProxy? = null, ports:
Array<MessagePort>? = arrayOf(), bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? =
false): MessageEventInit {\n    val o
 = js(\"({})\")\n    o[\"data\"] = data\n    o[\"origin\"] = origin\n    o[\"lastEventId\"] = lastEventId\n    o[\"source\"] =
source\n    o[\"ports\"] = ports\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] =
composed\n    return o\n}\n\n/**\n * Exposes the JavaScript
[EventSource](https://developer.mozilla.org/en/docs/Web/API/EventSource) to Kotlin\n */\npublic external open
class EventSource(url: String, eventSourceInitDict: EventSourceInit = definedExternally) : EventTarget {\n    open
val url: String\n    open val withCredentials: Boolean\n    open val readyState: Short\n    var onopen: ((Event) ->
dynamic)?\n    var onmessage: ((MessageEvent) -> dynamic)?\n    var onerror: ((Event) -> dynamic)?\n    fun
close()\n\n    companion object {\n        val CONNECTING: Short\n        val OPEN: Short\n        val CLOSED:
Short\n    }\n}\n\npublic external interface EventSourceInit {\n    var withCredentials: Boolean? /* = false */\n
get() = definedExternally\n
        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun EventSourceInit(withCredentials:
Boolean? = false): EventSourceInit {\n    val o = js(\"({})\")\n    o[\"withCredentials\"] = withCredentials\n    return
o\n}\n\n/**\n * Exposes the JavaScript [WebSocket](https://developer.mozilla.org/en/docs/Web/API/WebSocket) to
Kotlin\n */\npublic external open class WebSocket(url: String, protocols: dynamic = definedExternally) :
EventTarget {\n    open val url: String\n    open val readyState: Short\n    open val bufferedAmount: Number\n    var
onopen: ((Event) -> dynamic)?\n    var onerror: ((Event) -> dynamic)?\n    var onclose: ((Event) -> dynamic)?\n
open val extensions: String\n    open val protocol: String\n    var onmessage: ((MessageEvent) -> dynamic)?\n    var
binaryType: BinaryType\n    fun close(code: Short = definedExternally, reason: String = definedExternally)\n    fun
 send(data: String)\n    fun send(data: Blob)\n    fun send(data: ArrayBuffer)\n    fun send(data:
ArrayBufferView)\n\n    companion object {\n        val CONNECTING: Short\n        val OPEN: Short\n        val
CLOSING: Short\n        val CLOSED: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[CloseEvent](https://developer.mozilla.org/en/docs/Web/API/CloseEvent) to Kotlin\n */\npublic external open class
CloseEvent(type: String, eventInitDict: CloseEventInit = definedExternally) : Event {\n    open val wasClean:
Boolean\n    open val code: Short\n    open val reason: String\n\n    companion object {\n        val NONE: Short\n
 val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n
}\n}\n\npublic external interface CloseEventInit : EventInit {\n    var wasClean: Boolean? /* = false */\n        get() =
definedExternally\n        set(value) = definedExternally\n    var code: Short? /* = 0 */\n        get() =
definedExternally\n        set(value)
 = definedExternally\n    var reason: String? /* = \"\" */\n        get() = definedExternally\n        set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun CloseEventInit(wasClean: Boolean? =
false, code: Short? = 0, reason: String? = \"\", bubbles: Boolean? = false, cancelable: Boolean? = false, composed:
Boolean? = false): CloseEventInit {\n    val o = js(\"({})\")\n    o[\"wasClean\"] = wasClean\n    o[\"code\"] = code\n
  o[\"reason\"] = reason\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] =
composed\n    return o\n}\n\n/**\n * Exposes the JavaScript
[MessageChannel](https://developer.mozilla.org/en/docs/Web/API/MessageChannel) to Kotlin\n */\npublic external
open class MessageChannel {\n    open val port1: MessagePort\n    open val port2: MessagePort\n}\n\n/**\n *
Exposes the JavaScript [MessagePort](https://developer.mozilla.org/en/docs/Web/API/MessagePort)
 to Kotlin\n */\npublic external abstract class MessagePort : EventTarget, UnionMessagePortOrWindowProxy,
UnionMessagePortOrServiceWorker, UnionClientOrMessagePortOrServiceWorker {\n    open var onmessage:
```

((MessageEvent) -> dynamic)?\n   fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)\n   fun start()\n   fun close()\n}\n\n/**\n * Exposes the JavaScript [BroadcastChannel](https://developer.mozilla.org/en/docs/Web/API/BroadcastChannel) to Kotlin\n */\npublic external open class BroadcastChannel(name: String) : EventTarget {\n   open val name: String\n   var onmessage: ((MessageEvent) -> dynamic)?\n   fun postMessage(message: Any?)\n   fun close()\n}\n\n/**\n * Exposes the JavaScript [WorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/WorkerGlobalScope) to Kotlin\n */\npublic external abstract class WorkerGlobalScope : EventTarget, WindowOrWorkerGlobalScope, GlobalPerformance {\n   open val self: WorkerGlobalScope\n   open val location: WorkerLocation\n   open val navigator: WorkerNavigator\n   open var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?\n   open var onlanguagechange: ((Event) -> dynamic)?\n   open var onoffline: ((Event) -> dynamic)?\n   open var ononline: ((Event) -> dynamic)?\n   open var onrejectionhandled: ((Event) -> dynamic)?\n   open var onunhandledrejection: ((PromiseRejectionEvent) -> dynamic)?\n   fun importScripts(vararg urls: String)\n}\n\n/**\n * Exposes the JavaScript [DedicatedWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/DedicatedWorkerGlobalScope) to Kotlin\n */\npublic external abstract class DedicatedWorkerGlobalScope : WorkerGlobalScope {\n   open var onmessage: ((MessageEvent) -> dynamic)?\n   fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)\n   fun close()\n}\n\n/**\n * Exposes the JavaScript [SharedWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/SharedWorkerGlobalScope) to Kotlin\n */\npublic
external abstract class SharedWorkerGlobalScope : WorkerGlobalScope {\n   open val name: String\n   open val applicationCache: ApplicationCache\n   open var onconnect: ((Event) -> dynamic)?\n   fun close()\n}\n\n/**\n * Exposes the JavaScript [AbstractWorker](https://developer.mozilla.org/en/docs/Web/API/AbstractWorker) to Kotlin\n */\npublic external interface AbstractWorker {\n   var onerror: ((Event) -> dynamic)?\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n/**\n * Exposes the JavaScript [Worker](https://developer.mozilla.org/en/docs/Web/API/Worker) to Kotlin\n */\npublic external open class Worker(scriptURL: String, options: WorkerOptions = definedExternally) : EventTarget, AbstractWorker {\n   var onmessage: ((MessageEvent) -> dynamic)?\n   override var onerror: ((Event) -> dynamic)?\n   fun terminate()\n   fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)\n}\n\npublic external interface WorkerOptions
{\n   var type: WorkerType? /* = WorkerType.CLASSIC */\n   get() = definedExternally\n   set(value) = definedExternally\n   var credentials: RequestCredentials? /* = RequestCredentials.OMIT */\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun WorkerOptions(type: WorkerType? = WorkerType.CLASSIC, credentials: RequestCredentials? = RequestCredentials.OMIT): WorkerOptions {\n   val o = js(\"({})\")\n   o[\"type\"] = type\n   o[\"credentials\"] = credentials\n   return o\n}\n\n/**\n * Exposes the JavaScript [SharedWorker](https://developer.mozilla.org/en/docs/Web/API/SharedWorker) to Kotlin\n */\npublic external open class SharedWorker(scriptURL: String, name: String = definedExternally, options: WorkerOptions = definedExternally) : EventTarget, AbstractWorker {\n   open val port: MessagePort\n   override var onerror: ((Event) -> dynamic)?\n}\n\n/**\n
 * Exposes the JavaScript [NavigatorConcurrentHardware](https://developer.mozilla.org/en/docs/Web/API/NavigatorConcurrentHardware) to Kotlin\n */\npublic external interface NavigatorConcurrentHardware {\n   val hardwareConcurrency: Number\n}\n\n/**\n * Exposes the JavaScript [WorkerNavigator](https://developer.mozilla.org/en/docs/Web/API/WorkerNavigator) to Kotlin\n */\npublic external abstract class WorkerNavigator : NavigatorID, NavigatorLanguage, NavigatorOnLine, NavigatorConcurrentHardware {\n   open val serviceWorker: ServiceWorkerContainer\n}\n\n/**\n * Exposes the JavaScript [WorkerLocation](https://developer.mozilla.org/en/docs/Web/API/WorkerLocation) to Kotlin\n */\npublic external abstract class WorkerLocation {\n   open val href: String\n   open val origin: String\n   open val

protocol: String\n   open val host: String\n   open val hostname: String\n   open val port: String\n   open val pathname: String\n   open val search: String\n   open val hash: String\n}\n\n/**\n * Exposes the JavaScript [Storage](https://developer.mozilla.org/en/docs/Web/API/Storage) to Kotlin\n */\npublic external abstract class Storage {\n   open val length: Int\n   fun key(index: Int): String?\n   fun removeItem(key: String)\n   fun clear()\n   fun getItem(key: String): String?\n   fun setItem(key: String, value: String)\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Storage.get(key: String): String? = asDynamic()[key]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Storage.set(key: String, value: String) { asDynamic()[key] = value }\n\n/**\n * Exposes the JavaScript [WindowSessionStorage](https://developer.mozilla.org/en/docs/Web/API/WindowSessionStorage) to Kotlin\n */\npublic external interface WindowSessionStorage {\n   val sessionStorage: Storage\n}\n\n/**\n * Exposes the JavaScript [WindowLocalStorage](https://developer.mozilla.org/en/docs/Web/API/WindowLocalStorage) to Kotlin\n */\npublic external interface WindowLocalStorage {\n   val localStorage: Storage\n}\n\n/**\n * Exposes the JavaScript [StorageEvent](https://developer.mozilla.org/en/docs/Web/API/StorageEvent) to Kotlin\n */\npublic external open class StorageEvent(type: String, eventInitDict: StorageEventInit = definedExternally) : Event {\n   open val key: String?\n   open val oldValue: String?\n   open val newValue: String?\n   open val url: String\n   open val storageArea: Storage?\n\n   companion object {\n   val NONE: Short\n   val CAPTURING_PHASE: Short\n   val AT_TARGET: Short\n   val BUBBLING_PHASE: Short\n   }\n}\n\npublic external interface StorageEventInit : EventInit {\n   var key: String? /* = null */\n   get() = definedExternally\n   set(value) = definedExternally\n   var oldValue: String? /* = null */\n   get() = definedExternally\n   set(value) = definedExternally\n   var newValue: String? /* = null */\n   get() = definedExternally\n   set(value) = definedExternally\n   var url: String? /* = \"\" */\n   get() = definedExternally\n   set(value) = definedExternally\n   var storageArea: Storage? /* = null */\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun StorageEventInit(key: String? = null, oldValue: String? = null, newValue: String? = null, url: String? = \"\", storageArea: Storage? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): StorageEventInit {\n   val o = js(\"({})\")\n   o[\"key\"] = key\n   o[\"oldValue\"] = oldValue\n   o[\"newValue\"] = newValue\n   o[\"url\"] = url\n   o[\"storageArea\"] = storageArea\n   o[\"bubbles\"] = bubbles\n   o[\"cancelable\"] = cancelable\n   o[\"composed\"] = composed\n   return o\n}\n\npublic external abstract class HTMLAppletElement : HTMLElement {\n   open var align: String\n   open var alt: String\n   open var archive: String\n   open var code: String\n   open var codeBase: String\n   open var height: String\n   open var hspace: Int\n   open var name: String\n   open var _object: String\n   open var vspace: Int\n   open var width: String\n\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n   val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n   val DOCUMENT_POSITION_FOLLOWING: Short\n   val DOCUMENT_POSITION_CONTAINS: Short\n   val DOCUMENT_POSITION_CONTAINED_BY: Short\n   val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n   }\n}\n\n/**\n * Exposes the JavaScript [HTMLMarqueeElement](https://developer.mozilla.org/en/docs/Web/API/HTMLMarqueeElement) to Kotlin\n */\npublic external abstract class HTMLMarqueeElement : HTMLElement {\n   open var behavior: String\n   open var bgColor: String\n   open var direction: String\n   open var height: String\n   open var hspace: Int\n   open var

loop: Int\n   open var scrollAmount: Int\n   open var scrollDelay: Int\n   open var trueSpeed: Boolean\n   open var vspace: Int\n   open var width: String\n   open var onbounce: ((Event) -> dynamic)?\n   open var onfinish: ((Event) -> dynamic)?\n   open var onstart: ((Event) -> dynamic)?\n   fun start()\n   fun stop()\n\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE:
 Short\n   val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n   val DOCUMENT_POSITION_FOLLOWING: Short\n   val DOCUMENT_POSITION_CONTAINS: Short\n   val DOCUMENT_POSITION_CONTAINED_BY: Short\n   val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n   }\n}\n\n/**\n * Exposes the JavaScript [HTMLFrameSetElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFrameSetElement) to Kotlin\n */\npublic external abstract class HTMLFrameSetElement : HTMLElement, WindowEventHandlers {\n   open var cols: String\n   open var rows: String\n\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE:
 Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n   val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n   val DOCUMENT_POSITION_FOLLOWING: Short\n   val DOCUMENT_POSITION_CONTAINS: Short\n   val DOCUMENT_POSITION_CONTAINED_BY: Short\n   val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n   }\n}\n\npublic external abstract class HTMLFrameElement : HTMLElement {\n   open var name: String\n   open var scrolling: String\n   open var src: String\n   open var frameBorder: String\n   open var longDesc: String\n   open var noResize: Boolean\n   open val contentDocument: Document?\n   open
 val contentWindow: Window?\n   open var marginHeight: String\n   open var marginWidth: String\n\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n   val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n   val DOCUMENT_POSITION_FOLLOWING: Short\n   val DOCUMENT_POSITION_CONTAINS: Short\n   val DOCUMENT_POSITION_CONTAINED_BY: Short\n   val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n   }\n}\n\npublic external abstract class HTMLDirectoryElement : HTMLElement {\n   open var compact: Boolean\n\n   companion
 object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n   val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n   val DOCUMENT_POSITION_FOLLOWING: Short\n   val DOCUMENT_POSITION_CONTAINS: Short\n   val DOCUMENT_POSITION_CONTAINED_BY: Short\n   val

```
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript [HTMLFontElement](https://developer.mozilla.org/en/docs/Web/API/HTMLFontElement) to Kotlin\n */\npublic external abstract class HTMLFontElement : HTMLElement {\n    open var color: String\n    open var face: String\n    open var size: String\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external interface External {\n    fun AddSearchProvider()\n    fun IsSearchProviderInstalled()\n}\n\npublic external interface EventInit {\n    var bubbles: Boolean?\n        /* = false */\n        get() = definedExternally\n        set(value) = definedExternally\n    var cancelable: Boolean? /* = false */\n        get() = definedExternally\n        set(value) = definedExternally\n    var composed: Boolean? /* = false */\n        get() = definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun EventInit(bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): EventInit {\n    val o = js(\"({})\")\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\n/**\n * Exposes the JavaScript [CustomEvent](https://developer.mozilla.org/en/docs/Web/API/CustomEvent) to Kotlin\n */\npublic external open class CustomEvent(type: String, eventInitDict: CustomEventInit = definedExternally) : Event {\n    open val detail: Any?\n    fun initCustomEvent(type: String, bubbles: Boolean, cancelable: Boolean, detail: Any?)\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface CustomEventInit : EventInit {\n    var detail: Any? /* = null */\n        get() = definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun CustomEventInit(detail: Any? = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): CustomEventInit {\n    val o = js(\"({})\")\n    o[\"detail\"] = detail\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\npublic external interface EventListenerOptions {\n    var capture: Boolean? /* = false */\n        get() = definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun EventListenerOptions(capture: Boolean? = false): EventListenerOptions {\n    val o = js(\"({})\")\n    o[\"capture\"] = capture\n    return o\n}\n\npublic external interface AddEventListenerOptions : EventListenerOptions {\n    var passive: Boolean? /* = false */\n        get() = definedExternally\n        set(value) = definedExternally\n    var once: Boolean? /* = false */\n        get() = definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun AddEventListenerOptions(passive: Boolean? = false, once: Boolean? = false, capture: Boolean? = false): AddEventListenerOptions {\n    val o = js(\"({})\")\n    o[\"passive\"] = passive\n    o[\"once\"] = once\n    o[\"capture\"] = capture\n    return o\n}\n\npublic external interface NonElementParentNode {\n    fun getElementById(elementId: String): Element?\n}\n\n/**\n * Exposes the JavaScript [DocumentOrShadowRoot](https://developer.mozilla.org/en/docs/Web/API/DocumentOrShadowRoot) to Kotlin\n */\npublic external interface DocumentOrShadowRoot {\n    val fullscreenElement: Element?\n        get() = definedExternally\n}\n\n/**\n * Exposes the JavaScript
```

[ParentNode](https://developer.mozilla.org/en/docs/Web/API/ParentNode) to Kotlin\n */\npublic external interface ParentNode {\n val children: HTMLCollection\n val firstElementChild: Element?\n get() = definedExternally\n val lastElementChild: Element?\n get() = definedExternally\n val childElementCount: Int\n fun prepend(vararg nodes: dynamic)\n fun append(vararg nodes: dynamic)\n fun querySelector(selectors: String): Element?\n fun querySelectorAll(selectors: String): NodeList\n}\n\n/**\n * Exposes the JavaScript [NonDocumentTypeChildNode](https://developer.mozilla.org/en/docs/Web/API/NonDocumentTypeChildNode) to Kotlin\n */\npublic external interface NonDocumentTypeChildNode {\n val previousElementSibling: Element?\n get() = definedExternally\n val nextElementSibling: Element?\n get() = definedExternally\n}\n\n/**\n * Exposes the JavaScript [ChildNode](https://developer.mozilla.org/en/docs/Web/API/ChildNode) to Kotlin\n */\npublic external interface ChildNode {\n fun before(vararg nodes: dynamic)\n fun after(vararg nodes: dynamic)\n fun replaceWith(vararg nodes: dynamic)\n fun remove()\n}\n\n/**\n * Exposes the JavaScript [Slotable](https://developer.mozilla.org/en/docs/Web/API/Slotable) to Kotlin\n */\npublic external interface Slotable {\n val assignedSlot: HTMLSlotElement?\n get() = definedExternally\n}\n\n/**\n * Exposes the JavaScript [NodeList](https://developer.mozilla.org/en/docs/Web/API/NodeList) to Kotlin\n */\npublic external abstract class NodeList : ItemArrayLike<Node> {\n override fun item(index: Int): Node?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun NodeList.get(index: Int): Node? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript [HTMLCollection](https://developer.mozilla.org/en/docs/Web/API/HTMLCollection) to Kotlin\n */\npublic external abstract class HTMLCollection : ItemArrayLike<Element>, UnionElementOrHTMLCollection {\n override fun item(index: Int): Element?\n fun namedItem(name: String): Element?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun HTMLCollection.get(index: Int): Element? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun HTMLCollection.get(name: String): Element? = asDynamic()[name]\n\n/**\n * Exposes the JavaScript [MutationObserver](https://developer.mozilla.org/en/docs/Web/API/MutationObserver) to Kotlin\n */\npublic external open class MutationObserver(callback: (Array<MutationRecord>, MutationObserver) -> Unit) {\n fun observe(target: Node, options: MutationObserverInit = definedExternally)\n fun disconnect()\n fun takeRecords(): Array<MutationRecord>\n}\n\n/**\n * Exposes the JavaScript [MutationObserverInit](https://developer.mozilla.org/en/docs/Web/API/MutationObserverInit) to Kotlin\n */\npublic external interface MutationObserverInit {\n var childList: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var attributes: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var characterData: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var subtree: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var attributeOldValue: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var characterDataOldValue: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var attributeFilter: Array<String>?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MutationObserverInit(childList: Boolean? = false, attributes: Boolean? = undefined, characterData: Boolean? = undefined, subtree: Boolean? = false, attributeOldValue: Boolean? = undefined, characterDataOldValue: Boolean? = undefined, attributeFilter: Array<String>? = undefined): MutationObserverInit {\n val o = js(\"({})\")\n o[\"childList\"] = childList\n o[\"attributes\"] = attributes\n o[\"characterData\"] = characterData\n o[\"subtree\"] = subtree\n o[\"attributeOldValue\"] = attributeOldValue\n o[\"characterDataOldValue\"] = characterDataOldValue\n o[\"attributeFilter\"] = attributeFilter\n return o\n}\n\n/**\n * Exposes the JavaScript

[MutationRecord](https://developer.mozilla.org/en/docs/Web/API/MutationRecord)
to Kotlin\n */\npublic external abstract class MutationRecord {\n open val type: String\n open val target: Node\n open val addedNodes: NodeList\n open val removedNodes: NodeList\n open val previousSibling: Node?\n open val nextSibling: Node?\n open val attributeName: String?\n open val attributeNamespace: String?\n open val oldValue: String?\n}\n\n/**\n * Exposes the JavaScript
[Node](https://developer.mozilla.org/en/docs/Web/API/Node) to Kotlin\n */\npublic external abstract class Node : EventTarget {\n open val nodeType: Short\n open val nodeName: String\n open val baseURI: String\n open val isConnected: Boolean\n open val ownerDocument: Document?\n open val parentNode: Node?\n open val parentElement: Element?\n open val childNodes: NodeList\n open val firstChild: Node?\n open val lastChild: Node?\n open val previousSibling: Node?\n open val nextSibling: Node?\n open var nodeValue: String?\n open var textContent: String?\n fun getRootNode(options: GetRootNodeOptions = definedExternally): Node\n fun hasChildNodes(): Boolean\n fun normalize()\n fun cloneNode(deep: Boolean = definedExternally): Node\n fun isEqualNode(otherNode: Node?): Boolean\n fun isSameNode(otherNode: Node?): Boolean\n fun compareDocumentPosition(other: Node): Short\n fun contains(other: Node?): Boolean\n fun lookupPrefix(namespace: String?): String?\n fun lookupNamespaceURI(prefix: String?): String?\n fun isDefaultNamespace(namespace: String?): Boolean\n fun insertBefore(node: Node, child: Node?): Node\n fun appendChild(node: Node): Node\n fun replaceChild(node: Node, child: Node): Node\n fun removeChild(child: Node): Node\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external interface GetRootNodeOptions {\n var composed: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun GetRootNodeOptions(composed: Boolean? = false): GetRootNodeOptions {\n val o = js(\"({})\")\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript
[Document](https://developer.mozilla.org/en/docs/Web/API/Document) to Kotlin\n */\npublic external open class Document : Node, GlobalEventHandlers, DocumentAndElementEventHandlers, NonElementParentNode, DocumentOrShadowRoot, ParentNode, GeometryUtils {\n open val implementation: DOMImplementation\n open val URL: String\n open val documentURI: String\n open val origin: String\n open val compatMode: String\n open val characterSet: String\n open val charset: String\n open val inputEncoding: String\n open val contentType: String\n open val doctype: DocumentType?\n open val documentElement: Element?\n open val location: Location?\n var domain: String\n open val referrer: String\n var cookie: String\n open val lastModified: String\n open val readyState: DocumentReadyState\n var title: String\n var dir: String\n var body: HTMLElement?\n open val head: HTMLHeadElement?\n open val images: HTMLCollection\n open val embeds:
HTMLCollection\n open val plugins: HTMLCollection\n open val links: HTMLCollection\n open val forms: HTMLCollection\n open val scripts: HTMLCollection\n open val currentScript: HTMLOrSVGScriptElement?\n open val defaultView: Window?\n open val activeElement: Element?\n var designMode: String\n var onreadystatechange: ((Event) -> dynamic)?\n var fgColor: String\n var linkColor: String\n var vlinkColor: String\n var alinkColor: String\n var bgColor: String\n open val anchors: HTMLCollection\n open val applets: HTMLCollection\n open val all: HTMLAllCollection\n open val scrollingElement: Element?\n open

val styleSheets: StyleSheetList\n   open val rootElement: SVGSVGElement?\n   open val fullscreenEnabled: Boolean\n   open val fullscreen: Boolean\n   var onfullscreenchange: ((Event) -> dynamic)?\n   var onfullscreenerror: ((Event) -> dynamic)?\n   override var onabort: ((Event) -> dynamic)?\n   override var onblur: ((FocusEvent)
 -> dynamic)?\n   override var oncancel: ((Event) -> dynamic)?\n   override var oncanplay: ((Event) -> dynamic)?\n   override var oncanplaythrough: ((Event) -> dynamic)?\n   override var onchange: ((Event) -> dynamic)?\n   override var onclick: ((MouseEvent) -> dynamic)?\n   override var onclose: ((Event) -> dynamic)?\n   override var oncontextmenu: ((MouseEvent) -> dynamic)?\n   override var oncuechange: ((Event) -> dynamic)?\n   override var ondblclick: ((MouseEvent) -> dynamic)?\n   override var ondrag: ((DragEvent) -> dynamic)?\n   override var ondragend: ((DragEvent) -> dynamic)?\n   override var ondragenter: ((DragEvent) -> dynamic)?\n   override var ondragexit: ((DragEvent) -> dynamic)?\n   override var ondragleave: ((DragEvent) -> dynamic)?\n   override var ondragover: ((DragEvent) -> dynamic)?\n   override var ondragstart: ((DragEvent) -> dynamic)?\n   override var ondrop: ((DragEvent) -> dynamic)?\n   override var ondurationchange: ((Event) -> dynamic)?\n   override var onemptied: ((Event) -> dynamic)?\n   override var onended: ((Event) -> dynamic)?\n   override var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?\n   override var onfocus: ((FocusEvent) -> dynamic)?\n   override var oninput: ((InputEvent) -> dynamic)?\n   override var oninvalid: ((Event) -> dynamic)?\n   override var onkeydown: ((KeyboardEvent) -> dynamic)?\n   override var onkeypress: ((KeyboardEvent) -> dynamic)?\n   override var onkeyup: ((KeyboardEvent) -> dynamic)?\n   override var onload: ((Event) -> dynamic)?\n   override var onloadeddata: ((Event) -> dynamic)?\n   override var onloadedmetadata: ((Event) -> dynamic)?\n   override var onloadend: ((Event) -> dynamic)?\n   override var onloadstart: ((ProgressEvent) -> dynamic)?\n   override var onmousedown: ((MouseEvent) -> dynamic)?\n   override var onmouseenter: ((MouseEvent) -> dynamic)?\n   override var onmouseleave: ((MouseEvent) -> dynamic)?\n   override var onmousemove: ((MouseEvent)
 -> dynamic)?\n   override var onmouseout: ((MouseEvent) -> dynamic)?\n   override var onmouseover: ((MouseEvent) -> dynamic)?\n   override var onmouseup: ((MouseEvent) -> dynamic)?\n   override var onwheel: ((WheelEvent) -> dynamic)?\n   override var onpause: ((Event) -> dynamic)?\n   override var onplay: ((Event) -> dynamic)?\n   override var onplaying: ((Event) -> dynamic)?\n   override var onprogress: ((ProgressEvent) -> dynamic)?\n   override var onratechange: ((Event) -> dynamic)?\n   override var onreset: ((Event) -> dynamic)?\n   override var onresize: ((Event) -> dynamic)?\n   override var onscroll: ((Event) -> dynamic)?\n   override var onseeked: ((Event) -> dynamic)?\n   override var onseeking: ((Event) -> dynamic)?\n   override var onselect: ((Event) -> dynamic)?\n   override var onshow: ((Event) -> dynamic)?\n   override var onstalled: ((Event) -> dynamic)?\n   override var onsubmit: ((Event) -> dynamic)?\n   override var onsuspend: ((Event) -> dynamic)?\n   override var ontimeupdate: ((Event) -> dynamic)?\n   override var ontoggle: ((Event) -> dynamic)?\n   override var onvolumechange: ((Event) -> dynamic)?\n   override var onwaiting: ((Event) -> dynamic)?\n   override var ongotpointercapture: ((PointerEvent) -> dynamic)?\n   override var onlostpointercapture: ((PointerEvent) -> dynamic)?\n   override var onpointerdown: ((PointerEvent) -> dynamic)?\n   override var onpointermove: ((PointerEvent) -> dynamic)?\n   override var onpointerup: ((PointerEvent) -> dynamic)?\n   override var onpointercancel: ((PointerEvent) -> dynamic)?\n   override var onpointerover: ((PointerEvent) -> dynamic)?\n   override var onpointerout: ((PointerEvent) -> dynamic)?\n   override var onpointerenter: ((PointerEvent) -> dynamic)?\n   override var onpointerleave: ((PointerEvent) -> dynamic)?\n   override var oncopy: ((ClipboardEvent) -> dynamic)?\n   override var oncut: ((ClipboardEvent) -> dynamic)?\n   override var onpaste: ((ClipboardEvent)
 -> dynamic)?\n   override val fullscreenElement: Element?\n   override val children: HTMLCollection\n   override val firstElementChild: Element?\n   override val lastElementChild: Element?\n   override val childElementCount: Int\n   fun getElementsByTagName(qualifiedName: String): HTMLCollection\n   fun getElementsByTagNameNS(namespace: String?, localName: String): HTMLCollection\n   fun getElementsByClassName(classNames: String): HTMLCollection\n   fun createElement(localName: String, options: ElementCreationOptions = definedExternally): Element\n   fun createElementNS(namespace: String?, qualifiedName: String, options: ElementCreationOptions = definedExternally): Element\n   fun

createDocumentFragment(): DocumentFragment\n   fun createTextNode(data: String): Text\n   fun createCDATASection(data: String): CDATASection\n   fun createComment(data: String): Comment\n   fun createProcessingInstruction(target: String, data: String): ProcessingInstruction\n   fun importNode(node: Node, deep: Boolean = definedExternally): Node\n   fun adoptNode(node: Node): Node\n   fun createAttribute(localName: String): Attr\n   fun createAttributeNS(namespace: String?, qualifiedName: String): Attr\n   fun createEvent(`interface`: String): Event\n   fun createRange(): Range\n   fun createNodeIterator(root: Node, whatToShow: Int = definedExternally, filter: NodeFilter? = definedExternally): NodeIterator\n   fun createNodeIterator(root: Node, whatToShow: Int = definedExternally, filter: ((Node) -> Short)? = definedExternally): NodeIterator\n   fun createTreeWalker(root: Node, whatToShow: Int = definedExternally, filter: NodeFilter? = definedExternally): TreeWalker\n   fun createTreeWalker(root: Node, whatToShow: Int = definedExternally, filter: ((Node) -> Short)? = definedExternally): TreeWalker\n   fun getElementsByName(elementName: String): NodeList\n   fun open(type: String = definedExternally, replace: String = definedExternally): Document\n   fun open(url: String, name: String, features: String): Window\n   fun close()\n   fun write(vararg text: String)\n   fun writeln(vararg text: String)\n   fun hasFocus(): Boolean\n   fun execCommand(commandId: String, showUI: Boolean = definedExternally, value: String = definedExternally): Boolean\n   fun queryCommandEnabled(commandId: String): Boolean\n   fun queryCommandIndeterm(commandId: String): Boolean\n   fun queryCommandState(commandId: String): Boolean\n   fun queryCommandSupported(commandId: String): Boolean\n   fun queryCommandValue(commandId: String): String\n   fun clear()\n   fun captureEvents()\n   fun releaseEvents()\n   fun elementFromPoint(x: Double, y: Double): Element?\n   fun elementsFromPoint(x: Double, y: Double): Array<Element>\n   fun caretPositionFromPoint(x: Double, y: Double): CaretPosition?\n   fun createTouch(view: Window, target: EventTarget, identifier: Int, pageX: Int, pageY: Int, screenX: Int, screenY: Int): Touch\n   fun createTouchList(vararg touches: Touch): TouchList\n   fun exitFullscreen(): Promise<Unit>\n   override fun getElementById(elementId: String): Element?\n   override fun prepend(vararg nodes: dynamic)\n   override fun append(vararg nodes: dynamic)\n   override fun querySelector(selectors: String): Element?\n   override fun querySelectorAll(selectors: String): NodeList\n   override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n   override fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n   override fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n   override fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint\n\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n   val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n   val DOCUMENT_POSITION_FOLLOWING: Short\n   val DOCUMENT_POSITION_CONTAINS: Short\n   val DOCUMENT_POSITION_CONTAINED_BY: Short\n   val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n   }\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun Document.get(name: String): dynamic = asDynamic()[name]\n\n/**\n * Exposes the JavaScript [XMLDocument](https://developer.mozilla.org/en/docs/Web/API/XMLDocument) to Kotlin\n */\npublic external open class XMLDocument : Document {\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n   val

ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external interface ElementCreationOptions {\n var `is`: String?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ElementCreationOptions(`is`: String? = undefined): ElementCreationOptions {\n val o = js(\"({})\")\n o[\"is\"] = `is`\n return o\n}\n\n/**\n * Exposes the JavaScript [DOMImplementation](https://developer.mozilla.org/en/docs/Web/API/DOMImplementation) to Kotlin\n */\npublic external abstract class DOMImplementation {\n fun createDocumentType(qualifiedName: String, publicId: String, systemId: String): DocumentType\n fun createDocument(namespace: String?, qualifiedName: String, doctype: DocumentType? = definedExternally): XMLDocument\n fun createHTMLDocument(title: String = definedExternally): Document\n fun hasFeature(): Boolean\n}\n\n/**\n * Exposes the JavaScript [DocumentType](https://developer.mozilla.org/en/docs/Web/API/DocumentType) to Kotlin\n */\npublic external abstract class DocumentType : Node, ChildNode {\n open val name: String\n open val publicId: String\n open val systemId: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [DocumentFragment](https://developer.mozilla.org/en/docs/Web/API/DocumentFragment) to Kotlin\n */\npublic external open class DocumentFragment : Node, NonElementParentNode, ParentNode {\n override val children: HTMLCollection\n override val firstElementChild: Element?\n override val lastElementChild: Element?\n override val childElementCount: Int\n override fun getElementById(elementId: String): Element?\n override fun prepend(vararg nodes: dynamic)\n override fun append(vararg nodes: dynamic)\n override fun querySelector(selectors: String): Element?\n override fun querySelectorAll(selectors: String): NodeList\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [ShadowRoot](https://developer.mozilla.org/en/docs/Web/API/ShadowRoot) to Kotlin\n */\npublic external open class ShadowRoot : DocumentFragment, DocumentOrShadowRoot {\n open val mode:

ShadowRootMode\n   open val host: Element\n   override val fullscreenElement: Element?\n\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n   val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n
    val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n   val DOCUMENT_POSITION_FOLLOWING: Short\n   val DOCUMENT_POSITION_CONTAINS: Short\n   val DOCUMENT_POSITION_CONTAINED_BY: Short\n   val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n   }\n}\n\n/**\n * Exposes the JavaScript [Element](https://developer.mozilla.org/en/docs/Web/API/Element) to Kotlin\n */\npublic external abstract class Element : Node, ParentNode, NonDocumentTypeChildNode, ChildNode, Slotable, GeometryUtils, UnionElementOrHTMLCollection, UnionElementOrRadioNodeList, UnionElementOrMouseEvent, UnionElementOrProcessingInstruction {\n   open val namespaceURI: String?\n   open val prefix: String?\n   open val localName: String\n   open val tagName: String\n   open var id: String\n   open var className: String\n   open val classList: DOMTokenList\n   open var slot: String\n   open val attributes: NamedNodeMap\n   open val shadowRoot: ShadowRoot?\n   open var scrollTop: Double\n   open var scrollLeft: Double\n   open val scrollWidth: Int\n   open val scrollHeight: Int\n   open val clientTop: Int\n   open val clientLeft: Int\n   open val clientWidth: Int\n   open val clientHeight: Int\n   open var innerHTML: String\n   open var outerHTML: String\n   fun hasAttributes(): Boolean\n   fun getAttributeNames(): Array<String>\n   fun getAttribute(qualifiedName: String): String?\n   fun getAttributeNS(namespace: String?, localName: String): String?\n   fun setAttribute(qualifiedName: String, value: String)\n   fun setAttributeNS(namespace: String?, qualifiedName: String, value: String)\n   fun removeAttribute(qualifiedName: String)\n   fun removeAttributeNS(namespace: String?, localName: String)\n   fun hasAttribute(qualifiedName: String): Boolean\n   fun hasAttributeNS(namespace: String?, localName: String): Boolean\n   fun getAttributeNode(qualifiedName: String): Attr?\n   fun getAttributeNodeNS(namespace:
 String?, localName: String): Attr?\n   fun setAttributeNode(attr: Attr): Attr?\n   fun setAttributeNodeNS(attr: Attr): Attr?\n   fun removeAttributeNode(attr: Attr): Attr\n   fun attachShadow(init: ShadowRootInit): ShadowRoot\n   fun closest(selectors: String): Element?\n   fun matches(selectors: String): Boolean\n   fun webkitMatchesSelector(selectors: String): Boolean\n   fun getElementsByTagName(qualifiedName: String): HTMLCollection\n   fun getElementsByTagNameNS(namespace: String?, localName: String): HTMLCollection\n   fun getElementsByClassName(classNames: String): HTMLCollection\n   fun insertAdjacentElement(where: String, element: Element): Element?\n   fun insertAdjacentText(where: String, data: String)\n   fun getClientRects(): Array<DOMRect>\n   fun getBoundingClientRect(): DOMRect\n   fun scrollIntoView()\n   fun scrollIntoView(arg: dynamic)\n   fun scroll(options: ScrollToOptions = definedExternally)\n   fun scroll(x: Double, y:
 Double)\n   fun scrollTo(options: ScrollToOptions = definedExternally)\n   fun scrollTo(x: Double, y: Double)\n   fun scrollBy(options: ScrollToOptions = definedExternally)\n   fun scrollBy(x: Double, y: Double)\n   fun insertAdjacentHTML(position: String, text: String)\n   fun setPointerCapture(pointerId: Int)\n   fun releasePointerCapture(pointerId: Int)\n   fun hasPointerCapture(pointerId: Int): Boolean\n   fun requestFullscreen(): Promise<Unit>\n\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n   val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING:

Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external interface ShadowRootInit {\n var mode: ShadowRootMode?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ShadowRootInit(mode: ShadowRootMode?): ShadowRootInit {\n val o = js(\"({})\")\n o[\"mode\"] = mode\n return o\n}\n\n/**\n * Exposes the JavaScript [NamedNodeMap](https://developer.mozilla.org/en/docs/Web/API/NamedNodeMap) to Kotlin\n */\npublic external abstract class NamedNodeMap : ItemArrayLike<Attr> {\n fun getNamedItemNS(namespace: String?, localName: String): Attr?\n fun setNamedItem(attr: Attr): Attr?\n fun setNamedItemNS(attr: Attr): Attr?\n fun removeNamedItem(qualifiedName: String): Attr\n fun removeNamedItemNS(namespace: String?, localName: String): Attr\n override fun item(index: Int): Attr?\n fun getNamedItem(qualifiedName: String): Attr?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun NamedNodeMap.get(index: Int): Attr? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun NamedNodeMap.get(qualifiedName: String): Attr? = asDynamic()[qualifiedName]\n\n/**\n * Exposes the JavaScript [Attr](https://developer.mozilla.org/en/docs/Web/API/Attr) to Kotlin\n */\npublic external abstract class Attr : Node {\n open val namespaceURI: String?\n open val prefix: String?\n open val localName: String\n open val name: String\n open var value: String\n open val ownerElement: Element?\n open val specified: Boolean\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [CharacterData](https://developer.mozilla.org/en/docs/Web/API/CharacterData) to Kotlin\n */\npublic external abstract class CharacterData : Node, NonDocumentTypeChildNode, ChildNode {\n open var data: String\n open val length: Int\n fun substringData(offset: Int, count: Int): String\n fun appendData(data: String)\n fun insertData(offset: Int, data: String)\n fun deleteData(offset: Int, count: Int)\n fun replaceData(offset: Int, count: Int, data: String)\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [Text](https://developer.mozilla.org/en/docs/Web/API/Text) to Kotlin\n */\npublic external open class Text(data: String = definedExternally) : CharacterData, Slotable, GeometryUtils {\n open val wholeText: String\n override val assignedSlot: HTMLSlotElement?\n override val previousElementSibling: Element?\n override val nextElementSibling: Element?\n fun splitText(offset: Int):

Text\n override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n override fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n override fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n override fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint\n override fun before(vararg nodes: dynamic)\n override fun after(vararg nodes:
dynamic)\n override fun replaceWith(vararg nodes: dynamic)\n override fun remove()\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [CDATASection](https://developer.mozilla.org/en/docs/Web/API/CDATASection) to Kotlin\n */\npublic external open class CDATASection : Text {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [ProcessingInstruction](https://developer.mozilla.org/en/docs/Web/API/ProcessingInstruction) to Kotlin\n */\npublic external abstract
class ProcessingInstruction : CharacterData, LinkStyle, UnionElementOrProcessingInstruction {\n open val target: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [Comment](https://developer.mozilla.org/en/docs/Web/API/Comment)
to Kotlin\n */\npublic external open class Comment(data: String = definedExternally) : CharacterData {\n override val previousElementSibling: Element?\n override val nextElementSibling: Element?\n override fun before(vararg nodes: dynamic)\n override fun after(vararg nodes: dynamic)\n override fun replaceWith(vararg nodes: dynamic)\n override fun remove()\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val

PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [Range](https://developer.mozilla.org/en/docs/Web/API/Range) to Kotlin\n */\npublic external open class Range {\n open val startContainer: Node\n open val startOffset: Int\n open val endContainer: Node\n open val endOffset: Int\n open val collapsed: Boolean\n open val commonAncestorContainer: Node\n fun setStart(node: Node, offset: Int)\n fun setEnd(node: Node, offset: Int)\n fun setStartBefore(node: Node)\n fun setStartAfter(node: Node)\n fun setEndBefore(node: Node)\n fun setEndAfter(node: Node)\n fun collapse(toStart: Boolean = definedExternally)\n fun selectNode(node: Node)\n fun selectNodeContents(node: Node)\n fun compareBoundaryPoints(how: Short, sourceRange: Range): Short\n fun deleteContents()\n fun extractContents(): DocumentFragment\n fun cloneContents(): DocumentFragment\n fun insertNode(node: Node)\n fun surroundContents(newParent: Node)\n fun cloneRange(): Range\n fun detach()\n fun isPointInRange(node: Node, offset: Int): Boolean\n fun comparePoint(node: Node, offset: Int): Short\n fun intersectsNode(node: Node): Boolean\n fun getClientRects(): Array<DOMRect>\n fun getBoundingClientRect(): DOMRect\n fun createContextualFragment(fragment: String): DocumentFragment\n\n companion object {\n val START_TO_START: Short\n val START_TO_END: Short\n val END_TO_END: Short\n val END_TO_START: Short\n }\n}\n\n/**\n * Exposes the JavaScript [NodeIterator](https://developer.mozilla.org/en/docs/Web/API/NodeIterator) to Kotlin\n */\npublic external abstract class NodeIterator {\n open val root: Node\n open val referenceNode: Node\n open val pointerBeforeReferenceNode: Boolean\n open val whatToShow: Int\n open val filter: NodeFilter?\n fun nextNode(): Node?\n fun previousNode(): Node?\n fun detach()\n}\n\n/**\n * Exposes the JavaScript [TreeWalker](https://developer.mozilla.org/en/docs/Web/API/TreeWalker) to Kotlin\n */\npublic external abstract class TreeWalker {\n open val root: Node\n open val whatToShow: Int\n open val filter: NodeFilter?\n open var currentNode: Node\n fun parentNode(): Node?\n fun firstChild(): Node?\n fun lastChild(): Node?\n fun previousSibling(): Node?\n fun nextSibling(): Node?\n fun previousNode(): Node?\n fun nextNode(): Node?\n}\n\n/**\n * Exposes the JavaScript [NodeFilter](https://developer.mozilla.org/en/docs/Web/API/NodeFilter) to Kotlin\n */\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface NodeFilter {\n fun acceptNode(node: Node): Short\n\n companion object {\n val FILTER_ACCEPT: Short\n val FILTER_REJECT: Short\n val FILTER_SKIP: Short\n val SHOW_ALL: Int\n val SHOW_ELEMENT: Int\n val SHOW_ATTRIBUTE: Int\n val SHOW_TEXT: Int\n val SHOW_CDATA_SECTION: Int\n val SHOW_ENTITY_REFERENCE: Int\n val SHOW_ENTITY: Int\n val SHOW_PROCESSING_INSTRUCTION: Int\n val SHOW_COMMENT: Int\n val SHOW_DOCUMENT: Int\n val SHOW_DOCUMENT_TYPE: Int\n val SHOW_DOCUMENT_FRAGMENT: Int\n val SHOW_NOTATION: Int\n }\n}\n\n/**\n * Exposes the JavaScript [DOMTokenList](https://developer.mozilla.org/en/docs/Web/API/DOMTokenList) to Kotlin\n */\npublic external abstract class DOMTokenList : ItemArrayLike<String> {\n open var value: String\n fun contains(token: String): Boolean\n fun add(vararg tokens: String)\n fun remove(vararg tokens: String)\n fun toggle(token: String, force: Boolean = definedExternally): Boolean\n fun replace(token: String, newToken: String)\n fun supports(token: String): Boolean\n override fun item(index: Int): String?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",

\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun DOMTokenList.get(index: Int): String? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript [DOMPointReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMPointReadOnly) to Kotlin\n */\npublic external open class DOMPointReadOnly(x: Double, y: Double, z: Double, w: Double) {\n open val x: Double\n open val y: Double\n open val z: Double\n open val w: Double\n fun matrixTransform(matrix: DOMMatrixReadOnly): DOMPoint\n}\n\n/**\n * Exposes the JavaScript [DOMPoint](https://developer.mozilla.org/en/docs/Web/API/DOMPoint) to Kotlin\n */\npublic external open class DOMPoint : DOMPointReadOnly {\n constructor(point: DOMPointInit)\n constructor(x: Double = definedExternally, y: Double = definedExternally, z: Double = definedExternally, w: Double = definedExternally)\n override var x: Double\n override var y: Double\n override var z: Double\n override var w: Double\n}\n\n/**\n * Exposes the JavaScript [DOMPointInit](https://developer.mozilla.org/en/docs/Web/API/DOMPointInit) to Kotlin\n */\npublic external interface DOMPointInit {\n var x: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var y: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var z: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var w: Double? /* = 1.0 */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun DOMPointInit(x: Double? = 0.0, y: Double? = 0.0, z: Double? = 0.0, w: Double? = 1.0): DOMPointInit {\n val o = js(\"({})\")\n o[\"x\"] = x\n o[\"y\"] = y\n o[\"z\"] = z\n o[\"w\"] = w\n return o\n}\n\n/**\n * Exposes the JavaScript [DOMRect](https://developer.mozilla.org/en/docs/Web/API/DOMRect)
to Kotlin\n */\npublic external open class DOMRect(x: Double = definedExternally, y: Double = definedExternally, width: Double = definedExternally, height: Double = definedExternally) : DOMRectReadOnly {\n override var x: Double\n override var y: Double\n override var width: Double\n override var height: Double\n}\n\n/**\n * Exposes the JavaScript [DOMRectReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMRectReadOnly) to Kotlin\n */\npublic external open class DOMRectReadOnly(x: Double, y: Double, width: Double, height: Double) {\n open val x: Double\n open val y: Double\n open val width: Double\n open val height: Double\n open val top: Double\n open val right: Double\n open val bottom: Double\n open val left: Double\n}\n\npublic external interface DOMRectInit {\n var x: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var y: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var width: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n var height: Double? /* = 0.0 */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun DOMRectInit(x: Double? = 0.0, y: Double? = 0.0, width: Double? = 0.0, height: Double? = 0.0): DOMRectInit {\n val o = js(\"({})\")\n o[\"x\"] = x\n o[\"y\"] = y\n o[\"width\"] = width\n o[\"height\"] = height\n return o\n}\n\npublic external interface DOMRectList : ItemArrayLike<DOMRect> {\n override fun item(index: Int): DOMRect?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun DOMRectList.get(index: Int): DOMRect? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript [DOMQuad](https://developer.mozilla.org/en/docs/Web/API/DOMQuad)
to Kotlin\n */\npublic external open class DOMQuad {\n constructor(p1: DOMPointInit = definedExternally, p2: DOMPointInit = definedExternally, p3: DOMPointInit = definedExternally, p4: DOMPointInit = definedExternally)\n constructor(rect: DOMRectInit)\n open val p1: DOMPoint\n open val p2: DOMPoint\n open val p3: DOMPoint\n open val p4: DOMPoint\n open val bounds: DOMRectReadOnly\n}\n\n/**\n * Exposes the JavaScript [DOMMatrixReadOnly](https://developer.mozilla.org/en/docs/Web/API/DOMMatrixReadOnly) to Kotlin\n */\npublic external open class DOMMatrixReadOnly(numberSequence: Array<Double>) {\n open val a: Double\n

```
 open val b: Double\n    open val c: Double\n    open val d: Double\n    open val e: Double\n    open val f: Double\n
 open val m11: Double\n    open val m12: Double\n    open val m13: Double\n    open val m14: Double\n    open val
m21: Double\n    open val m22: Double\n    open val m23: Double\n    open val m24: Double\n    open val m31:
 Double\n    open val m32: Double\n    open val m33: Double\n    open val m34: Double\n    open val m41: Double\n
  open val m42: Double\n    open val m43: Double\n    open val m44: Double\n    open val is2D: Boolean\n    open
val isIdentity: Boolean\n    fun translate(tx: Double, ty: Double, tz: Double = definedExternally): DOMMatrix\n
fun scale(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally): DOMMatrix\n
  fun scale3d(scale: Double, originX: Double = definedExternally, originY: Double = definedExternally, originZ:
Double = definedExternally): DOMMatrix\n    fun scaleNonUniform(scaleX: Double, scaleY: Double =
definedExternally, scaleZ: Double = definedExternally, originX: Double = definedExternally, originY: Double =
definedExternally, originZ: Double = definedExternally): DOMMatrix\n    fun rotate(angle: Double, originX:
Double = definedExternally, originY: Double = definedExternally): DOMMatrix\n    fun rotateFromVector(x:
Double, y: Double):
 DOMMatrix\n    fun rotateAxisAngle(x: Double, y: Double, z: Double, angle: Double): DOMMatrix\n    fun
skewX(sx: Double): DOMMatrix\n    fun skewY(sy: Double): DOMMatrix\n    fun multiply(other: DOMMatrix):
DOMMatrix\n    fun flipX(): DOMMatrix\n    fun flipY(): DOMMatrix\n    fun inverse(): DOMMatrix\n    fun
transformPoint(point: DOMPointInit = definedExternally): DOMPoint\n    fun toFloat32Array(): Float32Array\n
fun toFloat64Array(): Float64Array\n}\n\n/**\n * Exposes the JavaScript
[DOMMatrix](https://developer.mozilla.org/en/docs/Web/API/DOMMatrix) to Kotlin\n */\npublic external open
class DOMMatrix() : DOMMatrixReadOnly {\n    constructor(transformList: String)\n    constructor(other:
DOMMatrixReadOnly)\n    constructor(array32: Float32Array)\n    constructor(array64: Float64Array)\n
constructor(numberSequence: Array<Double>)\n    override var a: Double\n    override var b: Double\n    override
var c: Double\n    override var d: Double\n    override var e: Double\n
    override var f: Double\n    override var m11: Double\n    override var m12: Double\n    override var m13:
Double\n    override var m14: Double\n    override var m21: Double\n    override var m22: Double\n    override var
m23: Double\n    override var m24: Double\n    override var m31: Double\n    override var m32: Double\n    override
var m33: Double\n    override var m34: Double\n    override var m41: Double\n    override var m42: Double\n
override var m43: Double\n    override var m44: Double\n    fun multiplySelf(other: DOMMatrix): DOMMatrix\n
fun preMultiplySelf(other: DOMMatrix): DOMMatrix\n    fun translateSelf(tx: Double, ty: Double, tz: Double =
definedExternally): DOMMatrix\n    fun scaleSelf(scale: Double, originX: Double = definedExternally, originY:
Double = definedExternally): DOMMatrix\n    fun scale3dSelf(scale: Double, originX: Double = definedExternally,
originY: Double = definedExternally, originZ: Double = definedExternally): DOMMatrix\n    fun
scaleNonUniformSelf(scaleX:
 Double, scaleY: Double = definedExternally, scaleZ: Double = definedExternally, originX: Double =
definedExternally, originY: Double = definedExternally, originZ: Double = definedExternally): DOMMatrix\n    fun
rotateSelf(angle: Double, originX: Double = definedExternally, originY: Double = definedExternally):
DOMMatrix\n    fun rotateFromVectorSelf(x: Double, y: Double): DOMMatrix\n    fun rotateAxisAngleSelf(x:
Double, y: Double, z: Double, angle: Double): DOMMatrix\n    fun skewXSelf(sx: Double): DOMMatrix\n    fun
skewYSelf(sy: Double): DOMMatrix\n    fun invertSelf(): DOMMatrix\n    fun setMatrixValue(transformList:
String): DOMMatrix\n}\n\npublic external interface ScrollOptions {\n    var behavior: ScrollBehavior? /* =
ScrollBehavior.AUTO */\n        get() = definedExternally\n        set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ScrollOptions(behavior:
ScrollBehavior?
 = ScrollBehavior.AUTO): ScrollOptions {\n    val o = js(\"({})\")\n    o[\"behavior\"] = behavior\n    return
o\n}\n\n/**\n * Exposes the JavaScript
[ScrollToOptions](https://developer.mozilla.org/en/docs/Web/API/ScrollToOptions) to Kotlin\n */\npublic external
interface ScrollToOptions : ScrollOptions {\n    var left: Double?\n        get() = definedExternally\n        set(value) =
```

definedExternally\n   var top: Double?\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ScrollToOptions(left: Double? = undefined, top: Double? = undefined, behavior: ScrollBehavior? = ScrollBehavior.AUTO): ScrollToOptions {\n val o = js(\"({})\")\n   o[\"left\"] = left\n   o[\"top\"] = top\n   o[\"behavior\"] = behavior\n   return o\n}\n\n/**\n * Exposes the JavaScript [MediaQueryList](https://developer.mozilla.org/en/docs/Web/API/MediaQueryList) to Kotlin\n */\npublic
 external abstract class MediaQueryList : EventTarget {\n   open val media: String\n   open val matches: Boolean\n   open var onchange: ((Event) -> dynamic)?\n   fun addListener(listener: EventListener?)\n   fun addListener(listener: ((Event) -> Unit)?)\n   fun removeListener(listener: EventListener?)\n   fun removeListener(listener: ((Event) -> Unit)?)\n}\n\n/**\n * Exposes the JavaScript [MediaQueryListEvent](https://developer.mozilla.org/en/docs/Web/API/MediaQueryListEvent) to Kotlin\n */\npublic external open class MediaQueryListEvent(type: String, eventInitDict: MediaQueryListEventInit = definedExternally) : Event {\n   open val media: String\n   open val matches: Boolean\n\n   companion object {\n   val NONE: Short\n   val CAPTURING_PHASE: Short\n   val AT_TARGET: Short\n   val BUBBLING_PHASE: Short\n   }\n}\n\npublic external interface MediaQueryListEventInit : EventInit {\n   var media: String? /* = \"\" */\n   get() = definedExternally\n
        set(value) = definedExternally\n   var matches: Boolean? /* = false */\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaQueryListEventInit(media: String? = \"\", matches: Boolean? = false, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): MediaQueryListEventInit {\n   val o = js(\"({})\")\n   o[\"media\"] = media\n   o[\"matches\"] = matches\n   o[\"bubbles\"] = bubbles\n   o[\"cancelable\"] = cancelable\n   o[\"composed\"] = composed\n   return o\n}\n\n/**\n * Exposes the JavaScript [Screen](https://developer.mozilla.org/en/docs/Web/API/Screen) to Kotlin\n */\npublic external abstract class Screen {\n   open val availWidth: Int\n   open val availHeight: Int\n   open val width: Int\n   open val height: Int\n   open val colorDepth: Int\n   open val pixelDepth: Int\n}\n\n/**\n * Exposes the
 JavaScript [CaretPosition](https://developer.mozilla.org/en/docs/Web/API/CaretPosition) to Kotlin\n */\npublic external abstract class CaretPosition {\n   open val offsetNode: Node\n   open val offset: Int\n   fun getClientRect(): DOMRect?\n}\n\npublic external interface ScrollIntoViewOptions : ScrollOptions {\n   var block: ScrollLogicalPosition? /* = ScrollLogicalPosition.CENTER */\n   get() = definedExternally\n   set(value) = definedExternally\n   var inline: ScrollLogicalPosition? /* = ScrollLogicalPosition.CENTER */\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ScrollIntoViewOptions(block: ScrollLogicalPosition? = ScrollLogicalPosition.CENTER, inline: ScrollLogicalPosition? = ScrollLogicalPosition.CENTER, behavior: ScrollBehavior? = ScrollBehavior.AUTO): ScrollIntoViewOptions {\n val o = js(\"({})\")\n   o[\"block\"] = block\n
    o[\"inline\"] = inline\n   o[\"behavior\"] = behavior\n   return o\n}\n\npublic external interface BoxQuadOptions {\n   var box: CSSBoxType? /* = CSSBoxType.BORDER */\n   get() = definedExternally\n   set(value) = definedExternally\n   var relativeTo: dynamic\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun BoxQuadOptions(box: CSSBoxType? = CSSBoxType.BORDER, relativeTo: dynamic = undefined): BoxQuadOptions {\n   val o = js(\"({})\")\n   o[\"box\"] = box\n   o[\"relativeTo\"] = relativeTo\n   return o\n}\n\npublic external interface ConvertCoordinateOptions {\n   var fromBox: CSSBoxType? /* = CSSBoxType.BORDER */\n   get() = definedExternally\n   set(value) = definedExternally\n   var toBox: CSSBoxType? /* = CSSBoxType.BORDER */\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",

\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ConvertCoordinateOptions(fromBox: CSSBoxType? = CSSBoxType.BORDER, toBox: CSSBoxType? = CSSBoxType.BORDER): ConvertCoordinateOptions {\n val o = js(\"({})\")\n o[\"fromBox\"] = fromBox\n o[\"toBox\"] = toBox\n return o\n}\n\n/**\n * Exposes the JavaScript [GeometryUtils](https://developer.mozilla.org/en/docs/Web/API/GeometryUtils) to Kotlin\n */\npublic external interface GeometryUtils {\n fun getBoxQuads(options: BoxQuadOptions = definedExternally): Array<DOMQuad>\n fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions = definedExternally): DOMQuad\n fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions = definedExternally): DOMQuad\n fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions = definedExternally): DOMPoint\n}\n\n/**\n * Exposes the JavaScript [Touch](https://developer.mozilla.org/en/docs/Web/API/Touch) to Kotlin\n */\npublic external abstract class Touch {\n open val identifier: Int\n open val target: EventTarget\n open val screenX: Int\n open val screenY: Int\n open val clientX: Int\n open val clientY: Int\n open val pageX: Int\n open val pageY: Int\n open val region: String?\n}\n\npublic external abstract class TouchList : ItemArrayLike<Touch> {\n override fun item(index: Int): Touch?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun TouchList.get(index: Int): Touch? = asDynamic()[index]\n\npublic external open class TouchEvent : UIEvent {\n open val touches: TouchList\n open val targetTouches: TouchList\n open val changedTouches: TouchList\n open val altKey: Boolean\n open val metaKey: Boolean\n open val ctrlKey: Boolean\n open val shiftKey: Boolean\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\n/**\n * Exposes the JavaScript [Image](https://developer.mozilla.org/en/docs/Web/API/Image) to Kotlin\n */\npublic external open class Image(width: Int = definedExternally, height: Int = definedExternally) : HTMLImageElement {\n override var onabort: ((Event) -> dynamic)?\n override var onblur: ((FocusEvent) -> dynamic)?\n override var oncancel: ((Event) -> dynamic)?\n override var oncanplay: ((Event) -> dynamic)?\n override var oncanplaythrough: ((Event) -> dynamic)?\n override var onchange: ((Event) -> dynamic)?\n override var onclick: ((MouseEvent) -> dynamic)?\n override var onclose: ((Event) -> dynamic)?\n override var oncontextmenu: ((MouseEvent) -> dynamic)?\n override var oncuechange: ((Event) -> dynamic)?\n override var ondblclick: ((MouseEvent) -> dynamic)?\n override var ondrag: ((DragEvent) -> dynamic)?\n override var ondragend: ((DragEvent) -> dynamic)?\n override var ondragenter: ((DragEvent) -> dynamic)?\n override var ondragexit: ((DragEvent) -> dynamic)?\n override var ondragleave: ((DragEvent) -> dynamic)?\n override var ondragover: ((DragEvent) -> dynamic)?\n override var ondragstart: ((DragEvent) -> dynamic)?\n override var ondrop: ((DragEvent) -> dynamic)?\n override var ondurationchange: ((Event) -> dynamic)?\n override var onemptied: ((Event) -> dynamic)?\n override var onended: ((Event) -> dynamic)?\n override var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?\n override var onfocus: ((FocusEvent) -> dynamic)?\n override var oninput: ((InputEvent) -> dynamic)?\n override var oninvalid: ((Event) -> dynamic)?\n override var onkeydown: ((KeyboardEvent) -> dynamic)?\n override var onkeypress: ((KeyboardEvent) -> dynamic)?\n override var onkeyup: ((KeyboardEvent) -> dynamic)?\n override var onload: ((Event) -> dynamic)?\n override var onloadeddata: ((Event) -> dynamic)?\n override var onloadedmetadata: ((Event) -> dynamic)?\n override var onloadend: ((Event) -> dynamic)?\n override var onloadstart: ((ProgressEvent) -> dynamic)?\n override var onmousedown: ((MouseEvent) -> dynamic)?\n override var onmouseenter: ((MouseEvent) -> dynamic)?\n override var onmouseleave: ((MouseEvent) -> dynamic)?\n override var onmousemove: ((MouseEvent) -> dynamic)?\n override var onmouseout: ((MouseEvent) -> dynamic)?\n override var onmouseover: ((MouseEvent) -> dynamic)?\n override var onmouseup: ((MouseEvent) -> dynamic)?\n override var onwheel: ((WheelEvent) -> dynamic)?\n override var onpause: ((Event) -> dynamic)?\n override var onplay: ((Event) -> dynamic)?\n override var onplaying: ((Event) -> dynamic)?\n override var onprogress: ((ProgressEvent) -> dynamic)?\n override var onratechange: ((Event) -> dynamic)?\n override var onreset: ((Event) -> dynamic)?\n

override var
onresize: ((Event) -> dynamic)?\n   override var onscroll: ((Event) -> dynamic)?\n   override var onseeked: ((Event) -> dynamic)?\n   override var onseeking: ((Event) -> dynamic)?\n   override var onselect: ((Event) -> dynamic)?\n   override var onshow: ((Event) -> dynamic)?\n   override var onstalled: ((Event) -> dynamic)?\n override var onsubmit: ((Event) -> dynamic)?\n   override var onsuspend: ((Event) -> dynamic)?\n   override var ontimeupdate: ((Event) -> dynamic)?\n   override var ontoggle: ((Event) -> dynamic)?\n   override var onvolumechange: ((Event) -> dynamic)?\n   override var onwaiting: ((Event) -> dynamic)?\n   override var ongotpointercapture: ((PointerEvent) -> dynamic)?\n   override var onlostpointercapture: ((PointerEvent) -> dynamic)?\n   override var onpointerdown: ((PointerEvent) -> dynamic)?\n   override var onpointermove: ((PointerEvent) -> dynamic)?\n   override var onpointerup: ((PointerEvent) -> dynamic)?\n   override var onpointercancel:
((PointerEvent) -> dynamic)?\n   override var onpointerover: ((PointerEvent) -> dynamic)?\n   override var onpointerout: ((PointerEvent) -> dynamic)?\n   override var onpointerenter: ((PointerEvent) -> dynamic)?\n override var onpointerleave: ((PointerEvent) -> dynamic)?\n   override var oncopy: ((ClipboardEvent) -> dynamic)?\n   override var oncut: ((ClipboardEvent) -> dynamic)?\n   override var onpaste: ((ClipboardEvent) -> dynamic)?\n   override var contentEditable: String\n   override val isContentEditable: Boolean\n   override val style: CSSStyleDeclaration\n   override val children: HTMLCollection\n   override val firstElementChild: Element?\n   override val lastElementChild: Element?\n   override val childElementCount: Int\n   override val previousElementSibling: Element?\n   override val nextElementSibling: Element?\n   override val assignedSlot: HTMLSlotElement?\n   override fun prepend(vararg nodes: dynamic)\n   override fun append(vararg nodes: dynamic)\n
    override fun querySelector(selectors: String): Element?\n   override fun querySelectorAll(selectors: String): NodeList\n   override fun before(vararg nodes: dynamic)\n   override fun after(vararg nodes: dynamic)\n   override fun replaceWith(vararg nodes: dynamic)\n   override fun remove()\n   override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n   override fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n   override fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n   override fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint\n\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n
    val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n   val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n   val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n   }\n}\n\npublic external open class Audio(src: String = definedExternally) : HTMLAudioElement {\n   override var onabort: ((Event) -> dynamic)?\n override var onblur: ((FocusEvent) -> dynamic)?\n   override var oncancel: ((Event) -> dynamic)?\n   override var oncanplay: ((Event) -> dynamic)?\n   override var oncanplaythrough: ((Event) -> dynamic)?\n   override var onchange: ((Event)
-> dynamic)?\n   override var onclick: ((MouseEvent) -> dynamic)?\n   override var onclose: ((Event) -> dynamic)?\n   override var oncontextmenu: ((MouseEvent) -> dynamic)?\n   override var oncuechange: ((Event) -> dynamic)?\n   override var ondblclick: ((MouseEvent) -> dynamic)?\n   override var ondrag: ((DragEvent) -> dynamic)?\n   override var ondragend: ((DragEvent) -> dynamic)?\n   override var ondragenter: ((DragEvent) -> dynamic)?\n   override var ondragexit: ((DragEvent) -> dynamic)?\n   override var ondragleave: ((DragEvent) ->

dynamic)?\n override var ondragover: ((DragEvent) -> dynamic)?\n override var ondragstart: ((DragEvent) -> dynamic)?\n override var ondrop: ((DragEvent) -> dynamic)?\n override var ondurationchange: ((Event) -> dynamic)?\n override var onemptied: ((Event) -> dynamic)?\n override var onended: ((Event) -> dynamic)?\n override var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?\n override var onfocus: ((FocusEvent) -> dynamic)?\n override var oninput: ((InputEvent) -> dynamic)?\n override var oninvalid: ((Event) -> dynamic)?\n override var onkeydown: ((KeyboardEvent) -> dynamic)?\n override var onkeypress: ((KeyboardEvent) -> dynamic)?\n override var onkeyup: ((KeyboardEvent) -> dynamic)?\n override var onload: ((Event) -> dynamic)?\n override var onloadeddata: ((Event) -> dynamic)?\n override var onloadedmetadata: ((Event) -> dynamic)?\n override var onloadend: ((Event) -> dynamic)?\n override var onloadstart: ((ProgressEvent) -> dynamic)?\n override var onmousedown: ((MouseEvent) -> dynamic)?\n override var onmouseenter: ((MouseEvent) -> dynamic)?\n override var onmouseleave: ((MouseEvent) -> dynamic)?\n override var onmousemove: ((MouseEvent) -> dynamic)?\n override var onmouseout: ((MouseEvent) -> dynamic)?\n override var onmouseover: ((MouseEvent) -> dynamic)?\n override var onmouseup: ((MouseEvent) -> dynamic)?\n override var onwheel: ((WheelEvent) -> dynamic)?\n override var onpause: ((Event) -> dynamic)?\n override var onplay: ((Event) -> dynamic)?\n override var onplaying: ((Event) -> dynamic)?\n override var onprogress: ((ProgressEvent) -> dynamic)?\n override var onratechange: ((Event) -> dynamic)?\n override var onreset: ((Event) -> dynamic)?\n override var onresize: ((Event) -> dynamic)?\n override var onscroll: ((Event) -> dynamic)?\n override var onseeked: ((Event) -> dynamic)?\n override var onseeking: ((Event) -> dynamic)?\n override var onselect: ((Event) -> dynamic)?\n override var onshow: ((Event) -> dynamic)?\n override var onstalled: ((Event) -> dynamic)?\n override var onsubmit: ((Event) -> dynamic)?\n override var onsuspend: ((Event) -> dynamic)?\n override var ontimeupdate: ((Event) -> dynamic)?\n override var ontoggle: ((Event) -> dynamic)?\n override var onvolumechange: ((Event) -> dynamic)?\n override var onwaiting: ((Event) -> dynamic)?\n override var ongotpointercapture: ((PointerEvent) -> dynamic)?\n override var onlostpointercapture: ((PointerEvent) -> dynamic)?\n override var onpointerdown: ((PointerEvent) -> dynamic)?\n override var onpointermove: ((PointerEvent) -> dynamic)?\n override var onpointerup: ((PointerEvent) -> dynamic)?\n override var onpointercancel: ((PointerEvent) -> dynamic)?\n override var onpointerover: ((PointerEvent) -> dynamic)?\n override var onpointerout: ((PointerEvent) -> dynamic)?\n override var onpointerenter: ((PointerEvent) -> dynamic)?\n override var onpointerleave: ((PointerEvent) -> dynamic)?\n override var oncopy: ((ClipboardEvent) -> dynamic)?\n override var oncut: ((ClipboardEvent) -> dynamic)?\n override var onpaste: ((ClipboardEvent) -> dynamic)?\n override var contentEditable: String\n override val isContentEditable: Boolean\n override val style: CSSStyleDeclaration\n override val children: HTMLCollection\n override val firstElementChild: Element?\n override val lastElementChild: Element?\n override val childElementCount: Int\n override val previousElementSibling: Element?\n override val nextElementSibling: Element?\n override val assignedSlot: HTMLSlotElement?\n override fun prepend(vararg nodes: dynamic)\n override fun append(vararg nodes: dynamic)\n override fun querySelector(selectors: String): Element?\n override fun querySelectorAll(selectors: String): NodeList\n override fun before(vararg nodes: dynamic)\n override fun after(vararg nodes: dynamic)\n override fun replaceWith(vararg nodes: dynamic)\n override fun remove()\n override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n override fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n override fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n override fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint\n\n companion object {\n val NETWORK_EMPTY: Short\n val NETWORK_IDLE: Short\n val NETWORK_LOADING: Short\n val NETWORK_NO_SOURCE: Short\n val HAVE_NOTHING: Short\n val HAVE_METADATA: Short\n val HAVE_CURRENT_DATA: Short\n val HAVE_FUTURE_DATA: Short\n val

HAVE_ENOUGH_DATA: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED:
Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [Option](https://developer.mozilla.org/en/docs/Web/API/Option) to Kotlin\n */\npublic external open class Option(text: String = definedExternally, value: String = definedExternally, defaultSelected: Boolean = definedExternally, selected: Boolean = definedExternally) : HTMLOptionElement {\n override var onabort: ((Event) -> dynamic)?\n override var onblur: ((FocusEvent) -> dynamic)?\n override var oncancel: ((Event) -> dynamic)?\n override var oncanplay: ((Event) -> dynamic)?\n override var oncanplaythrough: ((Event) -> dynamic)?\n override var onchange: ((Event) -> dynamic)?\n override var onclick: ((MouseEvent) -> dynamic)?\n override var onclose:
((Event) -> dynamic)?\n override var oncontextmenu: ((MouseEvent) -> dynamic)?\n override var oncuechange: ((Event) -> dynamic)?\n override var ondblclick: ((MouseEvent) -> dynamic)?\n override var ondrag: ((DragEvent) -> dynamic)?\n override var ondragend: ((DragEvent) -> dynamic)?\n override var ondragenter: ((DragEvent) -> dynamic)?\n override var ondragexit: ((DragEvent) -> dynamic)?\n override var ondragleave: ((DragEvent) -> dynamic)?\n override var ondragover: ((DragEvent) -> dynamic)?\n override var ondragstart: ((DragEvent) -> dynamic)?\n override var ondrop: ((DragEvent) -> dynamic)?\n override var ondurationchange: ((Event) -> dynamic)?\n override var onemptied: ((Event) -> dynamic)?\n override var onended: ((Event) -> dynamic)?\n override var onerror: ((dynamic, String, Int, Int, Any?) -> dynamic)?\n override var onfocus: ((FocusEvent) -> dynamic)?\n override var oninput: ((InputEvent) -> dynamic)?\n override var oninvalid:
((Event) -> dynamic)?\n override var onkeydown: ((KeyboardEvent) -> dynamic)?\n override var onkeypress: ((KeyboardEvent) -> dynamic)?\n override var onkeyup: ((KeyboardEvent) -> dynamic)?\n override var onload: ((Event) -> dynamic)?\n override var onloadeddata: ((Event) -> dynamic)?\n override var onloadedmetadata: ((Event) -> dynamic)?\n override var onloadend: ((Event) -> dynamic)?\n override var onloadstart: ((ProgressEvent) -> dynamic)?\n override var onmousedown: ((MouseEvent) -> dynamic)?\n override var onmouseenter: ((MouseEvent) -> dynamic)?\n override var onmouseleave: ((MouseEvent) -> dynamic)?\n override var onmousemove: ((MouseEvent) -> dynamic)?\n override var onmouseout: ((MouseEvent) -> dynamic)?\n override var onmouseover: ((MouseEvent) -> dynamic)?\n override var onmouseup: ((MouseEvent) -> dynamic)?\n override var onwheel: ((WheelEvent) -> dynamic)?\n override var onpause: ((Event) -> dynamic)?\n override
var onplay: ((Event) -> dynamic)?\n override var onplaying: ((Event) -> dynamic)?\n override var onprogress: ((ProgressEvent) -> dynamic)?\n override var onratechange: ((Event) -> dynamic)?\n override var onreset: ((Event) -> dynamic)?\n override var onresize: ((Event) -> dynamic)?\n override var onscroll: ((Event) -> dynamic)?\n override var onseeked: ((Event) -> dynamic)?\n override var onseeking: ((Event) -> dynamic)?\n override var onselect: ((Event) -> dynamic)?\n override var onshow: ((Event) -> dynamic)?\n override var onstalled: ((Event) -> dynamic)?\n override var onsubmit: ((Event) -> dynamic)?\n override var onsuspend: ((Event) -> dynamic)?\n override var ontimeupdate: ((Event) -> dynamic)?\n override var ontoggle: ((Event) -> dynamic)?\n override var onvolumechange: ((Event) -> dynamic)?\n override var onwaiting: ((Event) -> dynamic)?\n override var ongotpointercapture: ((PointerEvent) -> dynamic)?\n override var onlostpointercapture:

((PointerEvent) -> dynamic)?\n   override var onpointerdown: ((PointerEvent) -> dynamic)?\n   override var onpointermove: ((PointerEvent) -> dynamic)?\n   override var onpointerup: ((PointerEvent) -> dynamic)?\n override var onpointercancel: ((PointerEvent) -> dynamic)?\n   override var onpointerover: ((PointerEvent) -> dynamic)?\n   override var onpointerout: ((PointerEvent) -> dynamic)?\n   override var onpointerenter: ((PointerEvent) -> dynamic)?\n   override var onpointerleave: ((PointerEvent) -> dynamic)?\n   override var oncopy: ((ClipboardEvent) -> dynamic)?\n   override var oncut: ((ClipboardEvent) -> dynamic)?\n   override var onpaste: ((ClipboardEvent) -> dynamic)?\n   override var contentEditable: String\n   override val isContentEditable: Boolean\n   override val style: CSSStyleDeclaration\n   override val children: HTMLCollection\n   override val firstElementChild: Element?\n   override val lastElementChild: Element?\n override val childElementCount: Int\n   override val previousElementSibling: Element?\n   override val nextElementSibling: Element?\n   override val assignedSlot: HTMLSlotElement?\n   override fun prepend(vararg nodes: dynamic)\n override fun append(vararg nodes: dynamic)\n   override fun querySelector(selectors: String): Element?\n override fun querySelectorAll(selectors: String): NodeList\n   override fun before(vararg nodes: dynamic)\n override fun after(vararg nodes: dynamic)\n   override fun replaceWith(vararg nodes: dynamic)\n   override fun remove()\n   override fun getBoxQuads(options: BoxQuadOptions /* = definedExternally */): Array<DOMQuad>\n   override fun convertQuadFromNode(quad: dynamic, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n   override fun convertRectFromNode(rect: DOMRectReadOnly, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMQuad\n   override fun convertPointFromNode(point: DOMPointInit, from: dynamic, options: ConvertCoordinateOptions /* = definedExternally */): DOMPoint\n\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n   val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n   val DOCUMENT_POSITION_FOLLOWING: Short\n   val DOCUMENT_POSITION_CONTAINS: Short\n   val DOCUMENT_POSITION_CONTAINED_BY: Short\n   val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n   }\n}\n\npublic external interface UnionElementOrHTMLCollection\n\npublic external interface UnionElementOrRadioNodeList\n\npublic external interface UnionHTMLOptGroupElementOrHTMLOptionElement\n\npublic external interface UnionAudioTrackOrTextTrackOrVideoTrack\n\npublic external interface UnionElementOrMouseEvent\n\npublic external interface UnionMessagePortOrWindowProxy\n\npublic external interface MediaProvider\n\npublic external interface RenderingContext\n\npublic external interface HTMLOrSVGImageElement : CanvasImageSource\n\npublic external interface CanvasImageSource : ImageBitmapSource\n\npublic external interface ImageBitmapSource\n\npublic external interface HTMLOrSVGScriptElement\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface DocumentReadyState {\n   companion object\n}\n\npublic inline val DocumentReadyState.Companion.LOADING: DocumentReadyState get() = \"loading\".asDynamic().unsafeCast<DocumentReadyState>()\n\npublic inline val DocumentReadyState.Companion.INTERACTIVE: DocumentReadyState get() = \"interactive\".asDynamic().unsafeCast<DocumentReadyState>()\n\npublic inline val DocumentReadyState.Companion.COMPLETE: DocumentReadyState get() = \"complete\".asDynamic().unsafeCast<DocumentReadyState>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface CanPlayTypeResult {\n   companion object\n}\n\npublic inline val

```
CanPlayTypeResult.Companion.EMPTY: CanPlayTypeResult get() =
\"\".asDynamic().unsafeCast<CanPlayTypeResult>()\n\npublic inline val CanPlayTypeResult.Companion.MAYBE:
CanPlayTypeResult get() = \"maybe\".asDynamic().unsafeCast<CanPlayTypeResult>()\n\npublic inline val
CanPlayTypeResult.Companion.PROBABLY: CanPlayTypeResult get() =
\"probably\".asDynamic().unsafeCast<CanPlayTypeResult>()\n\n/* please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface TextTrackMode {\n    companion object\n}\n\npublic
 inline val TextTrackMode.Companion.DISABLED: TextTrackMode get() =
\"disabled\".asDynamic().unsafeCast<TextTrackMode>()\n\npublic inline val
TextTrackMode.Companion.HIDDEN: TextTrackMode get() =
\"hidden\".asDynamic().unsafeCast<TextTrackMode>()\n\npublic inline val
TextTrackMode.Companion.SHOWING: TextTrackMode get() =
\"showing\".asDynamic().unsafeCast<TextTrackMode>()\n\n/* please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface TextTrackKind {\n    companion object\n}\n\npublic inline val TextTrackKind.Companion.SUBTITLES:
TextTrackKind get() = \"subtitles\".asDynamic().unsafeCast<TextTrackKind>()\n\npublic inline val
TextTrackKind.Companion.CAPTIONS: TextTrackKind get() =
\"captions\".asDynamic().unsafeCast<TextTrackKind>()\n\npublic inline val
TextTrackKind.Companion.DESCRIPTIONS: TextTrackKind get() =
\"descriptions\".asDynamic().unsafeCast<TextTrackKind>()\n\npublic inline val
TextTrackKind.Companion.CHAPTERS:
 TextTrackKind get() = \"chapters\".asDynamic().unsafeCast<TextTrackKind>()\n\npublic inline val
TextTrackKind.Companion.METADATA: TextTrackKind get() =
\"metadata\".asDynamic().unsafeCast<TextTrackKind>()\n\n/* please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface SelectionMode {\n    companion object\n}\n\npublic inline val SelectionMode.Companion.SELECT:
SelectionMode get() = \"select\".asDynamic().unsafeCast<SelectionMode>()\n\npublic inline val
SelectionMode.Companion.START: SelectionMode get() =
\"start\".asDynamic().unsafeCast<SelectionMode>()\n\npublic inline val SelectionMode.Companion.END:
SelectionMode get() = \"end\".asDynamic().unsafeCast<SelectionMode>()\n\npublic inline val
SelectionMode.Companion.PRESERVE: SelectionMode get() =
\"preserve\".asDynamic().unsafeCast<SelectionMode>()\n\n/* please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic
 external interface CanvasFillRule {\n    companion object\n}\n\npublic inline val
CanvasFillRule.Companion.NONZERO: CanvasFillRule get() =
\"nonzero\".asDynamic().unsafeCast<CanvasFillRule>()\n\npublic inline val
CanvasFillRule.Companion.EVENODD: CanvasFillRule get() =
\"evenodd\".asDynamic().unsafeCast<CanvasFillRule>()\n\n/* please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface ImageSmoothingQuality {\n    companion object\n}\n\npublic inline val
ImageSmoothingQuality.Companion.LOW: ImageSmoothingQuality get() =
\"low\".asDynamic().unsafeCast<ImageSmoothingQuality>()\n\npublic inline val
ImageSmoothingQuality.Companion.MEDIUM: ImageSmoothingQuality get() =
\"medium\".asDynamic().unsafeCast<ImageSmoothingQuality>()\n\npublic inline val
ImageSmoothingQuality.Companion.HIGH: ImageSmoothingQuality get() =
\"high\".asDynamic().unsafeCast<ImageSmoothingQuality>()\n\n/*
 please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
```

```
interface CanvasLineCap {\n    companion object\n}\n\npublic inline val CanvasLineCap.Companion.BUTT:
CanvasLineCap get() = \"butt\".asDynamic().unsafeCast<CanvasLineCap>()\n\npublic inline val
CanvasLineCap.Companion.ROUND: CanvasLineCap get() =
\"round\".asDynamic().unsafeCast<CanvasLineCap>()\n\npublic inline val CanvasLineCap.Companion.SQUARE:
CanvasLineCap get() = \"square\".asDynamic().unsafeCast<CanvasLineCap>()\n\n/* please, don't implement this
interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic
external interface CanvasLineJoin {\n    companion object\n}\n\npublic inline val
CanvasLineJoin.Companion.ROUND: CanvasLineJoin get() =
\"round\".asDynamic().unsafeCast<CanvasLineJoin>()\n\npublic inline val CanvasLineJoin.Companion.BEVEL:
CanvasLineJoin get() = \"bevel\".asDynamic().unsafeCast<CanvasLineJoin>()\n\npublic
 inline val CanvasLineJoin.Companion.MITER: CanvasLineJoin get() =
\"miter\".asDynamic().unsafeCast<CanvasLineJoin>()\n\n/* please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface CanvasTextAlign {\n    companion object\n}\n\npublic inline val CanvasTextAlign.Companion.START:
CanvasTextAlign get() = \"start\".asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val
CanvasTextAlign.Companion.END: CanvasTextAlign get() =
\"end\".asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val CanvasTextAlign.Companion.LEFT:
CanvasTextAlign get() = \"left\".asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val
CanvasTextAlign.Companion.RIGHT: CanvasTextAlign get() =
\"right\".asDynamic().unsafeCast<CanvasTextAlign>()\n\npublic inline val
CanvasTextAlign.Companion.CENTER: CanvasTextAlign get() =
\"center\".asDynamic().unsafeCast<CanvasTextAlign>()\n\n/* please, don't implement this interface!
 */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface CanvasTextBaseline {\n    companion object\n}\n\npublic inline val CanvasTextBaseline.Companion.TOP:
CanvasTextBaseline get() = \"top\".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.HANGING: CanvasTextBaseline get() =
\"hanging\".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.MIDDLE: CanvasTextBaseline get() =
\"middle\".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.ALPHABETIC: CanvasTextBaseline get() =
\"alphabetic\".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.IDEOGRAPHIC: CanvasTextBaseline get() =
\"ideographic\".asDynamic().unsafeCast<CanvasTextBaseline>()\n\npublic inline val
CanvasTextBaseline.Companion.BOTTOM: CanvasTextBaseline get() =
\"bottom\".asDynamic().unsafeCast<CanvasTextBaseline>()\n\n/*
 please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface CanvasDirection {\n    companion object\n}\n\npublic inline val CanvasDirection.Companion.LTR:
CanvasDirection get() = \"ltr\".asDynamic().unsafeCast<CanvasDirection>()\n\npublic inline val
CanvasDirection.Companion.RTL: CanvasDirection get() =
\"rtl\".asDynamic().unsafeCast<CanvasDirection>()\n\npublic inline val CanvasDirection.Companion.INHERIT:
CanvasDirection get() = \"inherit\".asDynamic().unsafeCast<CanvasDirection>()\n\n/* please, don't implement this
interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic
external interface ScrollRestoration {\n    companion object\n}\n\npublic inline val
ScrollRestoration.Companion.AUTO: ScrollRestoration get() =
\"auto\".asDynamic().unsafeCast<ScrollRestoration>()\n\npublic inline val
ScrollRestoration.Companion.MANUAL: ScrollRestoration get() =
\"manual\".asDynamic().unsafeCast<ScrollRestoration>()\n\n/*
```

please, don't implement this interface!
/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface ImageOrientation {\n   companion object\n}\n\npublic inline val ImageOrientation.Companion.NONE: ImageOrientation get() = \"none\".asDynamic().unsafeCast<ImageOrientation>()\n\npublic inline val ImageOrientation.Companion.FLIPY: ImageOrientation get() = \"flipY\".asDynamic().unsafeCast<ImageOrientation>()\n\n/ please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface PremultiplyAlpha {\n   companion object\n}\n\npublic inline val PremultiplyAlpha.Companion.NONE: PremultiplyAlpha get() = \"none\".asDynamic().unsafeCast<PremultiplyAlpha>()\n\npublic inline val PremultiplyAlpha.Companion.PREMULTIPLY: PremultiplyAlpha get() = \"premultiply\".asDynamic().unsafeCast<PremultiplyAlpha>()\n\npublic inline val PremultiplyAlpha.Companion.DEFAULT:
PremultiplyAlpha get() = \"default\".asDynamic().unsafeCast<PremultiplyAlpha>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface ColorSpaceConversion {\n   companion object\n}\n\npublic inline val ColorSpaceConversion.Companion.NONE: ColorSpaceConversion get() = \"none\".asDynamic().unsafeCast<ColorSpaceConversion>()\n\npublic inline val ColorSpaceConversion.Companion.DEFAULT: ColorSpaceConversion get() = \"default\".asDynamic().unsafeCast<ColorSpaceConversion>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface ResizeQuality {\n   companion object\n}\n\npublic inline val ResizeQuality.Companion.PIXELATED: ResizeQuality get() = \"pixelated\".asDynamic().unsafeCast<ResizeQuality>()\n\npublic inline val ResizeQuality.Companion.LOW: ResizeQuality get() = \"low\".asDynamic().unsafeCast<ResizeQuality>()\n\npublic
inline val ResizeQuality.Companion.MEDIUM: ResizeQuality get() = \"medium\".asDynamic().unsafeCast<ResizeQuality>()\n\npublic inline val ResizeQuality.Companion.HIGH: ResizeQuality get() = \"high\".asDynamic().unsafeCast<ResizeQuality>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface BinaryType {\n   companion object\n}\n\npublic inline val BinaryType.Companion.BLOB: BinaryType get() = \"blob\".asDynamic().unsafeCast<BinaryType>()\n\npublic inline val BinaryType.Companion.ARRAYBUFFER: BinaryType get() = \"arraybuffer\".asDynamic().unsafeCast<BinaryType>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface WorkerType {\n   companion object\n}\n\npublic inline val WorkerType.Companion.CLASSIC: WorkerType get() = \"classic\".asDynamic().unsafeCast<WorkerType>()\n\npublic
inline val WorkerType.Companion.MODULE: WorkerType get() = \"module\".asDynamic().unsafeCast<WorkerType>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface ShadowRootMode {\n   companion object\n}\n\npublic inline val ShadowRootMode.Companion.OPEN: ShadowRootMode get() = \"open\".asDynamic().unsafeCast<ShadowRootMode>()\n\npublic inline val ShadowRootMode.Companion.CLOSED: ShadowRootMode get() = \"closed\".asDynamic().unsafeCast<ShadowRootMode>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface ScrollBehavior {\n   companion object\n}\n\npublic inline val ScrollBehavior.Companion.AUTO: ScrollBehavior get() = \"auto\".asDynamic().unsafeCast<ScrollBehavior>()\n\npublic inline val ScrollBehavior.Companion.INSTANT: ScrollBehavior get() = \"instant\".asDynamic().unsafeCast<ScrollBehavior>()\n\npublic

inline val ScrollBehavior.Companion.SMOOTH: ScrollBehavior get() = \"smooth\".asDynamic().unsafeCast<ScrollBehavior>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface ScrollLogicalPosition {\n   companion object\n}\n\npublic inline val ScrollLogicalPosition.Companion.START: ScrollLogicalPosition get() = \"start\".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\npublic inline val ScrollLogicalPosition.Companion.CENTER: ScrollLogicalPosition get() = \"center\".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\npublic inline val ScrollLogicalPosition.Companion.END: ScrollLogicalPosition get() = \"end\".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\npublic inline val ScrollLogicalPosition.Companion.NEAREST: ScrollLogicalPosition get() = \"nearest\".asDynamic().unsafeCast<ScrollLogicalPosition>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface CSSBoxType {\n   companion object\n}\n\npublic inline val CSSBoxType.Companion.MARGIN: CSSBoxType get() = \"margin\".asDynamic().unsafeCast<CSSBoxType>()\n\npublic inline val CSSBoxType.Companion.BORDER: CSSBoxType get() = \"border\".asDynamic().unsafeCast<CSSBoxType>()\n\npublic inline val CSSBoxType.Companion.PADDING: CSSBoxType get() = \"padding\".asDynamic().unsafeCast<CSSBoxType>()\n\npublic inline val CSSBoxType.Companion.CONTENT: CSSBoxType get() = \"content\".asDynamic().unsafeCast<CSSBoxType>()","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.fetch\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.files.*\nimport org.w3c.xhr.*\n\n/**\n * Exposes the JavaScript [Headers](https://developer.mozilla.org/en/docs/Web/API/Headers) to Kotlin\n */\npublic external open class Headers(init: dynamic = definedExternally) {\n   fun append(name: String, value: String)\n   fun delete(name: String)\n   fun get(name: String): String?\n   fun has(name: String): Boolean\n   fun set(name: String, value: String)\n}\n\n/**\n * Exposes the JavaScript [Body](https://developer.mozilla.org/en/docs/Web/API/Body) to Kotlin\n */\npublic external interface Body {\n   val bodyUsed: Boolean\n   fun arrayBuffer(): Promise<ArrayBuffer>\n   fun blob(): Promise<Blob>\n   fun formData(): Promise<FormData>\n   fun json(): Promise<Any?>\n   fun text(): Promise<String>\n}\n\n/**\n * Exposes the JavaScript [Request](https://developer.mozilla.org/en/docs/Web/API/Request) to Kotlin\n */\npublic external open class Request(input: dynamic, init: RequestInit = definedExternally) : Body {\n   open val method: String\n   open val url: String\n   open val headers: Headers\n   open val type: RequestType\n   open val destination: RequestDestination\n   open val referrer: String\n   open val referrerPolicy: dynamic\n   open val mode: RequestMode\n   open val credentials: RequestCredentials\n   open val cache: RequestCache\n   open val redirect: RequestRedirect\n   open val integrity: String\n   open val keepalive: Boolean\n   override val bodyUsed: Boolean\n   fun clone(): Request\n   override fun arrayBuffer(): Promise<ArrayBuffer>\n   override fun blob(): Promise<Blob>\n   override fun formData(): Promise<FormData>\n   override fun json(): Promise<Any?>\n   override fun text(): Promise<String>\n}\n\npublic external interface RequestInit {\n   var method: String?\n   get() = definedExternally\n   set(value) = definedExternally\n   var headers: dynamic\n   get() = definedExternally\n   set(value) = definedExternally\n   var body: dynamic\n   get() = definedExternally\n   set(value) = definedExternally\n   var referrer: String?\n   get() = definedExternally\n   set(value) = definedExternally\n   var referrerPolicy: dynamic\n   get() = definedExternally\n   set(value) = definedExternally\n   var mode: RequestMode?\n   get() = definedExternally\n   set(value) = definedExternally\n   var credentials: RequestCredentials?\n   get() = definedExternally\n   set(value) = definedExternally\n   var cache: RequestCache?\n   get() = definedExternally\n   set(value) =

definedExternally\n   var redirect: RequestRedirect?\n   get() = definedExternally\n   set(value) = definedExternally\n   var integrity: String?\n   get() = definedExternally\n   set(value) = definedExternally\n   var keepalive: Boolean?\n   get() = definedExternally\n   set(value) = definedExternally\n   var window: Any?\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun RequestInit(method: String? = undefined, headers: dynamic = undefined, body: dynamic = undefined, referrer: String? = undefined, referrerPolicy: dynamic = undefined, mode: RequestMode? = undefined, credentials: RequestCredentials? = undefined, cache: RequestCache? = undefined, redirect: RequestRedirect? = undefined, integrity: String? = undefined, keepalive: Boolean? = undefined, window: Any? = undefined): RequestInit {\n   val o = js(\"({})\")\n   o[\"method\"] = method\n   o[\"headers\"] = headers\n   o[\"body\"] = body\n   o[\"referrer\"] = referrer\n   o[\"referrerPolicy\"] = referrerPolicy\n   o[\"mode\"] = mode\n   o[\"credentials\"] = credentials\n   o[\"cache\"] = cache\n   o[\"redirect\"] = redirect\n   o[\"integrity\"] = integrity\n   o[\"keepalive\"] = keepalive\n   o[\"window\"] = window\n   return o\n}\n\n/**\n * Exposes the JavaScript [Response](https://developer.mozilla.org/en/docs/Web/API/Response) to Kotlin\n */\npublic external open class Response(body: dynamic = definedExternally, init: ResponseInit = definedExternally) : Body {\n   open val type: ResponseType\n   open val url: String\n   open val redirected: Boolean\n   open val status: Short\n   open val ok: Boolean\n   open val statusText: String\n   open val headers: Headers\n   open val body: dynamic\n   open val trailer: Promise<Headers>\n   override val bodyUsed: Boolean\n   fun clone(): Response\n   override fun arrayBuffer(): Promise<ArrayBuffer>\n   override fun blob(): Promise<Blob>\n   override fun formData(): Promise<FormData>\n   override fun json(): Promise<Any?>\n   override fun text(): Promise<String>\n\n   companion object {\n   fun error(): Response\n   fun redirect(url: String, status: Short = definedExternally): Response\n   }\n}\n\npublic external interface ResponseInit {\n   var status: Short? /* = 200 */\n   get() = definedExternally\n   set(value) = definedExternally\n   var statusText: String? /* = \"OK\" */\n   get() = definedExternally\n   set(value) = definedExternally\n   var headers: dynamic\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ResponseInit(status: Short? = 200, statusText: String? = \"OK\", headers: dynamic = undefined): ResponseInit {\n   val o = js(\"({})\")\n   o[\"status\"] = status\n   o[\"statusText\"] = statusText\n   o[\"headers\"] = headers\n   return o\n}\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface RequestType {\n   companion object\n}\n\npublic inline val RequestType.Companion.EMPTY: RequestType get() = \"\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val RequestType.Companion.AUDIO: RequestType get() = \"audio\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val RequestType.Companion.FONT: RequestType get() = \"font\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val RequestType.Companion.IMAGE: RequestType get() = \"image\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val RequestType.Companion.SCRIPT: RequestType get() = \"script\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val RequestType.Companion.STYLE: RequestType get() = \"style\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val RequestType.Companion.TRACK: RequestType get() = \"track\".asDynamic().unsafeCast<RequestType>()\n\npublic inline val RequestType.Companion.VIDEO: RequestType get() = \"video\".asDynamic().unsafeCast<RequestType>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface RequestDestination {\n   companion object\n}\n\npublic inline val RequestDestination.Companion.EMPTY: RequestDestination get() = \"\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val RequestDestination.Companion.DOCUMENT: RequestDestination get() =

\"document\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.EMBED: RequestDestination get() =
\"embed\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.FONT: RequestDestination get() =
\"font\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.IMAGE: RequestDestination get() =
\"image\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.MANIFEST: RequestDestination get() =
\"manifest\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.MEDIA: RequestDestination get() =
\"media\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.OBJECT: RequestDestination get()
= \"object\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.REPORT: RequestDestination get() =
\"report\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.SCRIPT: RequestDestination get() =
\"script\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.SERVICEWORKER: RequestDestination get() =
\"serviceworker\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.SHAREDWORKER: RequestDestination get() =
\"sharedworker\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.STYLE: RequestDestination get() =
\"style\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.WORKER: RequestDestination get() =
\"worker\".asDynamic().unsafeCast<RequestDestination>()\n\npublic inline val
RequestDestination.Companion.XSLT: RequestDestination
get() = \"xslt\".asDynamic().unsafeCast<RequestDestination>()\n\n/* please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface RequestMode {\n companion object\n}\n\npublic inline val RequestMode.Companion.NAVIGATE:
RequestMode get() = \"navigate\".asDynamic().unsafeCast<RequestMode>()\n\npublic inline val
RequestMode.Companion.SAME_ORIGIN: RequestMode get() = \"same-
origin\".asDynamic().unsafeCast<RequestMode>()\n\npublic inline val RequestMode.Companion.NO_CORS:
RequestMode get() = \"no-cors\".asDynamic().unsafeCast<RequestMode>()\n\npublic inline val
RequestMode.Companion.CORS: RequestMode get() = \"cors\".asDynamic().unsafeCast<RequestMode>()\n\n/*
please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface RequestCredentials {\n companion object\n}\n\npublic inline val RequestCredentials.Companion.OMIT:
RequestCredentials
get() = \"omit\".asDynamic().unsafeCast<RequestCredentials>()\n\npublic inline val
RequestCredentials.Companion.SAME_ORIGIN: RequestCredentials get() = \"same-
origin\".asDynamic().unsafeCast<RequestCredentials>()\n\npublic inline val
RequestCredentials.Companion.INCLUDE: RequestCredentials get() =
\"include\".asDynamic().unsafeCast<RequestCredentials>()\n\n/* please, don't implement this interface!
*/\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external
interface RequestCache {\n companion object\n}\n\npublic inline val RequestCache.Companion.DEFAULT:
RequestCache get() = \"default\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val
RequestCache.Companion.NO_STORE: RequestCache get() = \"no-
store\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val RequestCache.Companion.RELOAD:

RequestCache get() = \"reload\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val RequestCache.Companion.NO_CACHE: RequestCache get() = \"no-cache\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val RequestCache.Companion.FORCE_CACHE: RequestCache get() = \"force-cache\".asDynamic().unsafeCast<RequestCache>()\n\npublic inline val RequestCache.Companion.ONLY_IF_CACHED: RequestCache get() = \"only-if-cached\".asDynamic().unsafeCast<RequestCache>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface RequestRedirect {\n   companion object\n}\n\npublic inline val RequestRedirect.Companion.FOLLOW: RequestRedirect get() = \"follow\".asDynamic().unsafeCast<RequestRedirect>()\n\npublic inline val RequestRedirect.Companion.ERROR: RequestRedirect get() = \"error\".asDynamic().unsafeCast<RequestRedirect>()\n\npublic inline val RequestRedirect.Companion.MANUAL: RequestRedirect get() = \"manual\".asDynamic().unsafeCast<RequestRedirect>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface ResponseType {\n   companion object\n}\n\npublic inline val ResponseType.Companion.BASIC: ResponseType get() = \"basic\".asDynamic().unsafeCast<ResponseType>()\n\npublic inline val ResponseType.Companion.CORS: ResponseType get() = \"cors\".asDynamic().unsafeCast<ResponseType>()\n\npublic inline val ResponseType.Companion.DEFAULT: ResponseType get() = \"default\".asDynamic().unsafeCast<ResponseType>()\n\npublic inline val ResponseType.Companion.ERROR: ResponseType get() = \"error\".asDynamic().unsafeCast<ResponseType>()\n\npublic inline val ResponseType.Companion.OPAQUE: ResponseType get() = \"opaque\".asDynamic().unsafeCast<ResponseType>()\n\npublic inline val ResponseType.Companion.OPAQUEREDIRECT: ResponseType get() = \"opaqueredirect\".asDynamic().unsafeCast<ResponseType>()","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.mediacapture\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\n/**\n * Exposes the JavaScript [MediaStream](https://developer.mozilla.org/en/docs/Web/API/MediaStream) to Kotlin\n */\npublic external open class MediaStream() : EventTarget, MediaProvider {\n   constructor(stream: MediaStream)\n   constructor(tracks: Array<MediaStreamTrack>)\n   open val id: String\n   open val active: Boolean\n   var onaddtrack: ((MediaStreamTrackEvent) -> dynamic)?\n   var onremovetrack: ((MediaStreamTrackEvent) -> dynamic)?\n   fun getAudioTracks(): Array<MediaStreamTrack>\n   fun getVideoTracks(): Array<MediaStreamTrack>\n   fun getTracks(): Array<MediaStreamTrack>\n   fun getTrackById(trackId: String): MediaStreamTrack?\n   fun addTrack(track: MediaStreamTrack)\n   fun removeTrack(track: MediaStreamTrack)\n   fun clone(): MediaStream\n}\n\n/**\n * Exposes the JavaScript [MediaStreamTrack](https://developer.mozilla.org/en/docs/Web/API/MediaStreamTrack) to Kotlin\n */\npublic external abstract class MediaStreamTrack : EventTarget {\n   open val kind: String\n   open val id: String\n   open val label: String\n   open var enabled: Boolean\n   open val muted: Boolean\n   open var onmute: ((Event) -> dynamic)?\n   open var onunmute: ((Event) -> dynamic)?\n   open val readyState: MediaStreamTrackState\n   open var onended: ((Event) -> dynamic)?\n   open var onoverconstrained: ((Event) -> dynamic)?\n   fun clone(): MediaStreamTrack\n   fun stop()\n   fun getCapabilities(): MediaTrackCapabilities\n   fun getConstraints(): MediaTrackConstraints\n   fun getSettings(): MediaTrackSettings\n   fun applyConstraints(constraints: MediaTrackConstraints = definedExternally): Promise<Unit>\n}\n\n/**\n * Exposes the JavaScript [MediaTrackSupportedConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaTrackSupportedConstraints)

```
to Kotlin\n */\npublic external interface MediaTrackSupportedConstraints {\n    var width: Boolean? /* = true */\n
  get() = definedExternally\n        set(value) = definedExternally\n    var height: Boolean? /* = true */\n        get() =
definedExternally\n        set(value) = definedExternally\n    var aspectRatio: Boolean? /* = true */\n        get() =
definedExternally\n        set(value) = definedExternally\n    var frameRate: Boolean? /* = true */\n        get() =
definedExternally\n        set(value) = definedExternally\n    var facingMode: Boolean? /* = true */\n        get() =
definedExternally\n        set(value) = definedExternally\n    var resizeMode: Boolean? /* = true */\n        get() =
definedExternally\n        set(value) = definedExternally\n    var volume: Boolean? /* = true */\n        get() =
definedExternally\n        set(value) = definedExternally\n    var sampleRate:
 Boolean? /* = true */\n        get() = definedExternally\n        set(value) = definedExternally\n    var sampleSize:
Boolean? /* = true */\n        get() = definedExternally\n        set(value) = definedExternally\n    var
echoCancellation: Boolean? /* = true */\n        get() = definedExternally\n        set(value) = definedExternally\n
var autoGainControl: Boolean? /* = true */\n        get() = definedExternally\n        set(value) = definedExternally\n
var noiseSuppression: Boolean? /* = true */\n        get() = definedExternally\n        set(value) = definedExternally\n
var latency: Boolean? /* = true */\n        get() = definedExternally\n        set(value) = definedExternally\n    var
channelCount: Boolean? /* = true */\n        get() = definedExternally\n        set(value) = definedExternally\n    var
deviceId: Boolean? /* = true */\n        get() = definedExternally\n        set(value) = definedExternally\n    var
groupId: Boolean? /* = true */\n        get() = definedExternally\n
        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun
MediaTrackSupportedConstraints(width: Boolean? = true, height: Boolean? = true, aspectRatio: Boolean? = true,
frameRate: Boolean? = true, facingMode: Boolean? = true, resizeMode: Boolean? = true, volume: Boolean? = true,
sampleRate: Boolean? = true, sampleSize: Boolean? = true, echoCancellation: Boolean? = true, autoGainControl:
Boolean? = true, noiseSuppression: Boolean? = true, latency: Boolean? = true, channelCount: Boolean? = true,
deviceId: Boolean? = true, groupId: Boolean? = true): MediaTrackSupportedConstraints {\n    val o = js(\"({})\")\n
o[\"width\"] = width\n    o[\"height\"] = height\n    o[\"aspectRatio\"] = aspectRatio\n    o[\"frameRate\"] =
frameRate\n    o[\"facingMode\"] = facingMode\n    o[\"resizeMode\"] = resizeMode\n    o[\"volume\"] = volume\n
o[\"sampleRate\"] = sampleRate\n    o[\"sampleSize\"]
 = sampleSize\n    o[\"echoCancellation\"] = echoCancellation\n    o[\"autoGainControl\"] = autoGainControl\n
o[\"noiseSuppression\"] = noiseSuppression\n    o[\"latency\"] = latency\n    o[\"channelCount\"] = channelCount\n
o[\"deviceId\"] = deviceId\n    o[\"groupId\"] = groupId\n    return o\n}\n\npublic external interface
MediaTrackCapabilities {\n    var width: ULongRange?\n        get() = definedExternally\n        set(value) =
definedExternally\n    var height: ULongRange?\n        get() = definedExternally\n        set(value) =
definedExternally\n    var aspectRatio: DoubleRange?\n        get() = definedExternally\n        set(value) =
definedExternally\n    var frameRate: DoubleRange?\n        get() = definedExternally\n        set(value) =
definedExternally\n    var facingMode: Array<String>?\n        get() = definedExternally\n        set(value) =
definedExternally\n    var resizeMode: Array<String>?\n        get() = definedExternally\n        set(value) =
definedExternally\n
   var volume: DoubleRange?\n        get() = definedExternally\n        set(value) = definedExternally\n    var
sampleRate: ULongRange?\n        get() = definedExternally\n        set(value) = definedExternally\n    var
sampleSize: ULongRange?\n        get() = definedExternally\n        set(value) = definedExternally\n    var
echoCancellation: Array<Boolean>?\n        get() = definedExternally\n        set(value) = definedExternally\n    var
autoGainControl: Array<Boolean>?\n        get() = definedExternally\n        set(value) = definedExternally\n    var
noiseSuppression: Array<Boolean>?\n        get() = definedExternally\n        set(value) = definedExternally\n    var
latency: DoubleRange?\n        get() = definedExternally\n        set(value) = definedExternally\n    var channelCount:
ULongRange?\n        get() = definedExternally\n        set(value) = definedExternally\n    var deviceId: String?\n
get() = definedExternally\n        set(value) = definedExternally\n
   var groupId: String?\n        get() = definedExternally\n        set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
```

\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaTrackCapabilities(width: ULongRange? = undefined, height: ULongRange? = undefined, aspectRatio: DoubleRange? = undefined, frameRate: DoubleRange? = undefined, facingMode: Array<String>? = undefined, resizeMode: Array<String>? = undefined, volume: DoubleRange? = undefined, sampleRate: ULongRange? = undefined, sampleSize: ULongRange? = undefined, echoCancellation: Array<Boolean>? = undefined, autoGainControl: Array<Boolean>? = undefined, noiseSuppression: Array<Boolean>? = undefined, latency: DoubleRange? = undefined, channelCount: ULongRange? = undefined, deviceId: String? = undefined, groupId: String? = undefined): MediaTrackCapabilities {\n val o = js(\"({})\")\n o[\"width\"] = width\n o[\"height\"] = height\n o[\"aspectRatio\"] = aspectRatio\n o[\"frameRate\"]
= frameRate\n o[\"facingMode\"] = facingMode\n o[\"resizeMode\"] = resizeMode\n o[\"volume\"] = volume\n o[\"sampleRate\"] = sampleRate\n o[\"sampleSize\"] = sampleSize\n o[\"echoCancellation\"] = echoCancellation\n o[\"autoGainControl\"] = autoGainControl\n o[\"noiseSuppression\"] = noiseSuppression\n o[\"latency\"] = latency\n o[\"channelCount\"] = channelCount\n o[\"deviceId\"] = deviceId\n o[\"groupId\"] = groupId\n return o\n}\n\n/**\n * Exposes the JavaScript [MediaTrackConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaTrackConstraints) to Kotlin\n */\npublic external interface MediaTrackConstraints : MediaTrackConstraintSet {\n var advanced: Array<MediaTrackConstraintSet>?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaTrackConstraints(advanced: Array<MediaTrackConstraintSet>?
= undefined, width: dynamic = undefined, height: dynamic = undefined, aspectRatio: dynamic = undefined, frameRate: dynamic = undefined, facingMode: dynamic = undefined, resizeMode: dynamic = undefined, volume: dynamic = undefined, sampleRate: dynamic = undefined, sampleSize: dynamic = undefined, echoCancellation: dynamic = undefined, autoGainControl: dynamic = undefined, noiseSuppression: dynamic = undefined, latency: dynamic = undefined, channelCount: dynamic = undefined, deviceId: dynamic = undefined, groupId: dynamic = undefined): MediaTrackConstraints {\n val o = js(\"({})\")\n o[\"advanced\"] = advanced\n o[\"width\"] = width\n o[\"height\"] = height\n o[\"aspectRatio\"] = aspectRatio\n o[\"frameRate\"] = frameRate\n o[\"facingMode\"] = facingMode\n o[\"resizeMode\"] = resizeMode\n o[\"volume\"] = volume\n o[\"sampleRate\"] = sampleRate\n o[\"sampleSize\"] = sampleSize\n o[\"echoCancellation\"] = echoCancellation\n o[\"autoGainControl\"] =
autoGainControl\n o[\"noiseSuppression\"] = noiseSuppression\n o[\"latency\"] = latency\n o[\"channelCount\"] = channelCount\n o[\"deviceId\"] = deviceId\n o[\"groupId\"] = groupId\n return o\n}\n\npublic external interface MediaTrackConstraintSet {\n var width: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var height: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var aspectRatio: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var frameRate: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var facingMode: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var resizeMode: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var volume: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var sampleRate:
dynamic\n get() = definedExternally\n set(value) = definedExternally\n var sampleSize: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var echoCancellation: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var autoGainControl: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var noiseSuppression: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var latency: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var channelCount: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var deviceId: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var groupId: dynamic\n get() = definedExternally\n set(value) =

definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic
inline fun MediaTrackConstraintSet(width: dynamic = undefined, height: dynamic = undefined, aspectRatio:
dynamic = undefined, frameRate: dynamic = undefined, facingMode: dynamic = undefined, resizeMode: dynamic =
undefined, volume: dynamic = undefined, sampleRate: dynamic = undefined, sampleSize: dynamic = undefined,
echoCancellation: dynamic = undefined, autoGainControl: dynamic = undefined, noiseSuppression: dynamic =
undefined, latency: dynamic = undefined, channelCount: dynamic = undefined, deviceId: dynamic = undefined,
groupId: dynamic = undefined): MediaTrackConstraintSet {\n val o = js(\"({})\")\n o[\"width\"] = width\n
o[\"height\"] = height\n o[\"aspectRatio\"] = aspectRatio\n o[\"frameRate\"] = frameRate\n o[\"facingMode\"]
= facingMode\n o[\"resizeMode\"] = resizeMode\n o[\"volume\"] = volume\n o[\"sampleRate\"] =
sampleRate\n o[\"sampleSize\"] = sampleSize\n o[\"echoCancellation\"] = echoCancellation\n
o[\"autoGainControl\"]
= autoGainControl\n o[\"noiseSuppression\"] = noiseSuppression\n o[\"latency\"] = latency\n
o[\"channelCount\"] = channelCount\n o[\"deviceId\"] = deviceId\n o[\"groupId\"] = groupId\n return
o\n}\n\n/**\n * Exposes the JavaScript
[MediaTrackSettings](https://developer.mozilla.org/en/docs/Web/API/MediaTrackSettings) to Kotlin\n */\npublic
external interface MediaTrackSettings {\n var width: Int?\n get() = definedExternally\n set(value) =
definedExternally\n var height: Int?\n get() = definedExternally\n set(value) = definedExternally\n var
aspectRatio: Double?\n get() = definedExternally\n set(value) = definedExternally\n var frameRate:
Double?\n get() = definedExternally\n set(value) = definedExternally\n var facingMode: String?\n
get() = definedExternally\n set(value) = definedExternally\n var resizeMode: String?\n get() =
definedExternally\n set(value)
= definedExternally\n var volume: Double?\n get() = definedExternally\n set(value) =
definedExternally\n var sampleRate: Int?\n get() = definedExternally\n set(value) = definedExternally\n
var sampleSize: Int?\n get() = definedExternally\n set(value) = definedExternally\n var echoCancellation:
Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var autoGainControl: Boolean?\n
get() = definedExternally\n set(value) = definedExternally\n var noiseSuppression: Boolean?\n get() =
definedExternally\n set(value) = definedExternally\n var latency: Double?\n get() = definedExternally\n
set(value) = definedExternally\n var channelCount: Int?\n get() = definedExternally\n set(value) =
definedExternally\n var deviceId: String?\n get() = definedExternally\n set(value) = definedExternally\n
var groupId: String?\n
get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaTrackSettings(width: Int? =
undefined, height: Int? = undefined, aspectRatio: Double? = undefined, frameRate: Double? = undefined,
facingMode: String? = undefined, resizeMode: String? = undefined, volume: Double? = undefined, sampleRate: Int?
= undefined, sampleSize: Int? = undefined, echoCancellation: Boolean? = undefined, autoGainControl: Boolean? =
undefined, noiseSuppression: Boolean? = undefined, latency: Double? = undefined, channelCount: Int? = undefined,
deviceId: String? = undefined, groupId: String? = undefined): MediaTrackSettings {\n val o = js(\"({})\")\n
o[\"width\"] = width\n o[\"height\"] = height\n o[\"aspectRatio\"] = aspectRatio\n o[\"frameRate\"] =
frameRate\n o[\"facingMode\"] = facingMode\n o[\"resizeMode\"] = resizeMode\n o[\"volume\"] = volume\n
o[\"sampleRate\"]
= sampleRate\n o[\"sampleSize\"] = sampleSize\n o[\"echoCancellation\"] = echoCancellation\n
o[\"autoGainControl\"] = autoGainControl\n o[\"noiseSuppression\"] = noiseSuppression\n o[\"latency\"] =
latency\n o[\"channelCount\"] = channelCount\n o[\"deviceId\"] = deviceId\n o[\"groupId\"] = groupId\n
return o\n}\n\n/**\n * Exposes the JavaScript
[MediaStreamTrackEvent](https://developer.mozilla.org/en/docs/Web/API/MediaStreamTrackEvent) to Kotlin\n
*/\npublic external open class MediaStreamTrackEvent(type: String, eventInitDict: MediaStreamTrackEventInit) :
Event {\n open val track: MediaStreamTrack\n\n companion object {\n val NONE: Short\n val

CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface MediaStreamTrackEventInit : EventInit {\n var track: MediaStreamTrack?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaStreamTrackEventInit(track: MediaStreamTrack?, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): MediaStreamTrackEventInit {\n val o = js(\"({})\")\n o[\"track\"] = track\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\npublic external open class OverconstrainedErrorEvent(type: String, eventInitDict: OverconstrainedErrorEventInit) : Event {\n open val error: dynamic\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface OverconstrainedErrorEventInit : EventInit {\n var error: dynamic /* = null */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun OverconstrainedErrorEventInit(error: dynamic = null, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): OverconstrainedErrorEventInit {\n val o = js(\"({})\")\n o[\"error\"] = error\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript [MediaDevices](https://developer.mozilla.org/en/docs/Web/API/MediaDevices) to Kotlin\n */\npublic external abstract class MediaDevices : EventTarget {\n open var ondevicechange: ((Event) -> dynamic)?\n fun enumerateDevices(): Promise<Array<MediaDeviceInfo>>\n fun getSupportedConstraints(): MediaTrackSupportedConstraints\n fun getUserMedia(constraints: MediaStreamConstraints = definedExternally): Promise<MediaStream>\n}\n\n/**\n * Exposes the JavaScript [MediaDeviceInfo](https://developer.mozilla.org/en/docs/Web/API/MediaDeviceInfo) to Kotlin\n */\npublic external abstract class MediaDeviceInfo {\n open val deviceId: String\n open val kind: MediaDeviceKind\n open val label: String\n open val groupId: String\n fun toJSON(): dynamic\n}\n\npublic external abstract class InputDeviceInfo : MediaDeviceInfo {\n fun getCapabilities(): MediaTrackCapabilities\n}\n\n/**\n * Exposes the JavaScript [MediaStreamConstraints](https://developer.mozilla.org/en/docs/Web/API/MediaStreamConstraints) to Kotlin\n */\npublic external interface MediaStreamConstraints {\n var video: dynamic /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var audio: dynamic /* = false */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun MediaStreamConstraints(video: dynamic = false, audio: dynamic = false): MediaStreamConstraints {\n val o = js(\"({})\")\n o[\"video\"] = video\n o[\"audio\"] = audio\n return o\n}\n\npublic external interface ConstrainablePattern {\n var onoverconstrained: ((Event) -> dynamic)?\n get() = definedExternally\n set(value) = definedExternally\n fun getCapabilities(): Capabilities\n fun getConstraints(): Constraints\n fun getSettings(): Settings\n fun applyConstraints(constraints: Constraints = definedExternally): Promise<Unit>\n}\n\n/**\n * Exposes the JavaScript [DoubleRange](https://developer.mozilla.org/en/docs/Web/API/DoubleRange) to Kotlin\n */\npublic external interface DoubleRange {\n var max: Double?\n get() = definedExternally\n set(value) = definedExternally\n var min: Double?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun DoubleRange(max: Double? = undefined, min: Double? = undefined): DoubleRange {\n val o = js(\"({})\")\n o[\"max\"] = max\n o[\"min\"] = min\n return o\n}\n\npublic external interface ConstrainDoubleRange : DoubleRange {\n var exact: Double?\n get() = definedExternally\n set(value) = definedExternally\n var ideal: Double?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ConstrainDoubleRange(exact: Double?

= undefined, ideal: Double? = undefined, max: Double? = undefined, min: Double? = undefined): ConstrainDoubleRange {\n val o = js(\"({})\")\n o[\"exact\"] = exact\n o[\"ideal\"] = ideal\n o[\"max\"] = max\n o[\"min\"] = min\n return o\n}\n\npublic external interface ULongRange {\n var max: Int?\n get() = definedExternally\n set(value) = definedExternally\n var min: Int?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ULongRange(max: Int? = undefined, min: Int? = undefined): ULongRange {\n val o = js(\"({})\")\n o[\"max\"] = max\n o[\"min\"] = min\n return o\n}\n\npublic external interface ConstrainULongRange : ULongRange {\n var exact: Int?\n get() = definedExternally\n set(value) = definedExternally\n var ideal: Int?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ConstrainULongRange(exact: Int? = undefined, ideal: Int? = undefined, max: Int? = undefined, min: Int? = undefined): ConstrainULongRange {\n val o = js(\"({})\")\n o[\"exact\"] = exact\n o[\"ideal\"] = ideal\n o[\"max\"] = max\n o[\"min\"] = min\n return o\n}\n\n/**\n * Exposes the JavaScript [ConstrainBooleanParameters](https://developer.mozilla.org/en/docs/Web/API/ConstrainBooleanParameters) to Kotlin\n */\npublic external interface ConstrainBooleanParameters {\n var exact: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n var ideal: Boolean?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ConstrainBooleanParameters(exact: Boolean? = undefined, ideal: Boolean? = undefined): ConstrainBooleanParameters {\n val o = js(\"({})\")\n o[\"exact\"] = exact\n o[\"ideal\"] = ideal\n return o\n}\n\n/**\n * Exposes the JavaScript [ConstrainDOMStringParameters](https://developer.mozilla.org/en/docs/Web/API/ConstrainDOMStringParameters) to Kotlin\n */\npublic external interface ConstrainDOMStringParameters {\n var exact: dynamic\n get() = definedExternally\n set(value) = definedExternally\n var ideal: dynamic\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ConstrainDOMStringParameters(exact: dynamic = undefined, ideal: dynamic = undefined): ConstrainDOMStringParameters {\n val o = js(\"({})\")\n o[\"exact\"] = exact\n o[\"ideal\"] = ideal\n return o\n}\n\npublic external interface Capabilities\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun Capabilities(): Capabilities {\n val o = js(\"({})\")\n return o\n}\n\npublic external interface Settings\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun Settings(): Settings {\n val o = js(\"({})\")\n return o\n}\n\npublic external interface ConstraintSet\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ConstraintSet(): ConstraintSet {\n val o = js(\"({})\")\n return o\n}\n\npublic external interface Constraints : ConstraintSet {\n var advanced: Array<ConstraintSet>?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun Constraints(advanced: Array<ConstraintSet>? = undefined): Constraints {\n val o = js(\"({})\")\n o[\"advanced\"] = advanced\n return o\n}\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface MediaStreamTrackState {\n companion object\n}\n\npublic inline val MediaStreamTrackState.Companion.LIVE: MediaStreamTrackState get() = \"live\".asDynamic().unsafeCast<MediaStreamTrackState>()\n\npublic inline val MediaStreamTrackState.Companion.ENDED: MediaStreamTrackState get() = \"ended\".asDynamic().unsafeCast<MediaStreamTrackState>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external

interface VideoFacingModeEnum {\n
companion object\n}\n\npublic inline val VideoFacingModeEnum.Companion.USER: VideoFacingModeEnum get() = \"user\".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\npublic inline val VideoFacingModeEnum.Companion.ENVIRONMENT: VideoFacingModeEnum get() = \"environment\".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\npublic inline val VideoFacingModeEnum.Companion.LEFT: VideoFacingModeEnum get() = \"left\".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\npublic inline val VideoFacingModeEnum.Companion.RIGHT: VideoFacingModeEnum get() = \"right\".asDynamic().unsafeCast<VideoFacingModeEnum>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface VideoResizeModeEnum {\n companion object\n}\n\npublic inline val VideoResizeModeEnum.Companion.NONE: VideoResizeModeEnum get() = \"none\".asDynamic().unsafeCast<VideoResizeModeEnum>()\n\npublic inline val VideoResizeModeEnum.Companion.CROP_AND_SCALE: VideoResizeModeEnum get() = \"crop-and-scale\".asDynamic().unsafeCast<VideoResizeModeEnum>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface MediaDeviceKind {\n companion object\n}\n\npublic inline val MediaDeviceKind.Companion.AUDIOINPUT: MediaDeviceKind get() = \"audioinput\".asDynamic().unsafeCast<MediaDeviceKind>()\n\npublic inline val MediaDeviceKind.Companion.AUDIOOUTPUT: MediaDeviceKind get() = \"audiooutput\".asDynamic().unsafeCast<MediaDeviceKind>()\n\npublic inline val MediaDeviceKind.Companion.VIDEOINPUT: MediaDeviceKind get() = \"videoinput\".asDynamic().unsafeCast<MediaDeviceKind>()","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.mediasource\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\n/**\n * Exposes the JavaScript [MediaSource](https://developer.mozilla.org/en/docs/Web/API/MediaSource) to Kotlin\n */\npublic external open class MediaSource : EventTarget, MediaProvider {\n open val sourceBuffers: SourceBufferList\n open val activeSourceBuffers: SourceBufferList\n open val readyState: ReadyState\n var duration: Double\n var onsourceopen: ((Event) -> dynamic)?\n var onsourceended: ((Event) -> dynamic)?\n var onsourceclose: ((Event) -> dynamic)?\n fun addSourceBuffer(type: String): SourceBuffer\n fun removeSourceBuffer(sourceBuffer: SourceBuffer)\n fun endOfStream(error: EndOfStreamError = definedExternally)\n fun setLiveSeekableRange(start: Double, end: Double)\n fun clearLiveSeekableRange()\n\n companion object {\n fun isTypeSupported(type: String): Boolean\n }\n}\n\n/**\n * Exposes the JavaScript [SourceBuffer](https://developer.mozilla.org/en/docs/Web/API/SourceBuffer) to Kotlin\n */\npublic external abstract class SourceBuffer : EventTarget {\n open var mode: AppendMode\n open val updating: Boolean\n open val buffered: TimeRanges\n open var timestampOffset: Double\n open val audioTracks: AudioTrackList\n open val videoTracks: VideoTrackList\n open val textTracks: TextTrackList\n open var appendWindowStart: Double\n open var appendWindowEnd: Double\n open var onupdatestart: ((Event) -> dynamic)?\n open var onupdate: ((Event) -> dynamic)?\n open var onupdateend: ((Event) -> dynamic)?\n open var onerror: ((Event) -> dynamic)?\n open var onabort: ((Event) -> dynamic)?\n fun appendBuffer(data: dynamic)\n fun abort()\n fun remove(start: Double, end: Double)\n}\n\n/**\n * Exposes the JavaScript [SourceBufferList](https://developer.mozilla.org/en/docs/Web/API/SourceBufferList) to Kotlin\n */\npublic external

abstract class SourceBufferList : EventTarget {\n   open val length: Int\n   open var onaddsourcebuffer: ((Event) -> dynamic)?\n   open var onremovesourcebuffer: ((Event) -> dynamic)?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SourceBufferList.get(index: Int): SourceBuffer? = asDynamic()[index]\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface ReadyState {\n   companion object\n}\n\npublic inline val ReadyState.Companion.CLOSED: ReadyState get() = \"closed\".asDynamic().unsafeCast<ReadyState>()\n\npublic inline val ReadyState.Companion.OPEN: ReadyState get() = \"open\".asDynamic().unsafeCast<ReadyState>()\n\npublic inline val ReadyState.Companion.ENDED: ReadyState get() = \"ended\".asDynamic().unsafeCast<ReadyState>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface EndOfStreamError {\n   companion object\n}\n\npublic inline val EndOfStreamError.Companion.NETWORK: EndOfStreamError get() = \"network\".asDynamic().unsafeCast<EndOfStreamError>()\n\npublic inline val EndOfStreamError.Companion.DECODE: EndOfStreamError get() = \"decode\".asDynamic().unsafeCast<EndOfStreamError>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface AppendMode {\n   companion object\n}\n\npublic inline val AppendMode.Companion.SEGMENTS: AppendMode get() = \"segments\".asDynamic().unsafeCast<AppendMode>()\n\npublic inline val AppendMode.Companion.SEQUENCE: AppendMode get() = \"sequence\".asDynamic().unsafeCast<AppendMode>()","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt
file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.pointerevents\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\n\npublic external interface PointerEventInit : MouseEventInit {\n   var pointerId: Int? /* = 0 */\n   get() = definedExternally\n   set(value) = definedExternally\n   var width: Double? /* = 1.0 */\n   get() = definedExternally\n   set(value) = definedExternally\n   var height: Double? /* = 1.0 */\n   get() = definedExternally\n   set(value) = definedExternally\n   var pressure: Float? /* = 0f */\n   get() = definedExternally\n   set(value) = definedExternally\n   var tangentialPressure: Float? /* = 0f */\n   get() = definedExternally\n   set(value) = definedExternally\n   var tiltX: Int? /* = 0 */\n   get() = definedExternally\n   set(value) = definedExternally\n   var tiltY: Int? /* = 0 */\n   get() = definedExternally\n   set(value) = definedExternally\n   var twist: Int? /* = 0 */\n   get() = definedExternally\n   set(value) = definedExternally\n   var pointerType: String? /* = \"\" */\n   get() = definedExternally\n   set(value) = definedExternally\n   var isPrimary: Boolean? /* = false */\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun PointerEventInit(pointerId: Int? = 0, width: Double? = 1.0, height: Double? = 1.0, pressure: Float? = 0f, tangentialPressure: Float? = 0f, tiltX: Int? = 0, tiltY: Int? = 0, twist: Int? = 0, pointerType: String? = \"\", isPrimary: Boolean? = false, screenX: Int? = 0, screenY: Int? = 0, clientX: Int? = 0, clientY: Int? = 0, button: Short? = 0, buttons: Short? = 0, relatedTarget: EventTarget? = null, region: String? = null, ctrlKey: Boolean? = false, shiftKey: Boolean? = false, altKey: Boolean? = false, metaKey: Boolean? = false, modifierAltGraph: Boolean? = false, modifierCapsLock: Boolean? = false, modifierFn: Boolean? = false, modifierFnLock: Boolean? = false, modifierHyper: Boolean? = false, modifierNumLock: Boolean? = false, modifierScrollLock: Boolean? = false, modifierSuper: Boolean? = false, modifierSymbol: Boolean? = false, modifierSymbolLock: Boolean? = false, view: Window? = null, detail: Int? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): PointerEventInit {\n   val o = js(\"({})\")\n   o[\"pointerId\"] = pointerId\n   o[\"width\"] = width\n

o[\"height\"] = height\n   o[\"pressure\"] = pressure\n   o[\"tangentialPressure\"] = tangentialPressure\n   o[\"tiltX\"] = tiltX\n   o[\"tiltY\"] = tiltY\n   o[\"twist\"] = twist\n   o[\"pointerType\"] = pointerType\n   o[\"isPrimary\"] = isPrimary\n   o[\"screenX\"] = screenX\n   o[\"screenY\"] = screenY\n   o[\"clientX\"] = clientX\n   o[\"clientY\"] = clientY\n   o[\"button\"] = button\n   o[\"buttons\"] = buttons\n   o[\"relatedTarget\"] = relatedTarget\n   o[\"region\"] = region\n   o[\"ctrlKey\"] = ctrlKey\n   o[\"shiftKey\"] = shiftKey\n   o[\"altKey\"] = altKey\n   o[\"metaKey\"] = metaKey\n   o[\"modifierAltGraph\"] = modifierAltGraph\n   o[\"modifierCapsLock\"] = modifierCapsLock\n   o[\"modifierFn\"] = modifierFn\n   o[\"modifierFnLock\"] = modifierFnLock\n   o[\"modifierHyper\"] = modifierHyper\n   o[\"modifierNumLock\"] = modifierNumLock\n   o[\"modifierScrollLock\"] = modifierScrollLock\n   o[\"modifierSuper\"] = modifierSuper\n   o[\"modifierSymbol\"] = modifierSymbol\n   o[\"modifierSymbolLock\"] = modifierSymbolLock\n   o[\"view\"] = view\n   o[\"detail\"] = detail\n   o[\"bubbles\"] = bubbles\n   o[\"cancelable\"] = cancelable\n   o[\"composed\"] = composed\n   return o\n}\n\n/**\n * Exposes the JavaScript [PointerEvent](https://developer.mozilla.org/en/docs/Web/API/PointerEvent) to Kotlin\n */\npublic external open class PointerEvent(type: String, eventInitDict: PointerEventInit = definedExternally) : MouseEvent {\n   open val pointerId: Int\n   open val width: Double\n   open val height: Double\n   open val pressure: Float\n   open val tangentialPressure: Float\n   open val tiltX: Int\n   open val tiltY: Int\n   open val twist: Int\n   open val pointerType: String\n   open val isPrimary: Boolean\n\n   companion object {\n   val NONE: Short\n   val CAPTURING_PHASE: Short\n   val AT_TARGET: Short\n   val BUBBLING_PHASE: Short\n   }\n}","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.dom.svg\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.css.*\n\n/**\n * Exposes the JavaScript [SVGElement](https://developer.mozilla.org/en/docs/Web/API/SVGElement) to Kotlin\n */\npublic external abstract class SVGElement : Element, ElementCSSInlineStyle, GlobalEventHandlers, SVGElementInstance {\n   open val dataset: DOMStringMap\n   open val ownerSVGElement: SVGSVGElement?\n   open val viewportElement: SVGElement?\n   open var tabIndex: Int\n   fun focus()\n   fun blur()\n\n   companion object {\n   val ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val CDATA_SECTION_NODE: Short\n   val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE: Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n   val DOCUMENT_POSITION_FOLLOWING: Short\n   val DOCUMENT_POSITION_CONTAINS: Short\n   val DOCUMENT_POSITION_CONTAINED_BY: Short\n   val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n   }\n}\n\npublic external interface SVGBoundingBoxOptions {\n   var fill: Boolean? /* = true */\n   get() = definedExternally\n   set(value) = definedExternally\n   var stroke: Boolean? /* = false */\n   get() = definedExternally\n   set(value) = definedExternally\n   var markers: Boolean? /* = false */\n   get() = definedExternally\n   set(value) = definedExternally\n   var clipped: Boolean? /* = false */\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun SVGBoundingBoxOptions(fill: Boolean? = true, stroke: Boolean? = false, markers: Boolean? = false, clipped: Boolean? = false): SVGBoundingBoxOptions {\n   val o = js(\"({})\")\n   o[\"fill\"] = fill\n   o[\"stroke\"] = stroke\n   o[\"markers\"] = markers\n   o[\"clipped\"] = clipped\n   return o\n}\n\n/**\n * Exposes the JavaScript [SVGGraphicsElement](https://developer.mozilla.org/en/docs/Web/API/SVGGraphicsElement) to Kotlin\n

*/\npublic external abstract class SVGGraphicsElement : SVGElement, SVGTests {\n open val transform: SVGAnimatedTransformList\n fun getBBox(options: SVGBoundingBoxOptions = definedExternally): DOMRect\n fun getCTM(): DOMMatrix?\n fun getScreenCTM(): DOMMatrix?\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGGeometryElement](https://developer.mozilla.org/en/docs/Web/API/SVGGeometryElement) to Kotlin\n */\npublic external abstract class SVGGeometryElement : SVGGraphicsElement {\n open val pathLength: SVGAnimatedNumber\n fun isPointInFill(point: DOMPoint): Boolean\n fun isPointInStroke(point: DOMPoint): Boolean\n fun getTotalLength(): Float\n fun getPointAtLength(distance: Float): DOMPoint\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGNumber](https://developer.mozilla.org/en/docs/Web/API/SVGNumber) to Kotlin\n */\npublic external abstract class SVGNumber {\n open var value: Float\n}\n\n/**\n * Exposes the JavaScript [SVGLength](https://developer.mozilla.org/en/docs/Web/API/SVGLength) to Kotlin\n */\npublic external abstract class SVGLength {\n open val unitType: Short\n open var value: Float\n open var valueInSpecifiedUnits: Float\n open var valueAsString: String\n fun newValueSpecifiedUnits(unitType: Short, valueInSpecifiedUnits: Float)\n fun convertToSpecifiedUnits(unitType: Short)\n\n companion object {\n val SVG_LENGTHTYPE_UNKNOWN: Short\n val SVG_LENGTHTYPE_NUMBER: Short\n val SVG_LENGTHTYPE_PERCENTAGE: Short\n val SVG_LENGTHTYPE_EMS: Short\n val SVG_LENGTHTYPE_EXS: Short\n val SVG_LENGTHTYPE_PX: Short\n val SVG_LENGTHTYPE_CM: Short\n val SVG_LENGTHTYPE_MM: Short\n val SVG_LENGTHTYPE_IN: Short\n val SVG_LENGTHTYPE_PT: Short\n val SVG_LENGTHTYPE_PC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGAngle](https://developer.mozilla.org/en/docs/Web/API/SVGAngle) to Kotlin\n */\npublic external abstract class SVGAngle {\n open val unitType: Short\n open var value: Float\n open var valueInSpecifiedUnits: Float\n open var valueAsString: String\n fun newValueSpecifiedUnits(unitType: Short, valueInSpecifiedUnits: Float)\n fun convertToSpecifiedUnits(unitType: Short)\n\n companion object {\n val SVG_ANGLETYPE_UNKNOWN: Short\n val SVG_ANGLETYPE_UNSPECIFIED: Short\n val SVG_ANGLETYPE_DEG: Short\n val SVG_ANGLETYPE_RAD: Short\n val SVG_ANGLETYPE_GRAD: Short\n }\n}\n\npublic external abstract class SVGNameList {\n open val length: Int\n open val numberOfItems: Int\n fun clear()\n fun initialize(newItem: dynamic): dynamic\n fun insertItemBefore(newItem: dynamic, index: Int): dynamic\n fun replaceItem(newItem: dynamic, index: Int): dynamic\n fun removeItem(index: Int): dynamic\n fun appendItem(newItem: dynamic): dynamic\n fun getItem(index: Int):

dynamic\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGNameList.get(index: Int):
dynamic = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGNameList.set(index: Int,
newItem: dynamic) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGNumberList) to Kotlin\n */\npublic external
abstract class SVGNumberList {\n   open val length: Int\n   open val numberOfItems: Int\n   fun clear()\n   fun
initialize(newItem: SVGNumber): SVGNumber\n   fun insertItemBefore(newItem: SVGNumber, index: Int):
SVGNumber\n   fun replaceItem(newItem: SVGNumber, index: Int): SVGNumber\n   fun removeItem(index: Int):
SVGNumber\n   fun appendItem(newItem: SVGNumber): SVGNumber\n   fun getItem(index: Int):
SVGNumber\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGNumberList.get(index:
Int): SVGNumber? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic
inline operator fun SVGNumberList.set(index: Int, newItem: SVGNumber) { asDynamic()[index] = newItem
}\n\n/**\n * Exposes the JavaScript
[SVGLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGLengthList) to Kotlin\n */\npublic external
abstract class SVGLengthList {\n   open val length: Int\n   open val numberOfItems: Int\n   fun clear()\n   fun
initialize(newItem: SVGLength): SVGLength\n   fun insertItemBefore(newItem: SVGLength, index: Int):
SVGLength\n   fun replaceItem(newItem: SVGLength, index: Int): SVGLength\n   fun removeItem(index: Int):
SVGLength\n   fun appendItem(newItem: SVGLength): SVGLength\n   fun getItem(index: Int):
SVGLength\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGLengthList.get(index:
Int): SVGLength? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGLengthList.set(index:
Int, newItem: SVGLength) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGAnimatedBoolean](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedBoolean) to Kotlin\n
*/\npublic external abstract class SVGAnimatedBoolean {\n   open var baseVal: Boolean\n   open val animVal:
Boolean\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedEnumeration](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedEnumeration) to
Kotlin\n */\npublic external abstract class SVGAnimatedEnumeration {\n   open var baseVal: Short\n   open val
animVal: Short\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedInteger](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedInteger) to Kotlin\n
*/\npublic external abstract class SVGAnimatedInteger {\n   open var baseVal: Int\n   open val animVal:
Int\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedNumber](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumber) to Kotlin\n
*/\npublic external abstract class SVGAnimatedNumber
{\n   open var baseVal: Float\n   open val animVal: Float\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedLength](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLength) to Kotlin\n
*/\npublic external abstract class SVGAnimatedLength {\n   open val baseVal: SVGLength\n   open val animVal:
SVGLength\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedAngle](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedAngle) to Kotlin\n */\npublic
external abstract class SVGAnimatedAngle {\n   open val baseVal: SVGAngle\n   open val animVal:
SVGAngle\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedString](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedString) to Kotlin\n */\npublic
external abstract class SVGAnimatedString {\n   open var baseVal: String\n   open val animVal: String\n}\n\n/**\n
* Exposes the JavaScript [SVGAnimatedRect](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedRect)
to Kotlin\n */\npublic external abstract class SVGAnimatedRect

{\n open val baseVal: DOMRect\n open val animVal: DOMRectReadOnly\n}\n\n/**\n * Exposes the JavaScript [SVGAnimatedNumberList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedNumberList) to Kotlin\n */\npublic external abstract class SVGAnimatedNumberList {\n open val baseVal: SVGNumberList\n open val animVal: SVGNumberList\n}\n\n/**\n * Exposes the JavaScript [SVGAnimatedLengthList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedLengthList) to Kotlin\n */\npublic external abstract class SVGAnimatedLengthList {\n open val baseVal: SVGLengthList\n open val animVal: SVGLengthList\n}\n\n/**\n * Exposes the JavaScript [SVGStringList](https://developer.mozilla.org/en/docs/Web/API/SVGStringList) to Kotlin\n */\npublic external abstract class SVGStringList {\n open val length: Int\n open val numberOfItems: Int\n fun clear()\n fun initialize(newItem: String): String\n fun insertItemBefore(newItem: String, index: Int): String\n fun replaceItem(newItem:
String, index: Int): String\n fun removeItem(index: Int): String\n fun appendItem(newItem: String): String\n fun getItem(index: Int): String\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGStringList.get(index: Int): String? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGStringList.set(index: Int, newItem: String) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript [SVGUnitTypes](https://developer.mozilla.org/en/docs/Web/API/SVGUnitTypes) to Kotlin\n */\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface SVGUnitTypes {\n companion object {\n val SVG_UNIT_TYPE_UNKNOWN: Short\n val SVG_UNIT_TYPE_USERSPACEONUSE: Short\n val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGTests](https://developer.mozilla.org/en/docs/Web/API/SVGTests)
to Kotlin\n */\npublic external interface SVGTests {\n val requiredExtensions: SVGStringList\n val systemLanguage: SVGStringList\n}\n\npublic external interface SVGFitToViewBox {\n val viewBox: SVGAnimatedRect\n val preserveAspectRatio: SVGAnimatedPreserveAspectRatio\n}\n\n/**\n * Exposes the JavaScript [SVGZoomAndPan](https://developer.mozilla.org/en/docs/Web/API/SVGZoomAndPan) to Kotlin\n */\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface SVGZoomAndPan {\n var zoomAndPan: Short\n\n companion object {\n val SVG_ZOOMANDPAN_UNKNOWN: Short\n val SVG_ZOOMANDPAN_DISABLE: Short\n val SVG_ZOOMANDPAN_MAGNIFY: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGURIReference](https://developer.mozilla.org/en/docs/Web/API/SVGURIReference) to Kotlin\n */\npublic external interface SVGURIReference {\n val href: SVGAnimatedString\n}\n\n/**\n * Exposes the JavaScript [SVGSVGElement](https://developer.mozilla.org/en/docs/Web/API/SVGSVGElement)
to Kotlin\n */\npublic external abstract class SVGSVGElement : SVGGraphicsElement, SVGFitToViewBox, SVGZoomAndPan, WindowEventHandlers {\n open val x: SVGAnimatedLength\n open val y: SVGAnimatedLength\n open val width: SVGAnimatedLength\n open val height: SVGAnimatedLength\n open var currentScale: Float\n open val currentTranslate: DOMPointReadOnly\n fun getIntersectionList(rect: DOMRectReadOnly, referenceElement: SVGElement?): NodeList\n fun getEnclosureList(rect: DOMRectReadOnly, referenceElement: SVGElement?): NodeList\n fun checkIntersection(element: SVGElement, rect: DOMRectReadOnly): Boolean\n fun checkEnclosure(element: SVGElement, rect: DOMRectReadOnly): Boolean\n fun deselectAll()\n fun createSVGNumber(): SVGNumber\n fun createSVGLength(): SVGLength\n fun createSVGAngle(): SVGAngle\n fun createSVGPoint(): DOMPoint\n fun createSVGMatrix(): DOMMatrix\n fun createSVGRect(): DOMRect\n
fun createSVGTransform(): SVGTransform\n fun createSVGTransformFromMatrix(matrix: DOMMatrixReadOnly): SVGTransform\n fun getElementById(elementId: String): Element\n fun suspendRedraw(maxWaitMilliseconds: Int): Int\n fun unsuspendRedraw(suspendHandleID: Int)\n fun unsuspendRedrawAll()\n fun forceRedraw()\n\n companion object {\n val

```
SVG_ZOOMANDPAN_UNKNOWN: Short\n        val SVG_ZOOMANDPAN_DISABLE: Short\n        val
SVG_ZOOMANDPAN_MAGNIFY: Short\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE:
Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING:
 Short\n        val DOCUMENT_POSITION_FOLLOWING: Short\n        val
DOCUMENT_POSITION_CONTAINS: Short\n        val DOCUMENT_POSITION_CONTAINED_BY: Short\n
  val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the
JavaScript [SVGGElement](https://developer.mozilla.org/en/docs/Web/API/SVGGElement) to Kotlin\n */\npublic
external abstract class SVGGElement : SVGGraphicsElement {\n    companion object {\n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
        val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS:
Short\n        val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\npublic external abstract class
SVGUnknownElement : SVGGraphicsElement {\n    companion object {\n        val ELEMENT_NODE: Short\n
val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
     val DOCUMENT_POSITION_CONTAINED_BY:
 Short\n        val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes
the JavaScript [SVGDefsElement](https://developer.mozilla.org/en/docs/Web/API/SVGDefsElement) to Kotlin\n
*/\npublic external abstract class SVGDefsElement : SVGGraphicsElement {\n    companion object {\n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
     val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGDescElement](https://developer.mozilla.org/en/docs/Web/API/SVGDescElement) to Kotlin\n */\npublic
external abstract class SVGDescElement : SVGElement {\n    companion object {\n        val ELEMENT_NODE:
Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE:
Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
```

DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGMetadataElement](https://developer.mozilla.org/en/docs/Web/API/SVGMetadataElement) to Kotlin\n */\npublic external abstract class SVGMetadataElement : SVGElement {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGTitleElement](https://developer.mozilla.org/en/docs/Web/API/SVGTitleElement) to Kotlin\n */\npublic external abstract class SVGTitleElement : SVGElement {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGSymbolElement](https://developer.mozilla.org/en/docs/Web/API/SVGSymbolElement) to Kotlin\n */\npublic external abstract class SVGSymbolElement : SVGGraphicsElement, SVGFitToViewBox {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGUseElement](https://developer.mozilla.org/en/docs/Web/API/SVGUseElement) to Kotlin\n */\npublic external abstract class SVGUseElement : SVGGraphicsElement, SVGURIReference {\n open val x: SVGAnimatedLength\n open val y: SVGAnimatedLength\n open val width: SVGAnimatedLength\n open val height: SVGAnimatedLength\n open val instanceRoot: SVGElement?\n open val animatedInstanceRoot: SVGElement?\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val

ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external open class SVGUseElementShadowRoot : ShadowRoot {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external interface SVGElementInstance {\n val correspondingElement: SVGElement?\n get() = definedExternally\n val correspondingUseElement: SVGUseElement?\n get() = definedExternally\n}\n\npublic external open class ShadowAnimation(source: dynamic, newTarget: dynamic) {\n open val sourceAnimation: dynamic\n}\n\n/**\n * Exposes the JavaScript [SVGSwitchElement](https://developer.mozilla.org/en/docs/Web/API/SVGSwitchElement) to Kotlin\n */\npublic external abstract class SVGSwitchElement : SVGGraphicsElement {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external interface GetSVGDocument {\n fun getSVGDocument(): Document\n}\n\n/**\n * Exposes the JavaScript [SVGStyleElement](https://developer.mozilla.org/en/docs/Web/API/SVGStyleElement) to Kotlin\n */\npublic external abstract class SVGStyleElement : SVGElement, LinkStyle {\n open var type: String\n open var media: String\n open var title: String\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGTransform](https://developer.mozilla.org/en/docs/Web/API/SVGTransform) to Kotlin\n */\npublic external abstract class SVGTransform {\n open val type: Short\n open val matrix: DOMMatrix\n open val angle: Float\n fun setMatrix(matrix: DOMMatrixReadOnly)\n fun setTranslate(tx: Float, ty: Float)\n fun setScale(sx:

Float, sy: Float)\n   fun setRotate(angle: Float, cx: Float, cy:
Float)\n   fun setSkewX(angle: Float)\n   fun setSkewY(angle: Float)\n\n   companion object {\n   val
SVG_TRANSFORM_UNKNOWN: Short\n   val SVG_TRANSFORM_MATRIX: Short\n   val
SVG_TRANSFORM_TRANSLATE: Short\n   val SVG_TRANSFORM_SCALE: Short\n   val
SVG_TRANSFORM_ROTATE: Short\n   val SVG_TRANSFORM_SKEWX: Short\n   val
SVG_TRANSFORM_SKEWY: Short\n   }\n}\n\n/**\n * Exposes the JavaScript
[SVGTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGTransformList) to Kotlin\n */\npublic
external abstract class SVGTransformList {\n   open val length: Int\n   open val numberOfItems: Int\n   fun
clear()\n   fun initialize(newItem: SVGTransform): SVGTransform\n   fun insertItemBefore(newItem:
SVGTransform, index: Int): SVGTransform\n   fun replaceItem(newItem: SVGTransform, index: Int):
SVGTransform\n   fun removeItem(index: Int): SVGTransform\n   fun appendItem(newItem: SVGTransform):
SVGTransform\n   fun createSVGTransformFromMatrix(matrix:
DOMMatrixReadOnly): SVGTransform\n   fun consolidate(): SVGTransform?\n   fun getItem(index: Int):
SVGTransform\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGTransformList.get(index:
Int): SVGTransform? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGTransformList.set(index:
Int, newItem: SVGTransform) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGAnimatedTransformList](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedTransformList) to
Kotlin\n */\npublic external abstract class SVGAnimatedTransformList {\n   open val baseVal:
SVGTransformList\n   open val animVal: SVGTransformList\n}\n\n/**\n * Exposes the JavaScript
[SVGPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGPreserveAspectRatio) to Kotlin\n
*/\npublic external abstract class SVGPreserveAspectRatio
{\n   open var align: Short\n   open var meetOrSlice: Short\n\n   companion object {\n   val
SVG_PRESERVEASPECTRATIO_UNKNOWN: Short\n   val SVG_PRESERVEASPECTRATIO_NONE:
Short\n   val SVG_PRESERVEASPECTRATIO_XMINYMIN: Short\n   val
SVG_PRESERVEASPECTRATIO_XMIDYMIN: Short\n   val
SVG_PRESERVEASPECTRATIO_XMAXYMIN: Short\n   val
SVG_PRESERVEASPECTRATIO_XMINYMID: Short\n   val
SVG_PRESERVEASPECTRATIO_XMIDYMID: Short\n   val
SVG_PRESERVEASPECTRATIO_XMAXYMID: Short\n   val
SVG_PRESERVEASPECTRATIO_XMINYMAX: Short\n   val
SVG_PRESERVEASPECTRATIO_XMIDYMAX: Short\n   val
SVG_PRESERVEASPECTRATIO_XMAXYMAX: Short\n   val SVG_MEETORSLICE_UNKNOWN: Short\n
    val SVG_MEETORSLICE_MEET: Short\n   val SVG_MEETORSLICE_SLICE: Short\n   }\n}\n\n/**\n *
Exposes the JavaScript
[SVGAnimatedPreserveAspectRatio](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedPreserveAspect
Ratio) to Kotlin\n */\npublic external abstract class
SVGAnimatedPreserveAspectRatio {\n   open val baseVal: SVGPreserveAspectRatio\n   open val animVal:
SVGPreserveAspectRatio\n}\n\n/**\n * Exposes the JavaScript
[SVGPathElement](https://developer.mozilla.org/en/docs/Web/API/SVGPathElement) to Kotlin\n */\npublic
external abstract class SVGPathElement : SVGGeometryElement {\n   companion object {\n   val
ELEMENT_NODE: Short\n   val ATTRIBUTE_NODE: Short\n   val TEXT_NODE: Short\n   val
CDATA_SECTION_NODE: Short\n   val ENTITY_REFERENCE_NODE: Short\n   val ENTITY_NODE:
Short\n   val PROCESSING_INSTRUCTION_NODE: Short\n   val COMMENT_NODE: Short\n   val
DOCUMENT_NODE: Short\n   val DOCUMENT_TYPE_NODE: Short\n   val
DOCUMENT_FRAGMENT_NODE: Short\n   val NOTATION_NODE: Short\n   val
DOCUMENT_POSITION_DISCONNECTED: Short\n   val DOCUMENT_POSITION_PRECEDING: Short\n

val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGRectElement](https://developer.mozilla.org/en/docs/Web/API/SVGRectElement) to Kotlin\n */\npublic external abstract class SVGRectElement : SVGGeometryElement {\n open val x: SVGAnimatedLength\n open val y: SVGAnimatedLength\n open val width: SVGAnimatedLength\n open val height: SVGAnimatedLength\n open val rx: SVGAnimatedLength\n open val ry: SVGAnimatedLength\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGCircleElement](https://developer.mozilla.org/en/docs/Web/API/SVGCircleElement) to Kotlin\n */\npublic external abstract class SVGCircleElement : SVGGeometryElement {\n open val cx: SVGAnimatedLength\n open val cy: SVGAnimatedLength\n open val r: SVGAnimatedLength\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGEllipseElement](https://developer.mozilla.org/en/docs/Web/API/SVGEllipseElement) to Kotlin\n */\npublic external abstract class SVGEllipseElement : SVGGeometryElement {\n open val cx: SVGAnimatedLength\n open val cy: SVGAnimatedLength\n open val rx: SVGAnimatedLength\n open val ry: SVGAnimatedLength\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGLineElement](https://developer.mozilla.org/en/docs/Web/API/SVGLineElement) to Kotlin\n */\npublic external abstract class SVGLineElement : SVGGeometryElement {\n open val x1: SVGAnimatedLength\n open val y1: SVGAnimatedLength\n open val x2: SVGAnimatedLength\n open val y2: SVGAnimatedLength\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE:

```
 Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val
ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE:
Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
     val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGMeshElement](https://developer.mozilla.org/en/docs/Web/API/SVGMeshElement) to Kotlin\n */\npublic
external abstract class SVGMeshElement : SVGGeometryElement, SVGURIReference {\n    companion object {\n
    val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE:
 Short\n        val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val
ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE:
Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
     val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGAnimatedPoints](https://developer.mozilla.org/en/docs/Web/API/SVGAnimatedPoints) to Kotlin\n */\npublic
external interface SVGAnimatedPoints {\n    val points: SVGPointList\n    val animatedPoints:
SVGPointList\n}\n\npublic external abstract class SVGPointList {\n    open val length: Int\n
    open val numberOfItems: Int\n    fun clear()\n    fun initialize(newItem: DOMPoint): DOMPoint\n    fun
insertItemBefore(newItem: DOMPoint, index: Int): DOMPoint\n    fun replaceItem(newItem: DOMPoint, index:
Int): DOMPoint\n    fun removeItem(index: Int): DOMPoint\n    fun appendItem(newItem: DOMPoint):
DOMPoint\n    fun getItem(index: Int): DOMPoint\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGPointList.get(index: Int):
DOMPoint? = asDynamic()[index]\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun SVGPointList.set(index: Int,
newItem: DOMPoint) { asDynamic()[index] = newItem }\n\n/**\n * Exposes the JavaScript
[SVGPolylineElement](https://developer.mozilla.org/en/docs/Web/API/SVGPolylineElement) to Kotlin\n
*/\npublic external abstract class SVGPolylineElement : SVGGeometryElement, SVGAnimatedPoints {\n
companion object {\n        val
 ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
     val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGPolygonElement](https://developer.mozilla.org/en/docs/Web/API/SVGPolygonElement) to Kotlin\n
*/\npublic external abstract class SVGPolygonElement : SVGGeometryElement, SVGAnimatedPoints {\n
companion object {\n
     val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE:
Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
```

DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGTextContentElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextContentElement) to Kotlin\n */\npublic external abstract class SVGTextContentElement : SVGGraphicsElement {\n open val textLength: SVGAnimatedLength\n open val lengthAdjust: SVGAnimatedEnumeration\n fun getNumberOfChars(): Int\n fun getComputedTextLength(): Float\n fun getSubStringLength(charnum: Int, nchars: Int): Float\n fun getStartPositionOfChar(charnum: Int): DOMPoint\n fun getEndPositionOfChar(charnum: Int): DOMPoint\n fun getExtentOfChar(charnum: Int): DOMRect\n fun getRotationOfChar(charnum: Int): Float\n fun getCharNumAtPosition(point: DOMPoint): Int\n fun selectSubString(charnum: Int, nchars: Int)\n\n companion object {\n val LENGTHADJUST_UNKNOWN: Short\n val LENGTHADJUST_SPACING: Short\n val LENGTHADJUST_SPACINGANDGLYPHS: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n
val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGTextPositioningElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextPositioningElement) to Kotlin\n */\npublic external abstract class SVGTextPositioningElement : SVGTextContentElement {\n open val x: SVGAnimatedLengthList\n open val y: SVGAnimatedLengthList\n open val dx: SVGAnimatedLengthList\n open val dy: SVGAnimatedLengthList\n open val rotate: SVGAnimatedNumberList\n\n companion object {\n val LENGTHADJUST_UNKNOWN: Short\n val LENGTHADJUST_SPACING: Short\n val LENGTHADJUST_SPACINGANDGLYPHS: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGTextElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextElement) to Kotlin\n */\npublic external abstract class SVGTextElement : SVGTextPositioningElement {\n companion object {\n val LENGTHADJUST_UNKNOWN: Short\n val LENGTHADJUST_SPACING: Short\n val LENGTHADJUST_SPACINGANDGLYPHS: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val

DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGTSpanElement](https://developer.mozilla.org/en/docs/Web/API/SVGTSpanElement) to Kotlin\n */\npublic external abstract class SVGTSpanElement : SVGTextPositioningElement {\n companion object {\n val LENGTHADJUST_UNKNOWN: Short\n val LENGTHADJUST_SPACING: Short\n val LENGTHADJUST_SPACINGANDGLYPHS: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGTextPathElement](https://developer.mozilla.org/en/docs/Web/API/SVGTextPathElement) to Kotlin\n */\npublic external abstract class SVGTextPathElement : SVGTextContentElement, SVGURIReference {\n open val startOffset: SVGAnimatedLength\n open val method: SVGAnimatedEnumeration\n open val spacing: SVGAnimatedEnumeration\n\n companion object {\n val TEXTPATH_METHODTYPE_UNKNOWN: Short\n val TEXTPATH_METHODTYPE_ALIGN: Short\n val TEXTPATH_METHODTYPE_STRETCH: Short\n val TEXTPATH_SPACINGTYPE_UNKNOWN: Short\n val TEXTPATH_SPACINGTYPE_AUTO: Short\n val TEXTPATH_SPACINGTYPE_EXACT: Short\n val LENGTHADJUST_UNKNOWN: Short\n val LENGTHADJUST_SPACING: Short\n val LENGTHADJUST_SPACINGANDGLYPHS: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGImageElement](https://developer.mozilla.org/en/docs/Web/API/SVGImageElement) to Kotlin\n */\npublic external abstract class SVGImageElement : SVGGraphicsElement, SVGURIReference, HTMLOrSVGImageElement {\n open val x: SVGAnimatedLength\n open val y: SVGAnimatedLength\n open val width: SVGAnimatedLength\n open val height: SVGAnimatedLength\n open val preserveAspectRatio: SVGAnimatedPreserveAspectRatio\n open var crossOrigin: String?\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n

```
    val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGForeignObjectElement](https://developer.mozilla.org/en/docs/Web/API/SVGForeignObjectElement)
 to Kotlin\n */\npublic external abstract class SVGForeignObjectElement : SVGGraphicsElement {\n    open val x:
SVGAnimatedLength\n    open val y: SVGAnimatedLength\n    open val width: SVGAnimatedLength\n    open val
height: SVGAnimatedLength\n\n    companion object {\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC:
 Short\n    }\n}\n\npublic external abstract class SVGMarkerElement : SVGElement, SVGFitToViewBox {\n    open
val refX: SVGAnimatedLength\n    open val refY: SVGAnimatedLength\n    open val markerUnits:
SVGAnimatedEnumeration\n    open val markerWidth: SVGAnimatedLength\n    open val markerHeight:
SVGAnimatedLength\n    open val orientType: SVGAnimatedEnumeration\n    open val orientAngle:
SVGAnimatedAngle\n    open var orient: String\n    fun setOrientToAuto()\n    fun setOrientToAngle(angle:
SVGAngle)\n\n    companion object {\n        val SVG_MARKERUNITS_UNKNOWN: Short\n        val
SVG_MARKERUNITS_USERSPACEONUSE: Short\n        val SVG_MARKERUNITS_STROKEWIDTH:
Short\n        val SVG_MARKER_ORIENT_UNKNOWN: Short\n        val SVG_MARKER_ORIENT_AUTO:
Short\n        val SVG_MARKER_ORIENT_ANGLE: Short\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n
        val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
    val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGSolidcolorElement](https://developer.mozilla.org/en/docs/Web/API/SVGSolidcolorElement) to Kotlin\n
*/\npublic external abstract class SVGSolidcolorElement : SVGElement {\n    companion object {\n        val
ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val
CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n
    val ENTITY_NODE: Short\n        val PROCESSING_INSTRUCTION_NODE: Short\n        val
COMMENT_NODE: Short\n        val DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n
    val DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
    val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGGradientElement) to Kotlin\n
*/\npublic external abstract class SVGGradientElement : SVGElement, SVGURIReference, SVGUnitTypes {\n
open val gradientUnits: SVGAnimatedEnumeration\n    open val gradientTransform: SVGAnimatedTransformList\n
```

```
 open val spreadMethod: SVGAnimatedEnumeration\n\n    companion object {\n
   val SVG_SPREADMETHOD_UNKNOWN: Short\n        val SVG_SPREADMETHOD_PAD: Short\n        val
SVG_SPREADMETHOD_REFLECT: Short\n        val SVG_SPREADMETHOD_REPEAT: Short\n        val
SVG_UNIT_TYPE_UNKNOWN: Short\n        val SVG_UNIT_TYPE_USERSPACEONUSE: Short\n        val
SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n        val ELEMENT_NODE: Short\n        val
ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n        val
ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
   val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n
      val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the
JavaScript
[SVGLinearGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGLinearGradientElement) to
Kotlin\n */\npublic external abstract class SVGLinearGradientElement : SVGGradientElement {\n    open val x1:
SVGAnimatedLength\n    open val y1: SVGAnimatedLength\n    open val x2: SVGAnimatedLength\n    open val
y2: SVGAnimatedLength\n\n    companion object {\n        val SVG_SPREADMETHOD_UNKNOWN: Short\n
val SVG_SPREADMETHOD_PAD: Short\n        val SVG_SPREADMETHOD_REFLECT: Short\n        val
SVG_SPREADMETHOD_REPEAT: Short\n        val SVG_UNIT_TYPE_UNKNOWN: Short\n        val
SVG_UNIT_TYPE_USERSPACEONUSE: Short\n        val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX:
Short\n        val ELEMENT_NODE: Short\n        val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n
   val CDATA_SECTION_NODE: Short\n        val ENTITY_REFERENCE_NODE: Short\n        val
ENTITY_NODE: Short\n
      val PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
   val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY: Short\n        val
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n    }\n}\n\n/**\n * Exposes the JavaScript
[SVGRadialGradientElement](https://developer.mozilla.org/en/docs/Web/API/SVGRadialGradientElement) to
Kotlin\n */\npublic external abstract class SVGRadialGradientElement : SVGGradientElement {\n    open val cx:
SVGAnimatedLength\n    open val cy: SVGAnimatedLength\n    open val r: SVGAnimatedLength\n    open val fx:
SVGAnimatedLength\n    open val fy: SVGAnimatedLength\n    open val fr: SVGAnimatedLength\n\n    companion
 object {\n        val SVG_SPREADMETHOD_UNKNOWN: Short\n        val SVG_SPREADMETHOD_PAD:
Short\n        val SVG_SPREADMETHOD_REFLECT: Short\n        val SVG_SPREADMETHOD_REPEAT:
Short\n        val SVG_UNIT_TYPE_UNKNOWN: Short\n        val SVG_UNIT_TYPE_USERSPACEONUSE:
Short\n        val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n        val ELEMENT_NODE: Short\n
val ATTRIBUTE_NODE: Short\n        val TEXT_NODE: Short\n        val CDATA_SECTION_NODE: Short\n
val ENTITY_REFERENCE_NODE: Short\n        val ENTITY_NODE: Short\n        val
PROCESSING_INSTRUCTION_NODE: Short\n        val COMMENT_NODE: Short\n        val
DOCUMENT_NODE: Short\n        val DOCUMENT_TYPE_NODE: Short\n        val
DOCUMENT_FRAGMENT_NODE: Short\n        val NOTATION_NODE: Short\n        val
DOCUMENT_POSITION_DISCONNECTED: Short\n        val DOCUMENT_POSITION_PRECEDING: Short\n
   val DOCUMENT_POSITION_FOLLOWING: Short\n        val DOCUMENT_POSITION_CONTAINS: Short\n
    val DOCUMENT_POSITION_CONTAINED_BY:
```

Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class SVGMeshGradientElement : SVGGradientElement {\n companion object {\n val SVG_SPREADMETHOD_UNKNOWN: Short\n val SVG_SPREADMETHOD_PAD: Short\n val SVG_SPREADMETHOD_REFLECT: Short\n val SVG_SPREADMETHOD_REPEAT: Short\n val SVG_UNIT_TYPE_UNKNOWN: Short\n val SVG_UNIT_TYPE_USERSPACEONUSE: Short\n val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED:
Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class SVGMeshrowElement : SVGElement {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class SVGMeshpatchElement : SVGElement {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGStopElement](https://developer.mozilla.org/en/docs/Web/API/SVGStopElement) to Kotlin\n */\npublic external abstract class SVGStopElement : SVGElement {\n open val offset: SVGAnimatedNumber\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n *

Exposes the JavaScript [SVGPatternElement](https://developer.mozilla.org/en/docs/Web/API/SVGPatternElement) to Kotlin\n */\npublic external abstract class SVGPatternElement : SVGElement, SVGFitToViewBox, SVGURIReference, SVGUnitTypes {\n open val patternUnits: SVGAnimatedEnumeration\n open val patternContentUnits: SVGAnimatedEnumeration\n open val patternTransform: SVGAnimatedTransformList\n open val x: SVGAnimatedLength\n open val y: SVGAnimatedLength\n open val width: SVGAnimatedLength\n open val height: SVGAnimatedLength\n\n companion object {\n val SVG_UNIT_TYPE_UNKNOWN: Short\n val SVG_UNIT_TYPE_USERSPACEONUSE: Short\n val SVG_UNIT_TYPE_OBJECTBOUNDINGBOX: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class SVGHatchElement : SVGElement {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\npublic external abstract class SVGHatchpathElement : SVGElement {\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGCursorElement](https://developer.mozilla.org/en/docs/Web/API/SVGCursorElement) to Kotlin\n */\npublic external abstract class SVGCursorElement : SVGElement, SVGURIReference {\n open val x: SVGAnimatedLength\n open val y: SVGAnimatedLength\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING:

Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGScriptElement](https://developer.mozilla.org/en/docs/Web/API/SVGScriptElement) to Kotlin\n */\npublic external abstract class SVGScriptElement : SVGElement, SVGURIReference, HTMLOrSVGScriptElement {\n open var type: String\n open var crossOrigin: String?\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGAElement](https://developer.mozilla.org/en/docs/Web/API/SVGAElement) to Kotlin\n */\npublic external abstract class SVGAElement : SVGGraphicsElement, SVGURIReference {\n open val target: SVGAnimatedString\n open val download: SVGAnimatedString\n open val rel: SVGAnimatedString\n open val relList: SVGAnimatedString\n open val hreflang: SVGAnimatedString\n open val type: SVGAnimatedString\n\n companion object {\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}\n\n/**\n * Exposes the JavaScript [SVGViewElement](https://developer.mozilla.org/en/docs/Web/API/SVGViewElement) to Kotlin\n */\npublic external abstract class SVGViewElement : SVGElement, SVGFitToViewBox, SVGZoomAndPan {\n companion object {\n val SVG_ZOOMANDPAN_UNKNOWN: Short\n val SVG_ZOOMANDPAN_DISABLE: Short\n val SVG_ZOOMANDPAN_MAGNIFY: Short\n val ELEMENT_NODE: Short\n val ATTRIBUTE_NODE: Short\n val TEXT_NODE: Short\n val CDATA_SECTION_NODE: Short\n val ENTITY_REFERENCE_NODE: Short\n val ENTITY_NODE: Short\n val PROCESSING_INSTRUCTION_NODE: Short\n val COMMENT_NODE: Short\n val DOCUMENT_NODE: Short\n val DOCUMENT_TYPE_NODE: Short\n val DOCUMENT_FRAGMENT_NODE: Short\n val NOTATION_NODE: Short\n val DOCUMENT_POSITION_DISCONNECTED: Short\n val DOCUMENT_POSITION_PRECEDING: Short\n val DOCUMENT_POSITION_FOLLOWING: Short\n val DOCUMENT_POSITION_CONTAINS: Short\n val DOCUMENT_POSITION_CONTAINED_BY: Short\n val DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC: Short\n }\n}","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.files\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\nimport org.w3c.xhr.*\n\n/**\n * Exposes the JavaScript

[Blob](https://developer.mozilla.org/en/docs/Web/API/Blob) to Kotlin\n */\npublic external open class Blob(blobParts: Array<dynamic> = definedExternally, options: BlobPropertyBag = definedExternally) : MediaProvider, ImageBitmapSource {\n   open val size: Number\n   open val type: String\n   open val isClosed: Boolean\n   fun slice(start: Int = definedExternally, end: Int = definedExternally, contentType: String = definedExternally): Blob\n   fun close()\n}\n\npublic external interface BlobPropertyBag {\n   var type: String? /* = \"\" */\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun BlobPropertyBag(type: String? = \"\"): BlobPropertyBag {\n   val o = js(\"({})\")\n   o[\"type\"] = type\n   return o\n}\n\n/**\n * Exposes the JavaScript [File](https://developer.mozilla.org/en/docs/Web/API/File) to Kotlin\n */\npublic external open class File(fileBits: Array<dynamic>, fileName: String, options: FilePropertyBag = definedExternally) : Blob {\n   open val name: String\n   open val lastModified: Int\n}\n\npublic external interface FilePropertyBag : BlobPropertyBag {\n   var lastModified: Int?\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun FilePropertyBag(lastModified: Int? = undefined, type: String? = \"\"): FilePropertyBag {\n   val o = js(\"({})\")\n   o[\"lastModified\"] = lastModified\n   o[\"type\"] = type\n   return o\n}\n\n/**\n * Exposes the JavaScript [FileList](https://developer.mozilla.org/en/docs/Web/API/FileList) to Kotlin\n */\npublic external abstract class FileList : ItemArrayLike<File> {\n   override fun item(index: Int): File?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline operator fun FileList.get(index: Int): File? = asDynamic()[index]\n\n/**\n * Exposes the JavaScript [FileReader](https://developer.mozilla.org/en/docs/Web/API/FileReader) to Kotlin\n */\npublic external open class FileReader : EventTarget {\n   open val readyState: Short\n   open val result: dynamic\n   open val error: dynamic\n   var onloadstart: ((ProgressEvent) -> dynamic)?\n   var onprogress: ((ProgressEvent) -> dynamic)?\n   var onload: ((Event) -> dynamic)?\n   var onabort: ((Event) -> dynamic)?\n   var onerror: ((Event) -> dynamic)?\n   var onloadend: ((Event) -> dynamic)?\n   fun readAsArrayBuffer(blob: Blob)\n   fun readAsBinaryString(blob: Blob)\n   fun readAsText(blob: Blob, label: String = definedExternally)\n   fun readAsDataURL(blob: Blob)\n   fun abort()\n\n   companion object {\n   val EMPTY: Short\n   val LOADING: Short\n   val DONE: Short\n   }\n}\n\n/**\n * Exposes the JavaScript [FileReaderSync](https://developer.mozilla.org/en/docs/Web/API/FileReaderSync) to Kotlin\n */\npublic external open class FileReaderSync {\n   fun readAsArrayBuffer(blob: Blob): ArrayBuffer\n   fun readAsBinaryString(blob: Blob): String\n   fun readAsText(blob: Blob, label: String = definedExternally): String\n   fun readAsDataURL(blob: Blob): String\n}","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.notifications\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.events.*\nimport org.w3c.workers.*\n\n/**\n * Exposes the JavaScript [Notification](https://developer.mozilla.org/en/docs/Web/API/Notification) to Kotlin\n */\npublic external open class Notification(title: String, options: NotificationOptions = definedExternally) : EventTarget {\n   var onclick: ((MouseEvent) -> dynamic)?\n   var onerror: ((Event) -> dynamic)?\n   open val title: String\n   open val dir: NotificationDirection\n   open val lang: String\n   open val body: String\n   open val tag: String\n   open val image: String\n   open val icon: String\n   open val badge: String\n   open val sound: String\n   open val vibrate: Array<out Int>\n   open val timestamp: Number\n   open val renotify: Boolean\n   open val silent: Boolean\n   open val noscreen: Boolean\n   open val requireInteraction: Boolean\n   open val sticky: Boolean\n   open val data: Any?\n   open val actions: Array<out NotificationAction>\n   fun close()\n\n   companion object {\n   val permission: NotificationPermission\n

```
val maxActions: Int\n        fun requestPermission(deprecatedCallback: (NotificationPermission)
 -> Unit = definedExternally): Promise<NotificationPermission>\n    }\n}\n\npublic external interface
NotificationOptions {\n    var dir: NotificationDirection? /* = NotificationDirection.AUTO */\n        get() =
definedExternally\n        set(value) = definedExternally\n    var lang: String? /* = \"\" */\n        get() =
definedExternally\n        set(value) = definedExternally\n    var body: String? /* = \"\" */\n        get() =
definedExternally\n        set(value) = definedExternally\n    var tag: String? /* = \"\" */\n        get() =
definedExternally\n        set(value) = definedExternally\n    var image: String?\n        get() = definedExternally\n
set(value) = definedExternally\n    var icon: String?\n        get() = definedExternally\n        set(value) =
definedExternally\n    var badge: String?\n        get() = definedExternally\n        set(value) = definedExternally\n
var sound: String?\n        get() = definedExternally\n        set(value) = definedExternally\n
    var vibrate: dynamic\n        get() = definedExternally\n        set(value) = definedExternally\n    var timestamp:
Number?\n        get() = definedExternally\n        set(value) = definedExternally\n    var renotify: Boolean? /* = false
*/\n        get() = definedExternally\n        set(value) = definedExternally\n    var silent: Boolean? /* = false */\n
get() = definedExternally\n        set(value) = definedExternally\n    var noscreen: Boolean? /* = false */\n        get() =
definedExternally\n        set(value) = definedExternally\n    var requireInteraction: Boolean? /* = false */\n        get()
= definedExternally\n        set(value) = definedExternally\n    var sticky: Boolean? /* = false */\n        get() =
definedExternally\n        set(value) = definedExternally\n    var data: Any? /* = null */\n        get() =
definedExternally\n        set(value) = definedExternally\n    var actions: Array<NotificationAction>? /* = arrayOf()
*/\n        get() = definedExternally\n
        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun NotificationOptions(dir:
NotificationDirection? = NotificationDirection.AUTO, lang: String? = \"\", body: String? = \"\", tag: String? = \"\",
image: String? = undefined, icon: String? = undefined, badge: String? = undefined, sound: String? = undefined,
vibrate: dynamic = undefined, timestamp: Number? = undefined, renotify: Boolean? = false, silent: Boolean? =
false, noscreen: Boolean? = false, requireInteraction: Boolean? = false, sticky: Boolean? = false, data: Any? = null,
actions: Array<NotificationAction>? = arrayOf()): NotificationOptions {\n    val o = js(\"({})\")\n    o[\"dir\"] = dir\n
  o[\"lang\"] = lang\n    o[\"body\"] = body\n    o[\"tag\"] = tag\n    o[\"image\"] = image\n    o[\"icon\"] = icon\n
o[\"badge\"] = badge\n    o[\"sound\"] = sound\n    o[\"vibrate\"] = vibrate\n    o[\"timestamp\"] = timestamp\n
    o[\"renotify\"] = renotify\n    o[\"silent\"] = silent\n    o[\"noscreen\"] = noscreen\n    o[\"requireInteraction\"] =
requireInteraction\n    o[\"sticky\"] = sticky\n    o[\"data\"] = data\n    o[\"actions\"] = actions\n    return
o\n}\n\npublic external interface NotificationAction {\n    var action: String?\n    var title: String?\n    var icon:
String?\n        get() = definedExternally\n        set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun NotificationAction(action: String?,
title: String?, icon: String? = undefined): NotificationAction {\n    val o = js(\"({})\")\n    o[\"action\"] = action\n
o[\"title\"] = title\n    o[\"icon\"] = icon\n    return o\n}\n\npublic external interface GetNotificationOptions {\n    var
tag: String? /* = \"\" */\n        get() = definedExternally\n        set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic
 inline fun GetNotificationOptions(tag: String? = \"\"): GetNotificationOptions {\n    val o = js(\"({})\")\n
o[\"tag\"] = tag\n    return o\n}\n\n/**\n * Exposes the JavaScript
[NotificationEvent](https://developer.mozilla.org/en/docs/Web/API/NotificationEvent) to Kotlin\n */\npublic
external open class NotificationEvent(type: String, eventInitDict: NotificationEventInit) : ExtendableEvent {\n
open val notification: Notification\n    open val action: String\n\n    companion object {\n        val NONE: Short\n
 val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n
}\n}\n\npublic external interface NotificationEventInit : ExtendableEventInit {\n    var notification: Notification?\n
var action: String? /* = \"\" */\n        get() = definedExternally\n        set(value) =
definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
```

\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun NotificationEventInit(notification: Notification?, action: String? = \"\", bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): NotificationEventInit {\n val o = js(\"({})\")\n o[\"notification\"] = notification\n o[\"action\"] = action\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface NotificationPermission {\n companion object\n}\n\npublic inline val NotificationPermission.Companion.DEFAULT: NotificationPermission get() = \"default\".asDynamic().unsafeCast<NotificationPermission>()\n\npublic inline val NotificationPermission.Companion.DENIED: NotificationPermission get() = \"denied\".asDynamic().unsafeCast<NotificationPermission>()\n\npublic inline val NotificationPermission.Companion.GRANTED: NotificationPermission get() = \"granted\".asDynamic().unsafeCast<NotificationPermission>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface NotificationDirection {\n companion object\n}\n\npublic inline val NotificationDirection.Companion.AUTO: NotificationDirection get() = \"auto\".asDynamic().unsafeCast<NotificationDirection>()\n\npublic inline val NotificationDirection.Companion.LTR: NotificationDirection get() = \"ltr\".asDynamic().unsafeCast<NotificationDirection>()\n\npublic inline val NotificationDirection.Companion.RTL: NotificationDirection get() = \"rtl\".asDynamic().unsafeCast<NotificationDirection>()","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.workers\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\nimport org.w3c.fetch.*\nimport org.w3c.notifications.*\n\n/**\n * Exposes the JavaScript [ServiceWorker](https://developer.mozilla.org/en/docs/Web/API/ServiceWorker) to Kotlin\n */\npublic external abstract class ServiceWorker : EventTarget, AbstractWorker, UnionMessagePortOrServiceWorker, UnionClientOrMessagePortOrServiceWorker {\n open val scriptURL: String\n open val state: ServiceWorkerState\n open var onstatechange: ((Event) -> dynamic)?\n fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)\n}\n\n/**\n * Exposes the JavaScript [ServiceWorkerRegistration](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerRegistration) to Kotlin\n */\npublic external abstract class ServiceWorkerRegistration : EventTarget {\n open val installing: ServiceWorker?\n open val waiting: ServiceWorker?\n open val active: ServiceWorker?\n open val scope: String\n open var onupdatefound: ((Event) -> dynamic)?\n open val APISpace: dynamic\n fun update(): Promise<Unit>\n fun unregister(): Promise<Boolean>\n fun showNotification(title: String, options: NotificationOptions = definedExternally): Promise<Unit>\n fun getNotifications(filter: GetNotificationOptions = definedExternally): Promise<Array<Notification>>\n fun methodName(): Promise<dynamic>\n}\n\n/**\n * Exposes the JavaScript [ServiceWorkerContainer](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerContainer) to Kotlin\n */\npublic external abstract class ServiceWorkerContainer : EventTarget {\n open val controller: ServiceWorker?\n open val ready: Promise<ServiceWorkerRegistration>\n open var oncontrollerchange: ((Event) -> dynamic)?\n open var onmessage: ((MessageEvent) -> dynamic)?\n fun register(scriptURL: String, options: RegistrationOptions = definedExternally): Promise<ServiceWorkerRegistration>\n fun getRegistration(clientURL: String = definedExternally): Promise<Any?>\n fun getRegistrations(): Promise<Array<ServiceWorkerRegistration>>\n fun startMessages()\n}\n\npublic external interface RegistrationOptions {\n var scope: String?\n get() = definedExternally\n set(value) = definedExternally\n

var type: WorkerType? /* = WorkerType.CLASSIC */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun RegistrationOptions(scope: String? = undefined, type: WorkerType? = WorkerType.CLASSIC): RegistrationOptions {\n val o = js(\"({})\")\n o[\"scope\"] = scope\n o[\"type\"] = type\n return o\n}\n\n/**\n * Exposes the JavaScript [ServiceWorkerMessageEvent](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerMessageEvent) to Kotlin\n */\npublic external open class ServiceWorkerMessageEvent(type: String, eventInitDict: ServiceWorkerMessageEventInit
= definedExternally) : Event {\n open val data: Any?\n open val origin: String\n open val lastEventId: String\n open val source: UnionMessagePortOrServiceWorker?\n open val ports: Array<out MessagePort>?\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface ServiceWorkerMessageEventInit : EventInit {\n var data: Any?\n get() = definedExternally\n set(value) = definedExternally\n var origin: String?\n get() = definedExternally\n set(value) = definedExternally\n var lastEventId: String?\n get() = definedExternally\n set(value) = definedExternally\n var source: UnionMessagePortOrServiceWorker?\n get() = definedExternally\n set(value) = definedExternally\n var ports: Array<MessagePort>?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ServiceWorkerMessageEventInit(data: Any? = undefined, origin: String? = undefined, lastEventId: String? = undefined, source: UnionMessagePortOrServiceWorker? = undefined, ports: Array<MessagePort>? = undefined, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ServiceWorkerMessageEventInit {\n val o = js(\"({})\")\n o[\"data\"] = data\n o[\"origin\"] = origin\n o[\"lastEventId\"] = lastEventId\n o[\"source\"] = source\n o[\"ports\"] = ports\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript [ServiceWorkerGlobalScope](https://developer.mozilla.org/en/docs/Web/API/ServiceWorkerGlobalScope) to Kotlin\n */\npublic external abstract class ServiceWorkerGlobalScope : WorkerGlobalScope {\n open val clients: Clients\n
open val registration: ServiceWorkerRegistration\n open var oninstall: ((Event) -> dynamic)?\n open var onactivate: ((Event) -> dynamic)?\n open var onfetch: ((FetchEvent) -> dynamic)?\n open var onforeignfetch: ((Event) -> dynamic)?\n open var onmessage: ((MessageEvent) -> dynamic)?\n open var onnotificationclick: ((NotificationEvent) -> dynamic)?\n open var onnotificationclose: ((NotificationEvent) -> dynamic)?\n open var onfunctionalevent: ((Event) -> dynamic)?\n fun skipWaiting(): Promise<Unit>\n}\n\n/**\n * Exposes the JavaScript [Client](https://developer.mozilla.org/en/docs/Web/API/Client) to Kotlin\n */\npublic external abstract class Client : UnionClientOrMessagePortOrServiceWorker {\n open val url: String\n open val frameType: FrameType\n open val id: String\n fun postMessage(message: Any?, transfer: Array<dynamic> = definedExternally)\n}\n\n/**\n * Exposes the JavaScript [WindowClient](https://developer.mozilla.org/en/docs/Web/API/WindowClient)
to Kotlin\n */\npublic external abstract class WindowClient : Client {\n open val visibilityState: dynamic\n open val focused: Boolean\n fun focus(): Promise<WindowClient>\n fun navigate(url: String): Promise<WindowClient>\n}\n\n/**\n * Exposes the JavaScript [Clients](https://developer.mozilla.org/en/docs/Web/API/Clients) to Kotlin\n */\npublic external abstract class Clients {\n fun get(id: String): Promise<Any?>\n fun matchAll(options: ClientQueryOptions = definedExternally): Promise<Array<Client>>\n fun openWindow(url: String): Promise<WindowClient?>\n fun claim(): Promise<Unit>\n}\n\npublic external interface ClientQueryOptions {\n var includeUncontrolled: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var type: ClientType? /* = ClientType.WINDOW */\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic

```
inline fun ClientQueryOptions(includeUncontrolled: Boolean? = false, type: ClientType? = ClientType.WINDOW): ClientQueryOptions {\n    val o = js(\"({})\")\n    o[\"includeUncontrolled\"] = includeUncontrolled\n    o[\"type\"] = type\n    return o\n}\n\n/**\n * Exposes the JavaScript [ExtendableEvent](https://developer.mozilla.org/en/docs/Web/API/ExtendableEvent) to Kotlin\n */\npublic external open class ExtendableEvent(type: String, eventInitDict: ExtendableEventInit = definedExternally) : Event {\n    fun waitUntil(f: Promise<Any?>)\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface ExtendableEventInit : EventInit\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ExtendableEventInit(bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ExtendableEventInit {\n    val o = js(\"({})\")\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\n/**\n * Exposes the JavaScript [InstallEvent](https://developer.mozilla.org/en/docs/Web/API/InstallEvent) to Kotlin\n */\npublic external open class InstallEvent(type: String, eventInitDict: ExtendableEventInit = definedExternally) : ExtendableEvent {\n    fun registerForeignFetch(options: ForeignFetchOptions)\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface ForeignFetchOptions {\n    var scopes: Array<String>?\n    var origins: Array<String>?\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ForeignFetchOptions(scopes: Array<String>?, origins: Array<String>?): ForeignFetchOptions {\n    val o = js(\"({})\")\n    o[\"scopes\"] = scopes\n    o[\"origins\"] = origins\n    return o\n}\n\n/**\n * Exposes the JavaScript [FetchEvent](https://developer.mozilla.org/en/docs/Web/API/FetchEvent) to Kotlin\n */\npublic external open class FetchEvent(type: String, eventInitDict: FetchEventInit) : ExtendableEvent {\n    open val request: Request\n    open val clientId: String?\n    open val isReload: Boolean\n    fun respondWith(r: Promise<Response>)\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface FetchEventInit : ExtendableEventInit {\n    var request: Request?\n    var clientId: String? /* = null */\n        get() = definedExternally\n        set(value) = definedExternally\n    var isReload: Boolean? /* = false */\n        get() = definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun FetchEventInit(request: Request?, clientId: String? = null, isReload: Boolean? = false, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): FetchEventInit {\n    val o = js(\"({})\")\n    o[\"request\"] = request\n    o[\"clientId\"] = clientId\n    o[\"isReload\"] = isReload\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\npublic external open class ForeignFetchEvent(type: String, eventInitDict: ForeignFetchEventInit) : ExtendableEvent {\n    open val request: Request\n    open val origin: String\n    fun respondWith(r: Promise<ForeignFetchResponse>)\n\n    companion object {\n        val NONE: Short\n        val CAPTURING_PHASE: Short\n        val AT_TARGET: Short\n        val BUBBLING_PHASE: Short\n    }\n}\n\npublic external interface ForeignFetchEventInit : ExtendableEventInit {\n    var request: Request?\n    var origin: String? /* = \"null\" */\n        get() = definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ForeignFetchEventInit(request: Request?, origin: String? = \"null\", bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ForeignFetchEventInit {\n    val o = js(\"({})\")\n    o[\"request\"] = request\n    o[\"origin\"] = origin\n    o[\"bubbles\"] = bubbles\n    o[\"cancelable\"] = cancelable\n    o[\"composed\"] = composed\n    return o\n}\n\npublic external interface ForeignFetchResponse {\n    var response: Response?\n    var origin: String?\n        get() = definedExternally\n        set(value) = definedExternally\n    var headers: Array<String>?\n        get() = definedExternally\n        set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\",
```

\"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ForeignFetchResponse(response: Response?,
origin: String? = undefined, headers: Array<String>? = undefined): ForeignFetchResponse {\n val o = js(\"({})\")\n o[\"response\"] = response\n o[\"origin\"] = origin\n o[\"headers\"] = headers\n return o\n}\n\n/**\n * Exposes the JavaScript [ExtendableMessageEvent](https://developer.mozilla.org/en/docs/Web/API/ExtendableMessageEvent) to Kotlin\n */\npublic external open class ExtendableMessageEvent(type: String, eventInitDict: ExtendableMessageEventInit = definedExternally) : ExtendableEvent {\n open val data: Any?\n open val origin: String\n open val lastEventId: String\n open val source: UnionClientOrMessagePortOrServiceWorker?\n open val ports: Array<out MessagePort>?\n\n companion object {\n val NONE: Short\n val CAPTURING_PHASE: Short\n val AT_TARGET: Short\n val BUBBLING_PHASE: Short\n }\n}\n\npublic external interface ExtendableMessageEventInit : ExtendableEventInit {\n var data: Any?\n get() = definedExternally\n set(value) = definedExternally\n var origin: String?\n get() = definedExternally\n set(value) = definedExternally\n var lastEventId: String?\n get() = definedExternally\n set(value) = definedExternally\n var source: UnionClientOrMessagePortOrServiceWorker?\n get() = definedExternally\n set(value) = definedExternally\n var ports: Array<MessagePort>?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ExtendableMessageEventInit(data: Any? = undefined, origin: String? = undefined, lastEventId: String? = undefined, source: UnionClientOrMessagePortOrServiceWorker? = undefined, ports: Array<MessagePort>? = undefined, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ExtendableMessageEventInit {\n val o = js(\"({})\")\n o[\"data\"] = data\n o[\"origin\"] = origin\n o[\"lastEventId\"] = lastEventId\n o[\"source\"] = source\n o[\"ports\"] = ports\n o[\"bubbles\"] = bubbles\n o[\"cancelable\"] = cancelable\n o[\"composed\"] = composed\n return o\n}\n\n/**\n * Exposes the JavaScript [Cache](https://developer.mozilla.org/en/docs/Web/API/Cache) to Kotlin\n */\npublic external abstract class Cache {\n fun match(request: dynamic, options: CacheQueryOptions = definedExternally): Promise<Any?>\n fun matchAll(request: dynamic = definedExternally, options: CacheQueryOptions = definedExternally): Promise<Array<Response>>\n fun add(request: dynamic): Promise<Unit>\n fun addAll(requests: Array<dynamic>): Promise<Unit>\n fun put(request: dynamic, response: Response): Promise<Unit>\n fun delete(request: dynamic, options: CacheQueryOptions = definedExternally): Promise<Boolean>\n fun keys(request: dynamic = definedExternally, options: CacheQueryOptions = definedExternally): Promise<Array<Request>>\n}\n\npublic external interface CacheQueryOptions {\n var ignoreSearch: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var ignoreMethod: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var ignoreVary: Boolean? /* = false */\n get() = definedExternally\n set(value) = definedExternally\n var cacheName: String?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun CacheQueryOptions(ignoreSearch: Boolean? = false, ignoreMethod: Boolean? = false, ignoreVary: Boolean? = false, cacheName: String? = undefined): CacheQueryOptions {\n val o = js(\"({})\")\n o[\"ignoreSearch\"] = ignoreSearch\n o[\"ignoreMethod\"] = ignoreMethod\n o[\"ignoreVary\"] = ignoreVary\n o[\"cacheName\"] = cacheName\n return o\n}\n\npublic external
interface CacheBatchOperation {\n var type: String?\n get() = definedExternally\n set(value) = definedExternally\n var request: Request?\n get() = definedExternally\n set(value) = definedExternally\n var response: Response?\n get() = definedExternally\n set(value) = definedExternally\n var options: CacheQueryOptions?\n get() = definedExternally\n set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun CacheBatchOperation(type: String? =

undefined, request: Request? = undefined, response: Response? = undefined, options: CacheQueryOptions? = undefined): CacheBatchOperation {\n   val o = js(\"({})\")\n   o[\"type\"] = type\n   o[\"request\"] = request\n   o[\"response\"] = response\n   o[\"options\"] = options\n   return o\n}\n\n/**\n * Exposes the JavaScript [CacheStorage](https://developer.mozilla.org/en/docs/Web/API/CacheStorage)
 to Kotlin\n */\npublic external abstract class CacheStorage {\n   fun match(request: dynamic, options: CacheQueryOptions = definedExternally): Promise<Any?>\n   fun has(cacheName: String): Promise<Boolean>\n   fun open(cacheName: String): Promise<Cache>\n   fun delete(cacheName: String): Promise<Boolean>\n   fun keys(): Promise<Array<String>>\n}\n\npublic external open class FunctionalEvent : ExtendableEvent {\n   companion object {\n   val NONE: Short\n   val CAPTURING_PHASE: Short\n   val AT_TARGET: Short\n   val BUBBLING_PHASE: Short\n   }\n}\n\npublic external interface UnionMessagePortOrServiceWorker\n\npublic external interface UnionClientOrMessagePortOrServiceWorker\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface ServiceWorkerState {\n   companion object\n}\n\npublic inline val ServiceWorkerState.Companion.INSTALLING: ServiceWorkerState get() =
 \"installing\".asDynamic().unsafeCast<ServiceWorkerState>()\n\npublic inline val ServiceWorkerState.Companion.INSTALLED: ServiceWorkerState get() =
\"installed\".asDynamic().unsafeCast<ServiceWorkerState>()\n\npublic inline val ServiceWorkerState.Companion.ACTIVATING: ServiceWorkerState get() =
\"activating\".asDynamic().unsafeCast<ServiceWorkerState>()\n\npublic inline val ServiceWorkerState.Companion.ACTIVATED: ServiceWorkerState get() =
\"activated\".asDynamic().unsafeCast<ServiceWorkerState>()\n\npublic inline val ServiceWorkerState.Companion.REDUNDANT: ServiceWorkerState get() =
\"redundant\".asDynamic().unsafeCast<ServiceWorkerState>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface FrameType {\n   companion object\n}\n\npublic inline val FrameType.Companion.AUXILIARY: FrameType get() = \"auxiliary\".asDynamic().unsafeCast<FrameType>()\n\npublic inline val FrameType.Companion.TOP_LEVEL:
 FrameType get() = \"top-level\".asDynamic().unsafeCast<FrameType>()\n\npublic inline val FrameType.Companion.NESTED: FrameType get() = \"nested\".asDynamic().unsafeCast<FrameType>()\n\npublic inline val FrameType.Companion.NONE: FrameType get() =
\"none\".asDynamic().unsafeCast<FrameType>()\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface ClientType {\n   companion object\n}\n\npublic inline val ClientType.Companion.WINDOW: ClientType get() = \"window\".asDynamic().unsafeCast<ClientType>()\n\npublic inline val ClientType.Companion.WORKER: ClientType get() = \"worker\".asDynamic().unsafeCast<ClientType>()\n\npublic inline val ClientType.Companion.SHAREDWORKER: ClientType get() =
\"sharedworker\".asDynamic().unsafeCast<ClientType>()\n\npublic inline val ClientType.Companion.ALL: ClientType get() = \"all\".asDynamic().unsafeCast<ClientType>()","/*\n * Copyright 2010-2021 JetBrains s.r.o.
 and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// NOTE: THIS FILE IS AUTO-GENERATED, DO NOT EDIT!\n// See github.com/kotlin/dukat for details\n\npackage org.w3c.xhr\n\nimport kotlin.js.*\nimport org.khronos.webgl.*\nimport org.w3c.dom.*\nimport org.w3c.dom.events.*\nimport org.w3c.files.*\n\n/**\n * Exposes the JavaScript [XMLHttpRequestEventTarget](https://developer.mozilla.org/en/docs/Web/API/XMLHttpRequestEventTarget) to Kotlin\n */\npublic external abstract class XMLHttpRequestEventTarget : EventTarget {\n   open var onloadstart: ((ProgressEvent) -> dynamic)?\n   open var onprogress: ((ProgressEvent) -> dynamic)?\n   open var onabort: ((Event) -> dynamic)?\n   open var onerror: ((Event) -> dynamic)?\n   open var onload: ((Event) -> dynamic)?\n

open var ontimeout: ((Event) -> dynamic)?\n   open var onloadend: ((Event) -> dynamic)?\n}\n\npublic external abstract class XMLHttpRequestUpload : XMLHttpRequestEventTarget\n\n/**\n * Exposes the JavaScript [XMLHttpRequest](https://developer.mozilla.org/en/docs/Web/API/XMLHttpRequest) to Kotlin\n */\npublic external open class XMLHttpRequest : XMLHttpRequestEventTarget {\n   var onreadystatechange: ((Event) -> dynamic)?\n   open val readyState: Short\n   var timeout: Int\n   var withCredentials: Boolean\n   open val upload: XMLHttpRequestUpload\n   open val responseURL: String\n   open val status: Short\n   open val statusText: String\n   var responseType: XMLHttpRequestResponseType\n   open val response: Any?\n   open val responseText: String\n   open val responseXML: Document?\n   fun open(method: String, url: String)\n   fun open(method: String, url: String, async: Boolean, username: String? = definedExternally, password: String? = definedExternally)\n   fun setRequestHeader(name: String, value: String)\n   fun send(body: dynamic = definedExternally)\n   fun abort()\n   fun getResponseHeader(name: String): String?\n   fun getAllResponseHeaders(): String\n   fun overrideMimeType(mime: String)\n\n   companion object {\n   val UNSENT: Short\n   val OPENED: Short\n   val HEADERS_RECEIVED: Short\n   val LOADING: Short\n   val DONE: Short\n   }\n}\n\n/**\n * Exposes the JavaScript [FormData](https://developer.mozilla.org/en/docs/Web/API/FormData) to Kotlin\n */\npublic external open class FormData(form: HTMLFormElement = definedExternally) {\n   fun append(name: String, value: String)\n   fun append(name: String, value: Blob, filename: String = definedExternally)\n   fun delete(name: String)\n   fun get(name: String): dynamic\n   fun getAll(name: String): Array<dynamic>\n   fun has(name: String): Boolean\n   fun set(name: String, value: String)\n   fun set(name: String, value: Blob, filename: String = definedExternally)\n}\n\n/**\n * Exposes the JavaScript [ProgressEvent](https://developer.mozilla.org/en/docs/Web/API/ProgressEvent) to Kotlin\n */\npublic external open class ProgressEvent(type: String, eventInitDict: ProgressEventInit = definedExternally) : Event {\n   open val lengthComputable: Boolean\n   open val loaded: Number\n   open val total: Number\n\n   companion object {\n   val NONE: Short\n   val CAPTURING_PHASE: Short\n   val AT_TARGET: Short\n   val BUBBLING_PHASE: Short\n   }\n}\n\npublic external interface ProgressEventInit : EventInit {\n   var lengthComputable: Boolean? /* = false */\n   get() = definedExternally\n   set(value) = definedExternally\n   var loaded: Number? /* = 0 */\n   get() = definedExternally\n   set(value) = definedExternally\n   var total: Number? /* = 0 */\n   get() = definedExternally\n   set(value) = definedExternally\n}\n\n@Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\n@kotlin.internal.InlineOnly\npublic inline fun ProgressEventInit(lengthComputable: Boolean? = false, loaded: Number? = 0, total: Number? = 0, bubbles: Boolean? = false, cancelable: Boolean? = false, composed: Boolean? = false): ProgressEventInit {\n   val o = js(\"({})\")\n   o[\"lengthComputable\"] = lengthComputable\n   o[\"loaded\"] = loaded\n   o[\"total\"] = total\n   o[\"bubbles\"] = bubbles\n   o[\"cancelable\"] = cancelable\n   o[\"composed\"] = composed\n   return o\n}\n\n/* please, don't implement this interface! */\n@JsName(\"null\")\n@Suppress(\"NESTED_CLASS_IN_EXTERNAL_INTERFACE\")\npublic external interface XMLHttpRequestResponseType {\n   companion object\n}\n\npublic inline val XMLHttpRequestResponseType.Companion.EMPTY: XMLHttpRequestResponseType get() = \"\".asDynamic().unsafeCast<XMLHttpRequestResponseType>()\n\npublic inline val XMLHttpRequestResponseType.Companion.ARRAYBUFFER: XMLHttpRequestResponseType get() = \"arraybuffer\".asDynamic().unsafeCast<XMLHttpRequestResponseType>()\n\npublic inline val XMLHttpRequestResponseType.Companion.BLOB: XMLHttpRequestResponseType get() = \"blob\".asDynamic().unsafeCast<XMLHttpRequestResponseType>()\n\npublic inline val XMLHttpRequestResponseType.Companion.DOCUMENT: XMLHttpRequestResponseType get() = \"document\".asDynamic().unsafeCast<XMLHttpRequestResponseType>()\n\npublic inline val XMLHttpRequestResponseType.Companion.JSON: XMLHttpRequestResponseType get() = \"json\".asDynamic().unsafeCast<XMLHttpRequestResponseType>()\n\npublic inline val XMLHttpRequestResponseType.Companion.TEXT: XMLHttpRequestResponseType get() =

\"text\".asDynamic().unsafeCast<XMLHttpRequestResponseType>()","package kotlin\n\nimport kotlin.annotation.AnnotationTarget.*\n\n/**\n * This annotation marks the standard library API that is considered experimental and is not subject to the\n * [general compatibility guarantees](https://kotlinlang.org/docs/reference/evolution/components-stability.html) given for the standard library:\n * the behavior of such API may be changed or the API may be removed completely in any further release.\n *\n * > Beware using the annotated API especially if you're developing a library, since your library might become binary incompatible\n * with the future versions of the standard library.\n *\n * Any usage of a declaration annotated with `@ExperimentalStdlibApi` must be accepted either by\n * annotating that usage with the [OptIn] annotation, e.g. `@OptIn(ExperimentalStdlibApi::class)`,\n * or by using the compiler argument `-opt-in=kotlin.ExperimentalStdlibApi`.\n */\n@RequiresOptIn(level = RequiresOptIn.Level.ERROR)\n@Retention(AnnotationRetention.BINARY)\n@Target(\n   CLASS,\n   ANNOTATION_CLASS,\n   PROPERTY,\n   FIELD,\n   LOCAL_VARIABLE,\n   VALUE_PARAMETER,\n   CONSTRUCTOR,\n   FUNCTION,\n   PROPERTY_GETTER,\n   PROPERTY_SETTER,\n   TYPEALIAS\n)\n@MustBeDocumented\n@SinceKotlin(\"1.3\")\npublic annotation class ExperimentalStdlibApi\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\nimport kotlin.annotation.AnnotationTarget.*\nimport kotlin.experimental.ExperimentalTypeInference\n\n/**\n * Allows to infer generic type arguments of a function from the calls in the annotated function parameter of that function.\n *\n * When this annotation is placed on a generic function parameter of a function,\n * it enables to infer the type arguments of that generic function from the lambda body passed to that parameter.\n *\n * The calls that affect inference are either members of the receiver type of an annotated function parameter or\n * extensions for that type. The extensions must be themselves annotated with `@BuilderInference`.\n *\n * Example: we declare\n * ```\n * fun <T> sequence(@BuilderInference block: suspend SequenceScope<T>.() -> Unit): Sequence<T>\n * ```\n * and use it like\n * ```\n * val result = sequence { yield(\"result\") }\n * ```\n * Here the type argument of the resulting sequence is inferred to `String` from\n * the argument of the [SequenceScope.yield] function, that is called inside the lambda passed to [sequence].\n *\n * Note: this annotation is experimental, see [ExperimentalTypeInference] on how to opt-in for it.\n */\n@Target(VALUE_PARAMETER, FUNCTION, PROPERTY)\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin(\"1.3\")\n@ExperimentalTypeInference\npublic annotation class BuilderInference\n\n\n/**\n * Enables overload selection based on the type of the value returned from lambda argument.\n *\n * When two or more function overloads have otherwise the same parameter lists that differ only in the return type\n * of a functional parameter, this annotation enables overload selection by the type of the value returned from\n * the lambda function passed to this functional parameter.\n *\n * Example:\n * ```\n * @OverloadResolutionByLambdaReturnType\n * fun create(intProducer: () -> Int): Int\n *\n * fun create(doubleProducer: () -> Double): Double\n *\n * val newValue = create { 3.14 }\n * ```\n *\n * The annotation being applied to one of overloads allows to resolve this ambiguity by analyzing what value is returned\n * from the lambda function.\n *\n * This annotation is also used to discriminate the annotated overloads in case if overload selection still cannot\n * choose one of them even taking in account the result of lambda parameter analysis. In that case a warning is reported.\n *\n * Note: this annotation is experimental, see [ExperimentalTypeInference] on how to opt-in for it.\n */\n@Target(FUNCTION)\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin(\"1.4\")\n@ExperimentalTypeInference\npublic annotation class OverloadResolutionByLambdaReturnType","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\nimport kotlin.annotation.AnnotationTarget.*\nimport kotlin.internal.RequireKotlin\nimport kotlin.internal.RequireKotlinVersionKind\n\n/**\n * The experimental multiplatform support API marker.\n *\n * Any usage of a declaration annotated with

`@ExperimentalMultiplatform` must be accepted either by\n * annotating that usage with the [OptIn] annotation, e.g. `@OptIn(ExperimentalMultiplatform::class)`,\n * or by using the compiler argument `-opt-in=kotlin.ExperimentalMultiplatform`.\n */\n@RequiresOptIn\n@MustBeDocumented\n@Target(\n CLASS,\n ANNOTATION_CLASS,\n PROPERTY,\n FIELD,\n LOCAL_VARIABLE,\n VALUE_PARAMETER,\n CONSTRUCTOR,\n FUNCTION,\n PROPERTY_GETTER,\n PROPERTY_SETTER,\n TYPEALIAS\n)\n@Retention(AnnotationRetention.BINARY)\npublic annotation class ExperimentalMultiplatform\n\n/**\n * Marks an expected annotation class that it isn't required to have actual counterparts in all platforms.\n *\n * This annotation is only applicable to `expect` annotation classes in multi-platform projects
and marks that class as \"optional\".\n * Optional expected class is allowed to have no corresponding actual class on the platform. Optional annotations can only be used\n * to annotate something, not as types in signatures. If an optional annotation has no corresponding actual class on a platform,\n * the annotation entries where it's used are simply erased when compiling code on that platform.\n *\n * Note: this annotation is experimental, see [ExperimentalMultiplatform] on how to opt-in for it.\n */\n@Target(ANNOTATION_CLASS)\n@Retention(AnnotationRetention.BINARY)\n@ExperimentalMultiplatform\npublic annotation class OptionalExpectation\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\nimport kotlin.annotation.AnnotationRetention.*\nimport kotlin.annotation.AnnotationTarget.*\nimport kotlin.reflect.KClass\n\n/**\n
* Signals that the annotated annotation class is a marker of an API that requires an explicit opt-in.\n *\n * Call sites of any declaration that is either annotated with such a marker or mentions in its signature any\n * other declaration that requires opt-in should opt in to the API either by using [OptIn],\n * or by being annotated with that marker themselves, effectively causing further propagation of the opt-in requirement.\n *\n * The intended uses of opt-in markers include, but are not limited to the following:\n * - Experimental API for public preview that might change its semantics or affect binary compatibility.\n * - Internal declarations that should not be used outside the declaring library, but are `public` for technical reasons.\n * - Fragile or delicate API that needs a lot of expertise to use and thus require an explicit opt-in.\n *\n * ## Contagiousness\n *\n * When a declaration is marked with an opt-in requirement, it is considered to be contagious,
meaning that all its uses\n * or mentions in other declarations will require an explicit opt-in.\n * A rule of thumb for propagating is the following: if the marked declaration ceases to exist, only\n * the places with explicit opt-in (or the corresponding warning) will break. This rule does not imply transitivity,\n * e.g. the propagation does not propagate opt-in through inlining, making it the responsibility `inline` function author\n * to mark it properly.\n *\n * ### Type scopes\n *\n * A type is considered requiring opt-in if it is marked with an opt-in marker, or the outer declaration (class or interface) requires opt-in.\n * Any use of any declaration that mentions such type in its signature will require an explicit opt-in, even if it is not used\n * directly on the call site, and even if such declarations do not require opt-in directly.\n *\n * For example, consider the following declarations that are marked with non-propagating opt-in:\n * ```\n * @UnstableApi\n * class
Unstable\n *\n * @OptIn(UnstableApi::class)\n * fun foo(): Unstable = Unstable()\n *\n * @OptIn(UnstableApi::class)\n * fun bar(arg: Unstable = Unstable()) {}\n *\n * @OptIn(UnstableApi::class)\n * fun Unstable?.baz() {}\n * ```\n * and their respective call sites:\n * ```\n * fun outerFun() {\n * val s = foo()\n * bar()\n * null.baz()\n * }\n * ```\n * Even though call sites do not mention `Unstable` type directly, the corresponding opt-in warning or error will be triggered\n * in each call site due to propagation contagiousness. Note that the propagation is not transitive, i.e. calls to `outerFun`\n * itself would not trigger any further opt-in requirements.\n *\n * ### Lexical scopes\n *\n * If a type requires an opt-in, such requirement is propagated to its lexical scope and all its nested declarations.\n * For example, for the following scope:\n * ```\n * @UnstableApi\n * class Unstable {\n * fun memberFun() = ...\n *\n * class NestedClass {\n * fun nestedFun()

= ...\n *   }\n * }\n * ```\n *\n * Any use of `Unstable`, `NestedClass`, or their member functions will require an explicit opt-in.\n *\n * ### Overridden declarations\n *\n * Opt-in markers are also propagated through the inheritance and interface implementation.\n * If the base declaration requires an opt-in, overriding it requires either an explicit opt-in or\n * propagating the opt-in requirement.\n *\n * See also [Kotlin language documentation](https://kotlinlang.org/docs/opt-in-requirements.html) for more information.\n *\n * @property message message to be reported on usages of API without an explicit opt-in, or empty string for the default message.\n *   The default message is: \"This declaration is experimental and its usage should be marked with 'Marker'\n *   or '@OptIn(Marker::class)'\", where `Marker` is the opt-in requirement marker.\n * @property level specifies how usages of API without an explicit opt-in are reported in code.\n */\n@Target(ANNOTATION_CLASS)\n@Retention(BINARY)\n@SinceKotlin(\"1.3\")\npublic annotation class RequiresOptIn(\n   val message: String = \"\",\n   val level: Level = Level.ERROR\n) {\n   /**\n   * Severity of the diagnostic that should be reported on usages which did not explicitly opted into\n   * the API either by using [OptIn] or by being annotated with the corresponding marker annotation.\n   */\n   public enum class Level {\n   /** Specifies that a warning should be reported on incorrect usages of this API. */\n   WARNING,\n\n   /** Specifies that a compilation error should be reported on incorrect usages of this API. */\n   ERROR,\n   }\n}\n\n/**\n * Allows to use the API denoted by the given markers in the annotated file, declaration, or expression.\n * If a declaration is annotated with [OptIn], its usages are **not** required to opt in to that API.\n *\n * [markerClass] specifies marker annotations that require explicit opt-in. The marker annotation is\n * not required to be itself marked with [RequiresOptIn] to enable gradual migration of API from requiring opt-in to the regular one,\n * yet declaring such `OptIn` yields a compilation warning.\n *\n * See also [Kotlin language documentation](https://kotlinlang.org/docs/opt-in-requirements.html) for more information.\n *\n * @property markerClass specifies marker annotations that require explicit opt-in.\n * @see RequiresOptIn for a detailed description of opt-in semantics and propagation rules.\n */\n@Target(\n   CLASS, PROPERTY, LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER, PROPERTY_SETTER, EXPRESSION, FILE, TYPEALIAS\n)\n@Retention(SOURCE)\n@SinceKotlin(\"1.3\")\npublic annotation class OptIn(\n   vararg val markerClass: KClass<out Annotation>\n)\n\n/**\n * This annotation marks the experimental preview of the language feature [SubclassOptInRequired].\n *\n * > Note that this API is in a preview state and has a chance of being changed in the future.\n * Do not use it if you develop a library since your library can become source incompatible\n * with the future versions of Kotlin.\n */\n@Target(CLASS)\n@Retention(BINARY)\n@SinceKotlin(\"1.8\")\npublic annotation class ExperimentalSubclassOptIn\n\n/**\n * Annotation that marks open for subclassing classes and interfaces, and makes implementation\n * and extension of such declarations as requiring an explicit opt-in.\n *\n * When applied, any attempt to subclass the target declaration will trigger an opt-in\n * with the corresponding level and message.\n *\n * The intended uses of subclass opt-in markers include, but are not limited to the following API:\n * - Stable to use, but unstable to implement due to its further evolution.\n * - Stable to use, but closed for 3rd-part implementations due to internal or technical reasons.\n * - Stable to use, but delicate or fragile to implement.\n * - Stable to use, but with a contract that may be weakened in the future in a backwards-incompatible\n *   manner for external implementations.\n *\n * Contrary to regular [RequiresOptIn], there are three ways to opt-in into the subclassing requirement:\n * - Annotate declaration with the marker annotation, making it propagating.\n * - Annotate declaration with [OptIn] in order to opt in into the provided guarantees in a non-propagating manner.\n * - Annotate declaration with [SubclassOptInRequired] with the same marker class, making it further propagating only for subclassing.\n *\n * Uses of this annotation are limited to open and abstract classes, and non-`fun` interfaces.\n * Any other uses allowed by `CLASS` annotation target yield a compilation error.\n *\n * @property markerClass specifies marker annotation that require explicit opt-in.\n * @see RequiresOptIn for a detailed description of opt-in semantics and propagation rules.\n */\n@Target(CLASS)\n@Retention(BINARY)\n@SinceKotlin(\"1.8\")\n@ExperimentalSubclassOptIn\npublic

annotation class SubclassOptInRequired(\n
val markerClass: KClass<out Annotation>\n)\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\nimport kotlin.annotation.AnnotationRetention.BINARY\nimport kotlin.annotation.AnnotationTarget.*\nimport kotlin.reflect.KClass\n\n\n@Target(CLASS, PROPERTY, CONSTRUCTOR, FUNCTION, TYPEALIAS)\n@Retention(BINARY)\ninternal annotation class WasExperimental(\n vararg val markerClass: KClass<out Annotation>\n)\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\npackage kotlin.collections\n\nimport kotlin.js.JsName\n\n/**\n * Provides a skeletal implementation of the read-only [Collection] interface.\n *\n * @param E the type of elements contained in the collection. The collection is covariant in its element type.\n */\n@SinceKotlin(\"1.1\")\npublic abstract class AbstractCollection<out E> protected constructor() : Collection<E> {\n abstract override val size: Int\n abstract override fun iterator(): Iterator<E>\n\n override fun contains(element: @UnsafeVariance E): Boolean = any { it == element }\n\n override fun containsAll(elements: Collection<@UnsafeVariance E>): Boolean =\n elements.all { contains(it) } // use when js will support bound refs: elements.all(this::contains)\n\n override fun isEmpty(): Boolean = size == 0\n\n override fun toString(): String = joinToString(\", \", \"[\", \"]\") {\n if (it === this) \"(this Collection)\" else it.toString()\n }\n\n /**\n * Returns new array of type `Array<Any?>` with the elements of this collection.\n */\n @JsName(\"toArray\")\n protected open fun toArray(): Array<Any?> = copyToArrayImpl(this)\n\n /**\n * Fills the provided [array] or creates new array of the same type\n * and fills it with the elements of this collection.\n */\n protected open fun <T> toArray(array: Array<T>): Array<T> = copyToArrayImpl(this, array)\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n\nprivate enum class State {\n Ready,\n NotReady,\n Done,\n Failed\n}\n\n/**\n * A base class to simplify implementing iterators so that implementations only have to implement [computeNext]\n * to implement the iterator, calling [done] when the iteration is complete.\n */\npublic abstract class AbstractIterator<T> : Iterator<T> {\n private var state = State.NotReady\n private var nextValue: T? = null\n\n override fun hasNext(): Boolean {\n require(state != State.Failed)\n return when (state) {\n State.Done -> false\n State.Ready -> true\n else -> tryToComputeNext()\n }\n }\n\n override fun next(): T {\n if (!hasNext()) throw NoSuchElementException()\n state = State.NotReady\n @Suppress(\"UNCHECKED_CAST\")\n return nextValue as T\n }\n\n private fun tryToComputeNext(): Boolean {\n state = State.Failed\n computeNext()\n return state == State.Ready\n }\n\n /**\n * Computes the next item in the iterator.\n *\n * This callback method should call one of these two methods:\n *\n * * [setNext] with the next value of the iteration\n * * [done] to indicate there are no more elements\n *\n * Failure to call either method will result in the iteration terminating with a failed state\n */\n abstract protected fun computeNext(): Unit\n\n /**\n * Sets the next value in the iteration, called from the [computeNext] function\n */\n protected fun setNext(value: T): Unit {\n nextValue = value\n state = State.Ready\n }\n\n /**\n * Sets the state to done so that the iteration terminates.\n */\n protected fun done() {\n state = State.Done\n }\n}\n\n\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n/*\n * Based on GWT AbstractList\n * Copyright 2007 Google Inc.\n*/\n\npackage kotlin.collections\n\n/**\n * Provides a skeletal implementation of the read-only [List] interface.\n *\n * This class is intended to help implementing read-only lists so it doesn't support concurrent modification tracking.\n *\n * @param E the type of elements contained in the list. The list is covariant in its element type.\n */\n@SinceKotlin(\"1.1\")\npublic abstract class AbstractList<out E> protected constructor() : AbstractCollection<E>(),

```
List<E> {\n    abstract override val size: Int\n    abstract override fun get(index: Int): E\n\n    override fun iterator(): Iterator<E> = IteratorImpl()\n\n    override fun indexOf(element: @UnsafeVariance E): Int = indexOfFirst { it == element }\n\n    override fun lastIndexOf(element: @UnsafeVariance E): Int = indexOfLast { it == element }\n\n    override fun listIterator(): ListIterator<E> = ListIteratorImpl(0)\n\n    override fun listIterator(index: Int): ListIterator<E> = ListIteratorImpl(index)\n\n    override fun subList(fromIndex: Int, toIndex: Int): List<E> = SubList(this, fromIndex, toIndex)\n\n    private class SubList<out E>(private val list: AbstractList<E>, private val fromIndex: Int, toIndex: Int) : AbstractList<E>(), RandomAccess {\n        private var _size: Int = 0\n\n        init {\n            checkRangeIndexes(fromIndex, toIndex, list.size)\n            this._size = toIndex - fromIndex\n        }\n\n        override fun get(index: Int): E {\n            checkElementIndex(index, _size)\n\n            return list[fromIndex + index]\n        }\n\n        override val size: Int get() = _size\n    }\n\n    /**\n     * Compares this list with other list instance with the ordered structural equality.\n     *\n     * @return true, if [other] instance is a [List] of the same size, which contains the same elements in the same order.\n     */\n    override fun equals(other: Any?): Boolean {\n        if (other === this) return true\n        if (other !is List<*>) return false\n\n        return orderedEquals(this, other)\n    }\n\n    /**\n     * Returns the hash code value for this list.\n     */\n    override fun hashCode(): Int = orderedHashCode(this)\n\n    private open inner class IteratorImpl : Iterator<E> {\n        /** the index of the item that will be returned on the next call to [next]`()` */\n        protected var index = 0\n\n        override fun hasNext(): Boolean = index < size\n\n        override fun next(): E {\n            if (!hasNext()) throw NoSuchElementException()\n            return get(index++)\n        }\n    }\n\n    /**\n     * Implementation of [ListIterator] for abstract lists.\n     */\n    private open inner class ListIteratorImpl(index: Int) : IteratorImpl(), ListIterator<E> {\n\n        init {\n            checkPositionIndex(index, this@AbstractList.size)\n            this.index = index\n        }\n\n        override fun hasPrevious(): Boolean = index > 0\n\n        override fun nextIndex(): Int = index\n\n        override fun previous(): E {\n            if (!hasPrevious()) throw NoSuchElementException()\n            return get(--index)\n        }\n\n        override fun previousIndex(): Int = index - 1\n    }\n\n    internal companion object {\n        internal fun checkElementIndex(index: Int, size: Int) {\n            if (index < 0 || index >= size) {\n                throw IndexOutOfBoundsException(\"index: $index, size: $size\")\n            }\n        }\n\n        internal fun checkPositionIndex(index: Int, size: Int) {\n            if (index < 0 || index > size) {\n                throw IndexOutOfBoundsException(\"index: $index, size: $size\")\n            }\n        }\n\n        internal fun checkRangeIndexes(fromIndex: Int, toIndex: Int, size: Int) {\n            if (fromIndex < 0 || toIndex > size) {\n                throw IndexOutOfBoundsException(\"fromIndex: $fromIndex, toIndex: $toIndex, size: $size\")\n            }\n            if (fromIndex > toIndex) {\n                throw IllegalArgumentException(\"fromIndex: $fromIndex > toIndex: $toIndex\")\n            }\n        }\n\n        internal fun checkBoundsIndexes(startIndex: Int, endIndex: Int, size: Int) {\n            if (startIndex < 0 || endIndex > size) {\n                throw IndexOutOfBoundsException(\"startIndex: $startIndex, endIndex: $endIndex, size: $size\")\n            }\n            if (startIndex > endIndex) {\n                throw IllegalArgumentException(\"startIndex: $startIndex > endIndex: $endIndex\")\n            }\n        }\n\n        internal fun orderedHashCode(c: Collection<*>): Int {\n            var hashCode = 1\n            for (e in c) {\n                hashCode = 31 * hashCode + (e?.hashCode() ?: 0)\n            }\n            return hashCode\n        }\n\n        internal fun orderedEquals(c: Collection<*>, other: Collection<*>): Boolean {\n            if (c.size != other.size) return false\n\n            val otherIterator = other.iterator()\n            for (elem in c) {\n                val elemOther = otherIterator.next()\n                if (elem != elemOther) {\n                    return false\n                }\n            }\n            return true\n        }\n    }\n}","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n/*\n * Based on GWT AbstractMap\n * Copyright 2007 Google Inc.\n */\n\npackage kotlin.collections\n\n/**\n * Provides a skeletal implementation of the read-only [Map] interface.\n *\n * The implementor is required to implement [entries] property, which should return read-only set of map entries.\n *\n * @param K the type of map keys. The map is invariant in its key type.\n * @param V the type of map values. The map is covariant in its value type.\n */\n@SinceKotlin(\"1.1\")\npublic abstract class AbstractMap<K, out V>
```

```
protected constructor() : Map<K, V> {\n\n    override fun containsKey(key: K): Boolean {\n        return implFindEntry(key) != null\n    }\n\n    override fun containsValue(value: @UnsafeVariance V): Boolean = entries.any { it.value == value }\n\n    internal fun containsEntry(entry: Map.Entry<*, *>?): Boolean {\n        // since entry comes from @UnsafeVariance parameters it can be virtually anything\n        if (entry !is Map.Entry<*, *>) return false\n        val key = entry.key\n        val value = entry.value\n        val ourValue = get(key)\n\n        if (value != ourValue) {\n            return false\n        }\n\n        // Perhaps it was null and we don't contain the key?\n        if (ourValue == null && !containsKey(key)) {\n            return false\n        }\n\n        return true\n    }\n\n\n    /**\n     * Compares this map with other instance with the ordered structural equality.\n     *\n     * @return true, if [other] instance is a [Map] of the same size, all entries of which are contained in the [entries] set of this map.\n     */\n    override fun equals(other: Any?): Boolean {\n        if (other === this) return true\n        if (other !is Map<*, *>) return false\n        if (size != other.size) return false\n\n        return other.entries.all { containsEntry(it) }\n    }\n\n    override operator fun get(key: K): V? = implFindEntry(key)?.value\n\n\n    /**\n     * Returns the hash code value for this map.\n     *\n     * It is the same as the hashCode of [entries] set.\n     */\n    override fun hashCode():
 Int = entries.hashCode()\n\n    override fun isEmpty(): Boolean = size == 0\n    override val size: Int get() = entries.size\n\n    /**\n     * Returns a read-only [Set] of all keys in this map.\n     *\n     * Accessing this property first time creates a keys view from [entries].\n     * All subsequent accesses just return the created instance.\n     */\n
 override val keys: Set<K>\n        get() {\n            if (_keys == null) {\n                _keys = object : AbstractSet<K>() {\n                    override operator fun contains(element: K): Boolean = containsKey(element)\n\n
override operator fun iterator(): Iterator<K> {\n                        val entryIterator = entries.iterator()\n
return object : Iterator<K> {\n                            override fun hasNext(): Boolean = entryIterator.hasNext()\n
            override fun next(): K = entryIterator.next().key\n                        }\n                    }\n\n
                    override val size: Int get() = this@AbstractMap.size\n                }\n            }\n            return _keys!!\n
  }\n\n    @kotlin.jvm.Volatile\n    private var _keys: Set<K>? = null\n\n\n    override fun toString(): String = entries.joinToString(\", \", \"{\", \"}\") { toString(it) }\n\n    private fun toString(entry: Map.Entry<K, V>): String = toString(entry.key) + \"=\" + toString(entry.value)\n\n    private fun toString(o: Any?): String = if (o === this) \"(this Map)\" else o.toString()\n\n    /**\n     * Returns a read-only [Collection] of all values in this map.\n     *\n     * Accessing this property first time creates a values view from [entries].\n     * All subsequent accesses just return the created instance.\n     */\n    override val values: Collection<V>\n        get() {\n            if (_values == null) {\n
  _values = object : AbstractCollection<V>() {\n                    override operator fun contains(element: @UnsafeVariance
 V): Boolean = containsValue(element)\n\n                    override operator fun iterator(): Iterator<V> {\n
 val entryIterator = entries.iterator()\n                        return object : Iterator<V> {\n                            override fun hasNext(): Boolean = entryIterator.hasNext()\n                            override fun next(): V = entryIterator.next().value\n                        }\n                    }\n\n                    override val size: Int get() = this@AbstractMap.size\n                }\n
    }\n            return _values!!\n        }\n\n    @kotlin.jvm.Volatile\n    private var _values: Collection<V>? = null\n\n
  private fun implFindEntry(key: K): Map.Entry<K, V>? = entries.firstOrNull { it.key == key }\n\n    internal companion object {\n\n        internal fun entryHashCode(e: Map.Entry<*, *>): Int = with(e) { (key?.hashCode() ?: 0) xor (value?.hashCode() ?: 0) }\n        internal fun entryToString(e: Map.Entry<*, *>): String
 = with(e) { \"$key=$value\" }\n        internal fun entryEquals(e: Map.Entry<*, *>, other: Any?): Boolean {\n
if (other !is Map.Entry<*, *>) return false\n            return e.key == other.key && e.value == other.value\n        }\n
}\n}\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\npackage kotlin.collections\n\n/**\n * Provides a skeletal implementation of the read-only [Set] interface.\n *\n * This class is intended to help implementing read-only sets so it doesn't support concurrent modification tracking.\n
*\n * @param E the type of elements contained in the set. The set is covariant in its element type.\n
*/\n@SinceKotlin(\"1.1\")\npublic abstract class AbstractSet<out E> protected constructor() :
AbstractCollection<E>(), Set<E> {\n\n    /**\n     * Compares this set with other set instance with the
```

```
 unordered structural equality.\n     *\n     * @return true, if [other] instance is a [Set] of the same size, all elements of which are contained in this set.\n     */\n    override fun equals(other: Any?): Boolean {\n        if (other === this) return true\n        if (other !is Set<*>) return false\n        return setEquals(this, other)\n    }\n\n    /**\n     * Returns the hash code value for this set.\n     */\n    override fun hashCode(): Int = unorderedHashCode(this)\n\n    internal companion object {\n        internal fun unorderedHashCode(c: Collection<*>): Int {\n            var hashCode = 0\n            for (element in c) {\n                hashCode += (element?.hashCode() ?: 0)\n            }\n            return hashCode\n        }\n\n        internal fun setEquals(c: Set<*>, other: Set<*>): Boolean {\n            if (c.size != other.size) return false\n            return c.containsAll(other)\n        }\n    }\n}","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n/**\n * Resizable-array implementation of the deque data structure.\n *\n * The name deque is short for \"double ended queue\" and is usually pronounced \"deck\".\n *\n * The collection provide methods for convenient access to the both ends.\n * It also implements [MutableList] interface and supports efficient get/set operations by index.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic class ArrayDeque<E> : AbstractMutableList<E> {\n    private var head: Int = 0\n    private var elementData: Array<Any?>\n\n    override var size: Int = 0\n        private set\n\n    /**\n     * Constructs an empty deque with specified [initialCapacity], or throws [IllegalArgumentException] if [initialCapacity] is negative.\n     */\n    public constructor(initialCapacity: Int) {\n        elementData = when {\n            initialCapacity == 0 -> emptyElementData\n            initialCapacity > 0 -> arrayOfNulls(initialCapacity)\n            else -> throw IllegalArgumentException(\"Illegal Capacity: $initialCapacity\")\n        }\n    }\n\n    /**\n     * Constructs an empty deque.\n     */\n    public constructor() {\n        elementData = emptyElementData\n    }\n\n    /**\n     * Constructs a deque that contains the same elements as the specified [elements] collection in the same order.\n     */\n    public constructor(elements: Collection<E>) {\n        elementData = elements.toTypedArray()\n        size = elementData.size\n        if (elementData.isEmpty())\n            elementData = emptyElementData\n    }\n\n    /**\n     * Ensures that the capacity of this deque is at least equal to the specified [minCapacity].\n     *\n     * If the current capacity is less than the [minCapacity], a new backing storage is allocated with greater capacity.\n     * Otherwise, this method takes no action and simply returns.\n     */\n    private fun ensureCapacity(minCapacity: Int) {\n        if (minCapacity < 0) throw IllegalStateException(\"Deque is too big.\")  // overflow\n        if (minCapacity <= elementData.size) return\n        if (elementData === emptyElementData) {\n            elementData = arrayOfNulls(minCapacity.coerceAtLeast(defaultMinCapacity))\n            return\n        }\n\n        val newCapacity = newCapacity(elementData.size, minCapacity)\n        copyElements(newCapacity)\n    }\n\n    /**\n     * Creates a new array with the specified [newCapacity] size and copies elements in the [elementData] array to it.\n     */\n    private fun copyElements(newCapacity: Int) {\n        val newElements = arrayOfNulls<Any?>(newCapacity)\n        elementData.copyInto(newElements, 0, head, elementData.size)\n        elementData.copyInto(newElements, elementData.size - head, 0, head)\n        head = 0\n        elementData = newElements\n    }\n\n    @kotlin.internal.InlineOnly\n    private inline fun internalGet(internalIndex: Int): E {\n        @Suppress(\"UNCHECKED_CAST\")\n        return elementData[internalIndex] as E\n    }\n\n    private fun positiveMod(index: Int): Int = if (index >= elementData.size) index - elementData.size else index\n\n    private fun negativeMod(index: Int): Int = if (index < 0) index + elementData.size else index\n\n    @kotlin.internal.InlineOnly\n    private inline fun internalIndex(index: Int): Int = positiveMod(head + index)\n\n    private fun incremented(index: Int): Int = if (index == elementData.lastIndex) 0 else index + 1\n\n    private fun decremented(index: Int): Int = if (index == 0) elementData.lastIndex else index - 1\n\n    override fun isEmpty(): Boolean = size == 0\n\n    /**\n     * Returns the first element, or throws [NoSuchElementException] if this deque is empty.\n     */\n    public fun first(): E = if (isEmpty()) throw NoSuchElementException(\"ArrayDeque is empty.\") else internalGet(head)\n\n    /**\n     * Returns the first element, or `null` if this deque is empty.\n     */\n    public fun firstOrNull(): E? = if (isEmpty()) null else internalGet(head)\n\n    /**\n     * Returns the last element, or throws
```

```
[NoSuchElementException] if this deque is empty.\n     */\n    public fun last(): E = if (isEmpty()) throw
NoSuchElementException(\"ArrayDeque is empty.\") else internalGet(internalIndex(lastIndex))\n\n    /**\n     *
Returns the last element, or `null` if this deque is empty.\n     */\n    public fun lastOrNull(): E? = if (isEmpty()) null
else internalGet(internalIndex(lastIndex))\n\n    /**\n     * Prepends the specified [element] to this deque.\n     */\n
public fun addFirst(element: E) {\n        ensureCapacity(size + 1)\n\n        head = decremented(head)\n
elementData[head] = element\n        size += 1\n    }\n\n    /**\n     * Appends the specified [element] to this deque.\n
 */\n    public fun addLast(element: E) {\n        ensureCapacity(size +
1)\n\n        elementData[internalIndex(size)] = element\n        size += 1\n    }\n\n    /**\n     * Removes the first
element from this deque and returns that removed element, or throws [NoSuchElementException] if this deque is
empty.\n     */\n    public fun removeFirst(): E {\n        if (isEmpty()) throw NoSuchElementException(\"ArrayDeque
is empty.\")\n\n        val element = internalGet(head)\n        elementData[head] = null\n        head =
incremented(head)\n        size -= 1\n        return element\n    }\n\n    /**\n     * Removes the first element from this
deque and returns that removed element, or returns `null` if this deque is empty.\n     */\n    public fun
removeFirstOrNull(): E? = if (isEmpty()) null else removeFirst()\n\n    /**\n     * Removes the last element from this
deque and returns that removed element, or throws [NoSuchElementException] if this deque is empty.\n     */\n
public fun removeLast(): E {\n        if (isEmpty()) throw NoSuchElementException(\"ArrayDeque
 is empty.\")\n\n        val internalLastIndex = internalIndex(lastIndex)\n        val element =
internalGet(internalLastIndex)\n        elementData[internalLastIndex] = null\n        size -= 1\n        return element\n
}\n\n    /**\n     * Removes the last element from this deque and returns that removed element, or returns `null` if
this deque is empty.\n     */\n    public fun removeLastOrNull(): E? = if (isEmpty()) null else removeLast()\n\n    //
MutableList, MutableCollection\n    public override fun add(element: E): Boolean {\n        addLast(element)\n
return true\n    }\n\n    public override fun add(index: Int, element: E) {\n
AbstractList.checkPositionIndex(index, size)\n\n        if (index == size) {\n            addLast(element)\n
return\n        } else if (index == 0) {\n            addFirst(element)\n            return\n        }\n\n        ensureCapacity(size
+ 1)\n\n        // Elements in circular array lay in 2 ways:\n        //
 1. `head` is less than `tail`:      [#, #, e1, e2, e3, #]\n        //   2. `head` is greater than `tail`:    [e3, #, #, #, e1, e2]\n
 // where head is the index of the first element in the circular array,\n        // and tail is the index following the last
element.\n        //\n        // At this point the insertion index is not equal to head or tail.\n        // Also the circular array
can store at least one more element.\n        //\n        // Depending on where the given element must be inserted the
preceding or the succeeding\n        // elements will be shifted to make room for the element to be inserted.\n        //\n
    // In case the preceding elements are shifted:\n        //   * if the insertion index is greater than the head (regardless
of circular array form)\n        //     -> shift the preceding elements\n        //   * otherwise, the circular array has (2)
form and the insertion index is less than tail\n        //     -> shift all elements in the back of
 the array\n        //     -> shift preceding elements in the front of the array\n        // In case the succeeding elements
are shifted:\n        //   * if the insertion index is less than the tail (regardless of circular array form)\n        //     ->
shift the succeeding elements\n        //   * otherwise, the circular array has (2) form and the insertion index is greater
than head\n        //     -> shift all elements in the front of the array\n        //     -> shift succeeding elements in the
back of the array\n\n        val internalIndex = internalIndex(index)\n\n        if (index < (size + 1) shr 1) {\n            //
closer to the first element -> shift preceding elements\n            val decrementedInternalIndex =
decremented(internalIndex)\n            val decrementedHead = decremented(head)\n\n            if
(decrementedInternalIndex >= head) {\n                elementData[decrementedHead] = elementData[head]  // head can
be zero\n                elementData.copyInto(elementData,
 head, head + 1, decrementedInternalIndex + 1)\n            } else { // head > tail\n
elementData.copyInto(elementData, head - 1, head, elementData.size) // head can't be zero\n
elementData[elementData.size - 1] = elementData[0]\n                elementData.copyInto(elementData, 0, 1,
decrementedInternalIndex + 1)\n            }\n\n            elementData[decrementedInternalIndex] = element\n
head = decrementedHead\n        } else {\n            // closer to the last element -> shift succeeding elements\n
val tail = internalIndex(size)\n\n            if (internalIndex < tail) {\n                elementData.copyInto(elementData,
```

```
internalIndex + 1, internalIndex, tail)\n            } else { // head > tail\n                elementData.copyInto(elementData,
1, 0, tail)\n                elementData[0] = elementData[elementData.size - 1]\n
elementData.copyInto(elementData, internalIndex + 1, internalIndex, elementData.size
 - 1)\n            }\n\n            elementData[internalIndex] = element\n        }\n        size += 1\n    }\n\n    private fun
copyCollectionElements(internalIndex: Int, elements: Collection<E>) {\n        val iterator = elements.iterator()\n\n
  for (index in internalIndex until elementData.size) {\n            if (!iterator.hasNext()) break\n
elementData[index] = iterator.next()\n        }\n        for (index in 0 until head) {\n            if (!iterator.hasNext())
break\n            elementData[index] = iterator.next()\n        }\n\n        size += elements.size\n    }\n\n    public override
fun addAll(elements: Collection<E>): Boolean {\n        if (elements.isEmpty()) return false\n
ensureCapacity(this.size + elements.size)\n        copyCollectionElements(internalIndex(size), elements)\n        return
true\n    }\n\n    public override fun addAll(index: Int, elements: Collection<E>): Boolean {\n
AbstractList.checkPositionIndex(index, size)\n\n
        if (elements.isEmpty()) {\n            return false\n        } else if (index == size) {\n            return
addAll(elements)\n        }\n\n        ensureCapacity(this.size + elements.size)\n\n        val tail = internalIndex(size)\n
  val internalIndex = internalIndex(index)\n        val elementsSize = elements.size\n\n        if (index < (size + 1) shr
1) {\n            // closer to the first element -> shift preceding elements\n\n            var shiftedHead = head -
elementsSize\n\n            if (internalIndex >= head) {\n                if (shiftedHead >= 0) {\n
elementData.copyInto(elementData, shiftedHead, head, internalIndex)\n                } else { // head < tail, insertion
leads to head >= tail\n                    shiftedHead += elementData.size\n                    val elementsToShift =
internalIndex - head\n                    val shiftToBack = elementData.size - shiftedHead\n\n                    if (shiftToBack
>= elementsToShift) {\n
                        elementData.copyInto(elementData, shiftedHead, head, internalIndex)\n                    } else {\n
      elementData.copyInto(elementData, shiftedHead, head, head + shiftToBack)\n
elementData.copyInto(elementData, 0, head + shiftToBack, internalIndex)\n                    }\n                }\n            }
else { // head > tail, internalIndex < tail\n                elementData.copyInto(elementData, shiftedHead, head,
elementData.size)\n                if (elementsSize >= internalIndex) {\n                    elementData.copyInto(elementData,
elementData.size - elementsSize, 0, internalIndex)\n                } else {\n
elementData.copyInto(elementData, elementData.size - elementsSize, 0, elementsSize)\n
elementData.copyInto(elementData, 0, elementsSize, internalIndex)\n                }\n            }\n            head =
shiftedHead\n            copyCollectionElements(negativeMod(internalIndex
 - elementsSize), elements)\n        } else {\n            // closer to the last element -> shift succeeding elements\n\n
 val shiftedInternalIndex = internalIndex + elementsSize\n\n            if (internalIndex < tail) {\n                if (tail +
elementsSize <= elementData.size) {\n                    elementData.copyInto(elementData, shiftedInternalIndex,
internalIndex, tail)\n                } else { // head < tail, insertion leads to head >= tail\n                    if
(shiftedInternalIndex >= elementData.size) {\n                        elementData.copyInto(elementData,
shiftedInternalIndex - elementData.size, internalIndex, tail)\n                    } else {\n                        val shiftToFront =
tail + elementsSize - elementData.size\n                        elementData.copyInto(elementData, 0, tail - shiftToFront,
tail)\n                        elementData.copyInto(elementData, shiftedInternalIndex, internalIndex, tail - shiftToFront)\n

    }\n                }\n            } else { // head > tail, internalIndex > head\n
elementData.copyInto(elementData, elementsSize, 0, tail)\n                if (shiftedInternalIndex >= elementData.size)
{\n                    elementData.copyInto(elementData, shiftedInternalIndex - elementData.size, internalIndex,
elementData.size)\n                } else {\n                    elementData.copyInto(elementData, 0, elementData.size -
elementsSize, elementData.size)\n                    elementData.copyInto(elementData, shiftedInternalIndex,
internalIndex, elementData.size - elementsSize)\n                }\n            }\n
copyCollectionElements(internalIndex, elements)\n        }\n\n        return true\n    }\n\n    public override fun
get(index: Int): E {\n        AbstractList.checkElementIndex(index, size)\n\n        return
internalGet(internalIndex(index))\n    }\n\n    public override fun set(index: Int, element: E): E {\n
```

```
AbstractList.checkElementIndex(index,
 size)\n\n        val internalIndex = internalIndex(index)\n        val oldElement = internalGet(internalIndex)\n
elementData[internalIndex] = element\n\n        return oldElement\n    }\n\n    public override fun contains(element:
E): Boolean = indexOf(element) != -1\n\n    public override fun indexOf(element: E): Int {\n        val tail =
internalIndex(size)\n\n        if (head < tail) {\n            for (index in head until tail) {\n                if (element ==
elementData[index]) return index - head\n            }\n        } else if (head >= tail) {\n            for (index in head until
elementData.size) {\n                if (element == elementData[index]) return index - head\n            }\n            for
(index in 0 until tail) {\n                if (element == elementData[index]) return index + elementData.size - head\n
  }\n        }\n\n        return -1\n    }\n\n    public override fun lastIndexOf(element: E): Int {\n        val tail =
internalIndex(size)\n\n
        if (head < tail) {\n            for (index in tail - 1 downTo head) {\n                if (element == elementData[index])
return index - head\n            }\n        } else if (head > tail) {\n            for (index in tail - 1 downTo 0) {\n                if
(element == elementData[index]) return index + elementData.size - head\n            }\n            for (index in
elementData.lastIndex downTo head) {\n                if (element == elementData[index]) return index - head\n
}\n        }\n\n        return -1\n    }\n\n    public override fun remove(element: E): Boolean {\n        val index =
indexOf(element)\n        if (index == -1) return false\n        removeAt(index)\n        return true\n    }\n\n    public
override fun removeAt(index: Int): E {\n        AbstractList.checkElementIndex(index, size)\n\n        if (index ==
lastIndex) {\n            return removeLast()\n        } else if (index == 0) {\n            return removeFirst()\n
  }\n\n        val internalIndex = internalIndex(index)\n        val element = internalGet(internalIndex)\n\n        if
(index < size shr 1) {\n            // closer to the first element -> shift preceding elements\n            if (internalIndex >=
head) {\n                elementData.copyInto(elementData, head + 1, head, internalIndex)\n            } else { // head >
tail, internalIndex < head\n                elementData.copyInto(elementData, 1, 0, internalIndex)\n
elementData[0] = elementData[elementData.size - 1]\n                elementData.copyInto(elementData, head + 1,
head, elementData.size - 1)\n            }\n\n            elementData[head] = null\n            head = incremented(head)\n
} else {\n            // closer to the last element -> shift succeeding elements\n            val internalLastIndex =
internalIndex(lastIndex)\n\n            if (internalIndex <= internalLastIndex) {\n
elementData.copyInto(elementData, internalIndex, internalIndex
 + 1, internalLastIndex + 1)\n            } else { // head > tail, internalIndex > head\n
elementData.copyInto(elementData, internalIndex, internalIndex + 1, elementData.size)\n
elementData[elementData.size - 1] = elementData[0]\n                elementData.copyInto(elementData, 0, 1,
internalLastIndex + 1)\n            }\n\n            elementData[internalLastIndex] = null\n        }\n        size -= 1\n\n
return element\n    }\n\n    public override fun removeAll(elements: Collection<E>): Boolean = filterInPlace {
!elements.contains(it) }\n\n    public override fun retainAll(elements: Collection<E>): Boolean = filterInPlace {
elements.contains(it) }\n\n    private inline fun filterInPlace(predicate: (E) -> Boolean): Boolean {\n        if
(this.isEmpty() || elementData.isEmpty())\n            return false\n\n        val tail = internalIndex(size)\n        var
newTail = head\n        var modified = false\n\n        if (head < tail) {\n
      for (index in head until tail) {\n                val element = elementData[index]\n\n
@Suppress(\"UNCHECKED_CAST\")\n                if (predicate(element as E))\n
elementData[newTail++] = element\n                else\n                    modified = true\n            }\n\n
elementData.fill(null, newTail, tail)\n\n        } else {\n            for (index in head until elementData.size) {\n
val element = elementData[index]\n                elementData[index] = null\n\n
@Suppress(\"UNCHECKED_CAST\")\n                if (predicate(element as E))\n
elementData[newTail++] = element\n                else\n                    modified = true\n            }\n\n            newTail =
positiveMod(newTail)\n\n            for (index in 0 until tail) {\n                val element = elementData[index]\n
elementData[index] = null\n\n                @Suppress(\"UNCHECKED_CAST\")\n
        if (predicate(element as E)) {\n                    elementData[newTail] = element\n                    newTail =
incremented(newTail)\n                } else {\n                    modified = true\n                }\n            }\n        }\n        if
(modified)\n            size = negativeMod(newTail - head)\n\n        return modified\n    }\n\n    public override fun
```

clear() {\n val tail = internalIndex(size)\n if (head < tail) {\n elementData.fill(null, head, tail)\n } else if (isNotEmpty()) {\n elementData.fill(null, head, elementData.size)\n elementData.fill(null, 0, tail)\n }\n head = 0\n size = 0\n }\n\n @Suppress(\"NOTHING_TO_OVERRIDE\")\n override fun <T> toArray(array: Array<T>): Array<T> {\n @Suppress(\"UNCHECKED_CAST\")\n val dest = (if (array.size >= size) array else arrayOfNulls(array, size)) as Array<Any?>\n\n val tail = internalIndex(size)\n if (head
< tail) {\n elementData.copyInto(dest, startIndex = head, endIndex = tail)\n } else if (isNotEmpty()) {\n elementData.copyInto(dest, destinationOffset = 0, startIndex = head, endIndex = elementData.size)\n elementData.copyInto(dest, destinationOffset = elementData.size - head, startIndex = 0, endIndex = tail)\n }\n if (dest.size > size) {\n dest[size] = null // null-terminate\n }\n\n @Suppress(\"UNCHECKED_CAST\")\n return dest as Array<T>\n }\n\n @Suppress(\"NOTHING_TO_OVERRIDE\")\n override fun toArray(): Array<Any?> {\n return toArray(arrayOfNulls<Any?>(size))\n }\n\n // for testing\n internal fun <T> testToArray(array: Array<T>): Array<T> = toArray(array)\n internal fun testToArray(): Array<Any?> = toArray()\n\n internal companion object {\n private val emptyElementData = emptyArray<Any?>()\n private const val maxArraySize = Int.MAX_VALUE - 8\n
private const val defaultMinCapacity = 10\n\n internal fun newCapacity(oldCapacity: Int, minCapacity: Int): Int {\n // overflow-conscious\n var newCapacity = oldCapacity + (oldCapacity shr 1)\n if (newCapacity - minCapacity < 0)\n newCapacity = minCapacity\n if (newCapacity - maxArraySize > 0)\n newCapacity = if (minCapacity > maxArraySize) Int.MAX_VALUE else maxArraySize\n return newCapacity\n }\n }\n\n // For testing only\n internal fun internalStructure(structure: (head: Int, elements: Array<Any?>) -> Unit) {\n val tail = internalIndex(size)\n val head = if (isEmpty() || head < tail) head else head - elementData.size\n structure(head, toArray())\n }\n}","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt
file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"ArraysKt\")\n\n\npackage kotlin.collections\n\nimport kotlin.contracts.*\n\n\n/**\n * Returns a single list of all elements from all arrays in the given array.\n * @sample samples.collections.Arrays.Transformations.flattenArray\n */\npublic fun <T> Array<out Array<out T>>.flatten(): List<T> {\n val result = ArrayList<T>(sumOf { it.size })\n for (element in this) {\n result.addAll(element)\n }\n return result\n}\n\n/**\n * Returns a pair of lists, where\n * *first* list is built from the first values of each pair from this array,\n * *second* list is built from the second values of each pair from this array.\n * @sample samples.collections.Arrays.Transformations.unzipArray\n */\npublic fun <T, R> Array<out Pair<T, R>>.unzip(): Pair<List<T>, List<R>> {\n val listT = ArrayList<T>(size)\n val listR = ArrayList<R>(size)\n for (pair in this) {\n listT.add(pair.first)\n listR.add(pair.second)\n }\n return listT to listR\n}\n\n/**\n * Returns `true` if this nullable array is either null or empty.\n * @sample samples.collections.Arrays.Usage.arrayIsNullOrEmpty\n
/\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline fun Array<>?.isNullOrEmpty(): Boolean {\n contract {\n returns(false) implies (this@isNullOrEmpty != null)\n }\n\n return this == null || this.isEmpty()\n}\n\n/**\n * Returns this array if it's not empty\n * or the result of calling [defaultValue] function if the array is empty.\n *\n * @sample samples.collections.Arrays.Usage.arrayIfEmpty\n
/\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\n@Suppress(\"UPPER_BOUND_CANNOT_BE_ARRAY\")\npublic inline fun <C, R> C.ifEmpty(defaultValue: () -> R): R where C : Array<>, C : R =\n if (isEmpty()) defaultValue() else
this\n\n\n@OptIn(ExperimentalUnsignedTypes::class)\n@SinceKotlin(\"1.3\")\n@PublishedApi\n@kotlin.jvm.JvmName(\"contentDeepEquals\")\n@kotlin.js.JsName(\"contentDeepEqualsImpl\")\ninternal
fun <T> Array<out T>?.contentDeepEqualsImpl(other: Array<out T>?): Boolean {\n if (this === other) return true\n if (this == null || other == null || this.size != other.size) return false\n\n for (i in indices) {\n val v1 = this[i]\n val v2 = other[i]\n\n if (v1 === v2) {\n continue\n } else if (v1 == null || v2 == null) {\n

```
return false\n        }\n\n        when {\n            v1 is Array<*>     && v2 is Array<*>     -> if
(!v1.contentDeepEquals(v2)) return false\n            v1 is ByteArray    && v2 is ByteArray    -> if
(!v1.contentEquals(v2)) return false\n            v1 is ShortArray   && v2 is ShortArray   -> if (!v1.contentEquals(v2))
return false\n            v1 is IntArray     && v2 is IntArray     -> if (!v1.contentEquals(v2)) return false\n            v1 is
LongArray    && v2 is LongArray    -> if (!v1.contentEquals(v2)) return false\n            v1 is FloatArray   && v2 is
 FloatArray   -> if (!v1.contentEquals(v2)) return false\n            v1 is DoubleArray  && v2 is DoubleArray  -> if
(!v1.contentEquals(v2)) return false\n            v1 is CharArray    && v2 is CharArray    -> if (!v1.contentEquals(v2))
return false\n            v1 is BooleanArray && v2 is BooleanArray -> if (!v1.contentEquals(v2)) return false\n\n
v1 is UByteArray   && v2 is UByteArray   -> if (!v1.contentEquals(v2)) return false\n            v1 is UShortArray
&& v2 is UShortArray  -> if (!v1.contentEquals(v2)) return false\n            v1 is UIntArray    && v2 is UIntArray    -
> if (!v1.contentEquals(v2)) return false\n            v1 is ULongArray   && v2 is ULongArray   -> if
(!v1.contentEquals(v2)) return false\n\n            else -> if (v1 != v2) return false\n        }\n\n    }\n    return
true\n}\n\n@SinceKotlin(\"1.3\")\n@PublishedApi\n@kotlin.jvm.JvmName(\"contentDeepToString\")\n@kotlin.js.
JsName(\"contentDeepToStringImpl\")\ninternal fun <T> Array<out T>?.contentDeepToStringImpl():
 String {\n    if (this == null) return \"null\"\n    val length = size.coerceAtMost((Int.MAX_VALUE - 2) / 5) * 5 + 2
// in order not to overflow Int.MAX_VALUE\n    return buildString(length) {\n
contentDeepToStringInternal(this, mutableListOf())\n
}\n}\n\n@OptIn(ExperimentalUnsignedTypes::class)\nprivate fun <T> Array<out
T>.contentDeepToStringInternal(result: StringBuilder, processed: MutableList<Array<*>>) {\n    if (this in
processed) {\n        result.append(\"[...]\")\n        return\n    }\n    processed.add(this)\n    result.append('[')\n\n    for (i
in indices) {\n        if (i != 0) {\n            result.append(\", \")\n        }\n        val element = this[i]\n        when
(element) {\n            null            -> result.append(\"null\")\n            is Array<*>     ->
element.contentDeepToStringInternal(result, processed)\n            is ByteArray    ->
result.append(element.contentToString())\n            is ShortArray   -> result.append(element.contentToString())\n
            is IntArray     -> result.append(element.contentToString())\n            is LongArray    ->
result.append(element.contentToString())\n            is FloatArray   -> result.append(element.contentToString())\n
  is DoubleArray  -> result.append(element.contentToString())\n            is CharArray    ->
result.append(element.contentToString())\n            is BooleanArray -> result.append(element.contentToString())\n\n
        is UByteArray   -> result.append(element.contentToString())\n            is UShortArray  ->
result.append(element.contentToString())\n            is UIntArray    -> result.append(element.contentToString())\n
  is ULongArray   -> result.append(element.contentToString())\n\n            else            ->
result.append(element.toString())\n        }\n    }\n\n    result.append(']')\n
processed.removeAt(processed.lastIndex)\n}","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n
 * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\npackage kotlin.collections\n\n/**\n * Data class representing a value from a collection or sequence, along with
its index in that collection or sequence.\n *\n * @property value the underlying value.\n * @property index the
index of the value in the collection or sequence.\n */\npublic data class IndexedValue<out T>(public val index: Int,
public val value: T)\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmName(\"MapAccessorsKt\")\n\npackage
kotlin.collections\n\nimport kotlin.reflect.KProperty\nimport kotlin.internal.Exact\n\n/**\n * Returns the value of
the property for the given object from this read-only map.\n * @param thisRef the object for which the value is
requested
 (not used).\n * @param property the metadata for the property, used to get the name of property and lookup the
value corresponding to this name in the map.\n * @return the property value.\n *\n * @throws
NoSuchElementException when the map doesn't contain value for the property name and doesn't provide an implicit
default (see [withDefault]).\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <V, V1 : V> Map<in
```

String, @Exact V>.getValue(thisRef: Any?, property: KProperty<*>): V1 =\n @Suppress(\"UNCHECKED_CAST\") (getOrImplicitDefault(property.name) as V1)\n\n/**\n * Returns the value of the property for the given object from this mutable map.\n * @param thisRef the object for which the value is requested (not used).\n * @param property the metadata for the property, used to get the name of property and lookup the value corresponding to this name in the map.\n * @return the property value.\n *\n * @throws NoSuchElementException when the map doesn't contain value
for the property name and doesn't provide an implicit default (see [withDefault]).\n */\n@kotlin.jvm.JvmName(\"getVar\")\n@kotlin.internal.InlineOnly\npublic inline operator fun <V, V1 : V> MutableMap<in String, out @Exact V>.getValue(thisRef: Any?, property: KProperty<*>): V1 =\n @Suppress(\"UNCHECKED_CAST\") (getOrImplicitDefault(property.name) as V1)\n\n/**\n * Stores the value of the property for the given object in this mutable map.\n * @param thisRef the object for which the value is requested (not used).\n * @param property the metadata for the property, used to get the name of property and store the value associated with that name in the map.\n * @param value the value to set.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <V> MutableMap<in String, in V>.setValue(thisRef: Any?, property: KProperty<*>, value: V) {\n this.put(property.name, value)\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this
source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"MapsKt\")\n\npackage kotlin.collections\n\n/**\n * Returns the value for the given key, or the implicit default value for this map.\n * By default no implicit value is provided for maps and a [NoSuchElementException] is thrown.\n * To create a map with implicit default value use [withDefault] method.\n *\n * @throws NoSuchElementException when the map doesn't contain a value for the specified key and no implicit default was provided for that map.\n */\n@kotlin.jvm.JvmName(\"getOrImplicitDefaultNullable\")\n@PublishedApi\ninternal fun <K, V> Map<K, V>.getOrImplicitDefault(key: K): V {\n if (this is MapWithDefault)\n return this.getOrImplicitDefault(key)\n\n return getOrElseNullable(key, { throw NoSuchElementException(\"Key $key is missing in the map.\") })\n}\n\n/**\n * Returns a wrapper of this read-only
map, having the implicit default value provided with the specified function [defaultValue].\n *\n * This implicit default value is used when the original map doesn't contain a value for the key specified\n * and a value is obtained with [Map.getValue] function, for example when properties are delegated to the map.\n *\n * When this map already has an implicit default value provided with a former call to [withDefault], it is being replaced by this call.\n */\npublic fun <K, V> Map<K, V>.withDefault(defaultValue: (key: K) -> V): Map<K, V> =\n when (this) {\n is MapWithDefault -> this.map.withDefault(defaultValue)\n else -> MapWithDefaultImpl(this, defaultValue)\n }\n\n/**\n * Returns a wrapper of this mutable map, having the implicit default value provided with the specified function [defaultValue].\n *\n * This implicit default value is used when the original map doesn't contain a value for the key specified\n * and a value is obtained with [Map.getValue] function,
for example when properties are delegated to the map.\n *\n * When this map already has an implicit default value provided with a former call to [withDefault], it is being replaced by this call.\n */\n@kotlin.jvm.JvmName(\"withDefaultMutable\")\npublic fun <K, V> MutableMap<K, V>.withDefault(defaultValue: (key: K) -> V): MutableMap<K, V> =\n when (this) {\n is MutableMapWithDefault -> this.map.withDefault(defaultValue)\n else -> MutableMapWithDefaultImpl(this, defaultValue)\n }\n\n\nprivate interface MapWithDefault<K, out V> : Map<K, V> {\n public val map: Map<K, V>\n public fun getOrImplicitDefault(key: K): V\n}\n\nprivate interface MutableMapWithDefault<K, V> : MutableMap<K, V>, MapWithDefault<K, V> {\n public override val map: MutableMap<K, V>\n}\n\n\nprivate class MapWithDefaultImpl<K, out V>(public override val map: Map<K, V>, private val default: (key: K) -> V) : MapWithDefault<K, V> {\n override fun equals(other: Any?): Boolean = map.equals(other)\n override fun hashCode(): Int = map.hashCode()\n override fun toString(): String = map.toString()\n override val size: Int get() = map.size\n override fun isEmpty(): Boolean = map.isEmpty()\n override fun containsKey(key: K): Boolean = map.containsKey(key)\n override fun containsValue(value: @UnsafeVariance

V): Boolean = map.containsValue(value)\n   override fun get(key: K): V? = map.get(key)\n   override val keys: Set<K> get() = map.keys\n   override val values: Collection<V> get() = map.values\n   override val entries: Set<Map.Entry<K, V>> get() = map.entries\n\n   override fun getOrImplicitDefault(key: K): V = map.getOrElseNullable(key, { default(key) })\n}\n\nprivate class MutableMapWithDefaultImpl<K, V>(public override val map: MutableMap<K, V>, private val default: (key: K) -> V) : MutableMapWithDefault<K, V> {\n override fun equals(other: Any?): Boolean = map.equals(other)\n   override fun hashCode(): Int = map.hashCode()\n   override fun toString(): String = map.toString()\n   override val size: Int get() = map.size\n   override fun isEmpty(): Boolean = map.isEmpty()\n   override fun containsKey(key: K): Boolean = map.containsKey(key)\n   override fun containsValue(value: @UnsafeVariance V): Boolean = map.containsValue(value)\n   override fun get(key: K): V? = map.get(key)\n   override val keys: MutableSet<K> get() = map.keys\n   override val values: MutableCollection<V> get() = map.values\n   override val entries: MutableSet<MutableMap.MutableEntry<K, V>> get() = map.entries\n\n   override fun put(key: K, value: V): V? = map.put(key, value)\n   override fun remove(key: K): V? = map.remove(key)\n   override fun putAll(from: Map<out K, V>) = map.putAll(from)\n   override fun clear() = map.clear()\n\n   override fun getOrImplicitDefault(key: K): V = map.getOrElseNullable(key, { default(key) })\n}\n\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"CollectionsKt\")\n\npackage kotlin.collections\n\n/**\n * Removes a single instance of the specified element from this\n * collection, if it is present.\n *\n * Allows to overcome type-safety restriction of `remove` that requires to pass an element of type `E`.\n *\n * @return `true` if the element has been successfully removed; `false` if it was not present in the collection.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<out T>.remove(element: T): Boolean =\n   @Suppress(\"UNCHECKED_CAST\") (this as MutableCollection<T>).remove(element)\n\n/**\n * Removes all of this collection's elements that are also contained in the specified collection.\n\n * Allows to overcome type-safety restriction of `removeAll` that requires to pass a collection of type `Collection<E>`.\n *\n * @return `true` if any of the specified elements was removed from the collection, `false` if the collection was not modified.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<out T>.removeAll(elements: Collection<T>): Boolean =\n   @Suppress(\"UNCHECKED_CAST\") (this as MutableCollection<T>).removeAll(elements)\n\n/**\n * Retains only the elements in this collection that are contained in the specified collection.\n *\n * Allows to overcome type-safety restriction of `retainAll` that requires to pass a collection of type `Collection<E>`.\n *\n * @return `true` if any element was removed from the collection, `false` if the collection was not modified.\n */\n@kotlin.internal.InlineOnly\npublic inline fun <@kotlin.internal.OnlyInputTypes T> MutableCollection<out T>.retainAll(elements: Collection<T>): Boolean =\n   @Suppress(\"UNCHECKED_CAST\") (this as MutableCollection<T>).retainAll(elements)\n\n/**\n * Adds the specified [element] to this mutable collection.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> MutableCollection<in T>.plusAssign(element: T) {\n   this.add(element)\n}\n\n/**\n * Adds all elements of the given [elements] collection to this mutable collection.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> MutableCollection<in T>.plusAssign(elements: Iterable<T>) {\n   this.addAll(elements)\n}\n\n/**\n * Adds all elements of the given [elements] array to this mutable collection.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> MutableCollection<in T>.plusAssign(elements: Array<T>) {\n this.addAll(elements)\n}\n\n/**\n * Adds all elements of the given [elements] sequence to this mutable collection.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> MutableCollection<in T>.plusAssign(elements: Sequence<T>) {\n   this.addAll(elements)\n}\n\n/**\n * Removes a single instance of the specified [element] from this mutable collection.\n

*/\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> MutableCollection<in T>.minusAssign(element: T) {\n this.remove(element)\n}\n\n/**\n * Removes all elements contained in the given [elements] collection from this mutable collection.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> MutableCollection<in T>.minusAssign(elements: Iterable<T>) {\n this.removeAll(elements)\n}\n\n/**\n * Removes all elements contained in the given [elements] array from this mutable collection.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> MutableCollection<in T>.minusAssign(elements: Array<T>) {\n this.removeAll(elements)\n}\n\n/**\n * Removes all elements contained in the given [elements] sequence from this mutable collection.\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> MutableCollection<in T>.minusAssign(elements: Sequence<T>) {\n this.removeAll(elements)\n}\n\n/**\n * Adds all elements of the given [elements] collection
to this [MutableCollection].\n */\npublic fun <T> MutableCollection<in T>.addAll(elements: Iterable<T>): Boolean {\n when (elements) {\n is Collection -> return addAll(elements)\n else -> {\n var result: Boolean = false\n for (item in elements)\n if (add(item)) result = true\n return result\n }\n }\n}\n\n/**\n * Adds all elements of the given [elements] sequence to this [MutableCollection].\n */\npublic fun <T> MutableCollection<in T>.addAll(elements: Sequence<T>): Boolean {\n var result: Boolean = false\n for (item in elements) {\n if (add(item)) result = true\n }\n return result\n}\n\n/**\n * Adds all elements of the given [elements] array to this [MutableCollection].\n */\npublic fun <T> MutableCollection<in T>.addAll(elements: Array<out T>): Boolean {\n return addAll(elements.asList())\n}\n\n/**\n * Converts this [Iterable] to a list if it is not a [Collection].\n * Otherwise, returns
this.\n */\ninternal fun <T> Iterable<T>.convertToListIfNotCollection(): Collection<T> =\n if (this is Collection) this else toList()\n\n/**\n * Removes all elements from this [MutableCollection] that are also contained in the given [elements] collection.\n */\npublic fun <T> MutableCollection<in T>.removeAll(elements: Iterable<T>): Boolean {\n return removeAll(elements.convertToListIfNotCollection())\n}\n\n/**\n * Removes all elements from this [MutableCollection] that are also contained in the given [elements] sequence.\n */\npublic fun <T> MutableCollection<in T>.removeAll(elements: Sequence<T>): Boolean {\n val list = elements.toList()\n return list.isNotEmpty() && removeAll(list)\n}\n\n/**\n * Removes all elements from this [MutableCollection] that are also contained in the given [elements] array.\n */\npublic fun <T> MutableCollection<in T>.removeAll(elements: Array<out T>): Boolean {\n return elements.isNotEmpty() && removeAll(elements.asList())\n}\n\n/**\n * Retains
only elements of this [MutableCollection] that are contained in the given [elements] collection.\n */\npublic fun <T> MutableCollection<in T>.retainAll(elements: Iterable<T>): Boolean {\n return retainAll(elements.convertToListIfNotCollection())\n}\n\n/**\n * Retains only elements of this [MutableCollection] that are contained in the given [elements] array.\n */\npublic fun <T> MutableCollection<in T>.retainAll(elements: Array<out T>): Boolean {\n if (elements.isNotEmpty())\n return retainAll(elements.asList())\n else\n return retainNothing()\n}\n\n/**\n * Retains only elements of this [MutableCollection] that are contained in the given [elements] sequence.\n */\npublic fun <T> MutableCollection<in T>.retainAll(elements: Sequence<T>): Boolean {\n val list = elements.toList()\n if (list.isNotEmpty())\n return retainAll(list)\n else\n return retainNothing()\n}\n\nprivate fun MutableCollection<*>.retainNothing(): Boolean {\n val result
= isNotEmpty()\n clear()\n return result\n}\n\n\n/**\n * Removes all elements from this [MutableIterable] that match the given [predicate].\n *\n * @return `true` if any element was removed from this collection, or `false` when no elements were removed and collection was not modified.\n */\npublic fun <T> MutableIterable<T>.removeAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate, true)\n\n/**\n * Retains only elements of this [MutableIterable] that match the given [predicate].\n *\n * @return `true` if any element was removed from this collection, or `false` when all elements were retained and collection was not modified.\n */\npublic fun <T> MutableIterable<T>.retainAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate, false)\n\nprivate fun <T> MutableIterable<T>.filterInPlace(predicate: (T) -> Boolean, predicateResultToRemove: Boolean): Boolean {\n var result = false\n with(iterator()) {\n while (hasNext())\n if (predicate(next())

```
 == predicateResultToRemove) {\n                remove()\n                result = true\n            }\n    }\n    return result\n}\n\n\n/**\n * Removes the element at the specified [index] from this list.\n * In Kotlin one should use the [MutableList.removeAt] function instead.\n */\n@Deprecated(\"Use removeAt(index) instead.\", ReplaceWith(\"removeAt(index)\"), level = DeprecationLevel.ERROR)\n@kotlin.internal.InlineOnly\npublic inline fun <T> MutableList<T>.remove(index: Int): T = removeAt(index)\n\n/**\n * Removes the first element from this mutable list and returns that removed element, or throws [NoSuchElementException] if this list is empty.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T> MutableList<T>.removeFirst(): T = if (isEmpty()) throw NoSuchElementException(\"List is empty.\") else removeAt(0)\n\n/**\n * Removes the first element from this mutable list and returns that removed element, or returns `null` if this list is
 empty.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T> MutableList<T>.removeFirstOrNull(): T? = if (isEmpty()) null else removeAt(0)\n\n/**\n * Removes the last element from this mutable list and returns that removed element, or throws [NoSuchElementException] if this list is empty.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T> MutableList<T>.removeLast(): T = if (isEmpty()) throw NoSuchElementException(\"List is empty.\") else removeAt(lastIndex)\n\n/**\n * Removes the last element from this mutable list and returns that removed element, or returns `null` if this list is empty.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T> MutableList<T>.removeLastOrNull(): T? = if (isEmpty()) null else removeAt(lastIndex)\n\n/**\n * Removes all elements from this [MutableList] that match the given [predicate].\n *\n * @return `true` if any element was removed from this
 collection, or `false` when no elements were removed and collection was not modified.\n */\npublic fun <T> MutableList<T>.removeAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate, true)\n\n/**\n * Retains only elements of this [MutableList] that match the given [predicate].\n *\n * @return `true` if any element was removed from this collection, or `false` when all elements were retained and collection was not modified.\n */\npublic fun <T> MutableList<T>.retainAll(predicate: (T) -> Boolean): Boolean = filterInPlace(predicate, false)\n\nprivate fun <T> MutableList<T>.filterInPlace(predicate: (T) -> Boolean, predicateResultToRemove: Boolean): Boolean {\n    if (this !is RandomAccess)\n        return (this as MutableIterable<T>).filterInPlace(predicate, predicateResultToRemove)\n\n    var writeIndex: Int = 0\n    for (readIndex in 0..lastIndex) {\n        val element = this[readIndex]\n        if (predicate(element) == predicateResultToRemove)\n            continue\n\n
        if (writeIndex != readIndex)\n            this[writeIndex] = element\n\n        writeIndex++\n    }\n    if (writeIndex < size) {\n        for (removeIndex in lastIndex downTo writeIndex)\n            removeAt(removeIndex)\n\n        return true\n    } else {\n        return false\n    }\n}\n","/*\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin.collections\n\n/** An iterator over a sequence of values of type `Byte`. */\npublic abstract class ByteIterator : Iterator<Byte> {\n override final fun next() = nextByte()\n\n    /** Returns the next value in the sequence without boxing. */\n    public abstract fun nextByte(): Byte\n}\n\n/** An iterator over a sequence of values of type `Char`. */\npublic abstract class CharIterator : Iterator<Char> {\n    override final
 fun next() = nextChar()\n\n    /** Returns the next value in the sequence without boxing. */\n    public abstract fun nextChar(): Char\n}\n\n/** An iterator over a sequence of values of type `Short`. */\npublic abstract class ShortIterator : Iterator<Short> {\n    override final fun next() = nextShort()\n\n    /** Returns the next value in the sequence without boxing. */\n    public abstract fun nextShort(): Short\n}\n\n/** An iterator over a sequence of values of type `Int`. */\npublic abstract class IntIterator : Iterator<Int> {\n    override final fun next() = nextInt()\n\n    /** Returns the next value in the sequence without boxing. */\n    public abstract fun nextInt(): Int\n}\n\n/** An iterator over a sequence of values of type `Long`. */\npublic abstract class LongIterator : Iterator<Long> {\n override final fun next() = nextLong()\n\n    /** Returns the next value in the sequence without boxing. */\n    public
```

```
abstract fun nextLong(): Long\n}\n\n/** An iterator over a sequence
 of values of type `Float`. */\npublic abstract class FloatIterator : Iterator<Float> {\n    override final fun next() =
nextFloat()\n\n    /** Returns the next value in the sequence without boxing. */\n    public abstract fun nextFloat():
Float\n}\n\n/** An iterator over a sequence of values of type `Double`. */\npublic abstract class DoubleIterator :
Iterator<Double> {\n    override final fun next() = nextDouble()\n\n    /** Returns the next value in the sequence
without boxing. */\n    public abstract fun nextDouble(): Double\n}\n\n/** An iterator over a sequence of values of
type `Boolean`. */\npublic abstract class BooleanIterator : Iterator<Boolean> {\n    override final fun next() =
nextBoolean()\n\n    /** Returns the next value in the sequence without boxing. */\n    public abstract fun
nextBoolean(): Boolean\n}\n\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language
contributors.\n * Use of this source code is governed by the Apache 2.0 license that
 can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"CollectionsKt\")\n\npackage
kotlin.collections\n\nprivate open class ReversedListReadOnly<out T>(private val delegate: List<T>) :
AbstractList<T>() {\n    override val size: Int get() = delegate.size\n    override fun get(index: Int): T =
delegate[reverseElementIndex(index)]\n}\n\nprivate class ReversedList<T>(private val delegate: MutableList<T>) :
AbstractMutableList<T>() {\n    override val size: Int get() = delegate.size\n    override fun get(index: Int): T =
delegate[reverseElementIndex(index)]\n\n    override fun clear() = delegate.clear()\n    override fun removeAt(index:
Int): T = delegate.removeAt(reverseElementIndex(index))\n\n    override fun set(index: Int, element: T): T =
delegate.set(reverseElementIndex(index), element)\n    override fun add(index: Int, element: T) {\n
delegate.add(reversePositionIndex(index), element)\n    }\n}\n\nprivate fun List<*>.reverseElementIndex(index:
 Int) =\n    if (index in 0..lastIndex) lastIndex - index else throw IndexOutOfBoundsException(\"Element index
$index must be in range [${0..lastIndex}].\")\n\nprivate fun List<*>.reversePositionIndex(index: Int) =\n    if (index
in 0..size) size - index else throw IndexOutOfBoundsException(\"Position index $index must be in range
[${0..size}].\")\n\n\n/**\n * Returns a reversed read-only view of the original List.\n * All changes made in the
original list will be reflected in the reversed one.\n * @sample samples.collections.ReversedViews.asReversedList\n
*/\npublic fun <T> List<T>.asReversed(): List<T> = ReversedListReadOnly(this)\n\n/**\n * Returns a reversed
mutable view of the original mutable List.\n * All changes made in the original list will be reflected in the reversed
one and vice versa.\n * @sample samples.collections.ReversedViews.asReversedMutableList\n
*/\n@kotlin.jvm.JvmName(\"asReversedMutable\")\npublic fun <T> MutableList<T>.asReversed():
 MutableList<T> = ReversedList(this)\n\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming
Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"SequencesKt\")\n@file:OptIn(Experimenta
lTypeInference::class)\n\npackage kotlin.sequences\n\nimport kotlin.coroutines.*\nimport
kotlin.coroutines.intrinsics.*\nimport kotlin.experimental.ExperimentalTypeInference\n\n/**\n * Builds a
[Sequence] lazily yielding values one by one.\n *\n * @see kotlin.sequences.generateSequence\n *\n * @sample
samples.collections.Sequences.Building.buildSequenceYieldAll\n * @sample
samples.collections.Sequences.Building.buildFibonacciSequence\n
*/\n@SinceKotlin(\"1.3\")\n@Suppress(\"DEPRECATION\")\npublic fun <T> sequence(@BuilderInference block:
suspend SequenceScope<T>.() -> Unit): Sequence<T> = Sequence { iterator(block) }\n\n/**\n * Builds
 an [Iterator] lazily yielding values one by one.\n *\n * @sample
samples.collections.Sequences.Building.buildIterator\n * @sample samples.collections.Iterables.Building.iterable\n
*/\n@SinceKotlin(\"1.3\")\n@Suppress(\"DEPRECATION\")\npublic fun <T> iterator(@BuilderInference block:
suspend SequenceScope<T>.() -> Unit): Iterator<T> {\n    val iterator = SequenceBuilderIterator<T>()\n
iterator.nextStep = block.createCoroutineUnintercepted(receiver = iterator, completion = iterator)\n    return
iterator\n}\n\n/**\n * The scope for yielding values of a [Sequence] or an [Iterator], provides [yield] and [yieldAll]
suspension functions.\n *\n * @see sequence\n * @see iterator\n *\n * @sample
samples.collections.Sequences.Building.buildSequenceYieldAll\n * @sample
```

samples.collections.Sequences.Building.buildFibonacciSequence\n */\n@RestrictsSuspension\n@SinceKotlin(\"1.3\")\npublic abstract class SequenceScope<in T> internal constructor() {\n /**\n * Yields a value to the [Iterator] being built and suspends\n * until the next value is requested.\n *\n * @sample samples.collections.Sequences.Building.buildSequenceYieldAll\n * @sample samples.collections.Sequences.Building.buildFibonacciSequence\n */\n public abstract suspend fun yield(value: T)\n\n /**\n * Yields all values from the `iterator` to the [Iterator] being built\n * and suspends until all these values are iterated and the next one is requested.\n *\n * The sequence of values returned by the given iterator can be potentially infinite.\n *\n * @sample samples.collections.Sequences.Building.buildSequenceYieldAll\n */\n public abstract suspend fun yieldAll(iterator: Iterator<T>)\n\n /**\n * Yields a collections of values to the [Iterator] being built\n * and suspends until all these values are iterated and the next one is requested.\n *\n * @sample samples.collections.Sequences.Building.buildSequenceYieldAll\n */\n public suspend fun yieldAll(elements: Iterable<T>) {\n if (elements is Collection && elements.isEmpty()) return\n return yieldAll(elements.iterator())\n }\n\n /**\n * Yields potentially infinite sequence of values to the [Iterator] being built\n * and suspends until all these values are iterated and the next one is requested.\n *\n * The sequence can be potentially infinite.\n *\n * @sample samples.collections.Sequences.Building.buildSequenceYieldAll\n */\n public suspend fun yieldAll(sequence: Sequence<T>) = yieldAll(sequence.iterator())\n}\n\nprivate typealias State = Int\n\nprivate const val State_NotReady: State = 0\nprivate const val State_ManyNotReady: State = 1\nprivate const val State_ManyReady: State = 2\nprivate const val State_Ready: State = 3\nprivate const val State_Done: State = 4\nprivate const val State_Failed: State = 5\n\nprivate class SequenceBuilderIterator<T> : SequenceScope<T>(), Iterator<T>, Continuation<Unit> {\n private var state = State_NotReady\n private var nextValue: T? = null\n private var nextIterator: Iterator<T>? = null\n var nextStep: Continuation<Unit>? = null\n\n override fun hasNext(): Boolean {\n while (true) {\n when (state) {\n State_NotReady -> {}\n State_ManyNotReady ->\n if (nextIterator!!.hasNext()) {\n state = State_ManyReady\n return true\n } else {\n nextIterator = null\n }\n State_Done -> return false\n State_Ready, State_ManyReady -> return true\n else -> throw exceptionalState()\n }\n\n state = State_Failed\n val step = nextStep!!\n nextStep = null\n step.resume(Unit)\n }\n }\n\n override fun next(): T {\n when (state) {\n State_NotReady, State_ManyNotReady -> return nextNotReady()\n State_ManyReady -> {\n state = State_ManyNotReady\n return nextIterator!!.next()\n }\n State_Ready -> {\n state = State_NotReady\n @Suppress(\"UNCHECKED_CAST\")\n val result = nextValue as T\n nextValue = null\n return result\n }\n else -> throw exceptionalState()\n }\n }\n\n private fun nextNotReady(): T {\n if (!hasNext()) throw NoSuchElementException() else return next()\n }\n\n private fun exceptionalState(): Throwable = when (state) {\n State_Done -> NoSuchElementException()\n State_Failed -> IllegalStateException(\"Iterator has failed.\")\n else -> IllegalStateException(\"Unexpected state of the iterator: $state\")\n }\n\n override suspend fun yield(value: T) {\n nextValue = value\n state = State_Ready\n return suspendCoroutineUninterceptedOrReturn { c ->\n nextStep = c\n COROUTINE_SUSPENDED\n }\n }\n\n override suspend fun yieldAll(iterator: Iterator<T>) {\n if (!iterator.hasNext()) return\n nextIterator = iterator\n state = State_ManyReady\n return suspendCoroutineUninterceptedOrReturn { c ->\n nextStep = c\n COROUTINE_SUSPENDED\n }\n }\n\n // Completion continuation implementation\n override fun resumeWith(result: Result<Unit>) {\n result.getOrThrow() // just rethrow exception if it is there\n state = State_Done\n }\n\n override val context: CoroutineContext\n get() = EmptyCoroutineContext\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\ninternal fun checkWindowSizeStep(size:

```
Int, step: Int) {\n    require(size > 0 && step > 0) {\n        if (size != step)\n            \"Both size $size and step $step must be greater than zero.\"\n        else\n            \"size $size must be greater than zero.\"\n    }\n}\n\ninternal fun <T> Sequence<T>.windowedSequence(size: Int, step: Int, partialWindows: Boolean, reuseBuffer: Boolean): Sequence<List<T>> {\n    checkWindowSizeStep(size, step)\n    return Sequence { windowedIterator(iterator(), size, step, partialWindows, reuseBuffer) }\n}\n\ninternal fun <T> windowedIterator(iterator: Iterator<T>, size: Int, step: Int, partialWindows: Boolean, reuseBuffer: Boolean): Iterator<List<T>> {\n    if (!iterator.hasNext()) return EmptyIterator\n    return iterator<List<T>> {\n        val bufferInitialCapacity = size.coerceAtMost(1024)\n        val gap = step - size\n        if (gap >= 0) {\n            var buffer = ArrayList<T>(bufferInitialCapacity)\n            var skip = 0\n            for (e in iterator) {\n                if (skip > 0) { skip -= 1; continue }\n                buffer.add(e)\n                if (buffer.size == size) {\n                    yield(buffer)\n                    if (reuseBuffer) buffer.clear() else buffer = ArrayList(size)\n                    skip = gap\n                }\n            }\n            if (buffer.isNotEmpty()) {\n                if (partialWindows || buffer.size == size) yield(buffer)\n            }\n        } else {\n            var buffer = RingBuffer<T>(bufferInitialCapacity)\n            for (e in iterator) {\n                buffer.add(e)\n                if (buffer.isFull()) {\n                    if (buffer.size < size) { buffer = buffer.expanded(maxCapacity = size); continue }\n\n                    yield(if (reuseBuffer) buffer else ArrayList(buffer))\n                    buffer.removeFirst(step)\n                }\n            }\n            if (partialWindows) {\n                while (buffer.size > step) {\n                    yield(if (reuseBuffer) buffer else ArrayList(buffer))\n                    buffer.removeFirst(step)\n                }\n                if (buffer.isNotEmpty()) yield(buffer)\n            }\n        }\n    }\n}\n\ninternal class MovingSubList<out E>(private val list: List<E>) : AbstractList<E>(), RandomAccess {\n    private var fromIndex: Int = 0\n    private var _size: Int = 0\n\n    fun move(fromIndex: Int, toIndex: Int) {\n        checkRangeIndexes(fromIndex, toIndex, list.size)\n        this.fromIndex = fromIndex\n        this._size = toIndex - fromIndex\n    }\n\n    override fun get(index: Int): E {\n        checkElementIndex(index, _size)\n\n        return list[fromIndex + index]\n    }\n\n    override val size: Int get() = _size\n}\n\n\n/**\n * Provides ring buffer implementation.\n *\n * Buffer overflow is not allowed so [add] doesn't overwrite tail but raises an exception.\n */\nprivate class RingBuffer<T>(private val buffer: Array<Any?>, filledSize: Int) : AbstractList<T>(), RandomAccess {\n    init {\n        require(filledSize >= 0) { \"ring buffer filled size should not be negative but it is $filledSize\" }\n        require(filledSize <= buffer.size) { \"ring buffer filled size: $filledSize cannot be larger than the buffer size: ${buffer.size}\" }\n    }\n\n    constructor(capacity: Int) : this(arrayOfNulls<Any?>(capacity), 0)\n\n    private val capacity = buffer.size\n    private var startIndex: Int = 0\n\n    override var size: Int = filledSize\n        private set\n\n    override fun get(index: Int): T {\n        checkElementIndex(index, size)\n        @Suppress(\"UNCHECKED_CAST\")\n        return buffer[startIndex.forward(index)] as T\n    }\n\n    fun isFull() = size == capacity\n\n    override fun iterator(): Iterator<T> = object : AbstractIterator<T>() {\n        private var count = size\n        private var index = startIndex\n\n        override fun computeNext() {\n            if (count == 0) {\n                done()\n            } else {\n                @Suppress(\"UNCHECKED_CAST\")\n                setNext(buffer[index] as T)\n                index = index.forward(1)\n                count--\n            }\n        }\n    }\n\n    @Suppress(\"UNCHECKED_CAST\")\n    override fun <T> toArray(array: Array<T>): Array<T> {\n        val result: Array<T?> =\n            if (array.size < this.size) array.copyOf(this.size) else array as Array<T?>\n\n        val size = this.size\n\n        var widx = 0\n        var idx = startIndex\n\n        while (widx < size && idx < capacity) {\n            result[widx] = buffer[idx] as T\n            widx++\n            idx++\n        }\n\n        idx = 0\n        while (widx < size) {\n            result[widx] = buffer[idx] as T\n            widx++\n            idx++\n        }\n        if (result.size > this.size) result[this.size] = null\n\n        return result as Array<T>\n    }\n\n    override fun toArray(): Array<Any?> {\n        return toArray(arrayOfNulls(size))\n    }\n\n    /**\n     * Creates a new ring buffer with the capacity equal to the minimum of [maxCapacity] and 1.5 * [capacity].\n     * The returned ring buffer contains the same elements as this ring buffer.\n     */\n    fun expanded(maxCapacity: Int): RingBuffer<T> {\n        val newCapacity = (capacity + (capacity shr 1) + 1).coerceAtMost(maxCapacity)\n        val newBuffer = if (startIndex == 0) buffer.copyOf(newCapacity) else toArray(arrayOfNulls(newCapacity))\n        return RingBuffer(newBuffer, size)\n    }\n\n    /**\n     * Add [element] to the buffer or fail with [IllegalStateException] if
```

```
no free space available in the buffer\n     */\n    fun add(element: T) {\n        if (isFull()) {\n            throw IllegalStateException(\"ring buffer is full\")\n        }\n\n        buffer[startIndex.forward(size)] = element\n        size++\n    }\n\n    /**\n     * Removes [n] first elements from the buffer or fails with [IllegalArgumentException] if not enough elements in the buffer to remove\n     */\n    fun removeFirst(n: Int) {\n        require(n >= 0) { \"n shouldn't be negative but it is $n\" }\n        require(n <= size) { \"n shouldn't be greater than the buffer size: n = $n, size = $size\" }\n\n        if (n > 0) {\n            val start = startIndex\n            val end = start.forward(n)\n\n            if (start > end) {\n                buffer.fill(null, start, capacity)\n                buffer.fill(null, 0, end)\n            } else {\n                buffer.fill(null, start, end)\n            }\n\n            startIndex = end\n            size -= n\n        }\n    }\n\n\n    @Suppress(\"NOTHING_TO_INLINE\")\n    private inline fun Int.forward(n: Int): Int = (this + n) % capacity\n}\n","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.collections\n\n// UByteArray
================================================================================\n@ExperimentalUnsignedTypes\nprivate fun partition(\n    array: UByteArray, left: Int, right: Int): Int {\n    var i = left\n    var j = right\n    val pivot = array[(left + right) / 2]\n    while (i <= j) {\n        while (array[i] < pivot)\n            i++\n        while (array[j] > pivot)\n            j--\n        if (i <= j) {\n            val tmp = array[i]\n            array[i] = array[j]\n            array[j] = tmp\n            i++\n            j--\n        }\n    }\n    return i\n}\n\n@ExperimentalUnsignedTypes\nprivate fun quickSort(\n    array: UByteArray, left: Int, right: Int) {\n    val index = partition(array, left, right)\n    if (left < index - 1)\n        quickSort(array, left, index - 1)\n    if (index < right)\n        quickSort(array, index, right)\n}\n\n// UShortArray
================================================================================\n@ExperimentalUnsignedTypes\nprivate fun partition(\n    array: UShortArray, left: Int,
    right: Int): Int {\n    var i = left\n    var j = right\n    val pivot = array[(left + right) / 2]\n    while (i <= j) {\n        while (array[i] < pivot)\n            i++\n        while (array[j] > pivot)\n            j--\n        if (i <= j) {\n            val tmp = array[i]\n            array[i] = array[j]\n            array[j] = tmp\n            i++\n            j--\n        }\n    }\n    return i\n}\n\n@ExperimentalUnsignedTypes\nprivate fun quickSort(\n    array: UShortArray, left: Int, right: Int) {\n    val index = partition(array, left, right)\n    if (left < index - 1)\n        quickSort(array, left, index - 1)\n    if (index < right)\n        quickSort(array, index, right)\n}\n\n// UIntArray
================================================================================\n@ExperimentalUnsignedTypes\nprivate fun partition(\n    array: UIntArray, left: Int, right: Int): Int {\n    var i = left\n    var j = right\n    val pivot = array[(left + right) / 2]\n    while (i <= j)
    {\n        while (array[i] < pivot)\n            i++\n        while (array[j] > pivot)\n            j--\n        if (i <= j) {\n            val tmp = array[i]\n            array[i] = array[j]\n            array[j] = tmp\n            i++\n            j--\n        }\n    }\n    return i\n}\n\n@ExperimentalUnsignedTypes\nprivate fun quickSort(\n    array: UIntArray, left: Int, right: Int) {\n    val index = partition(array, left, right)\n    if (left < index - 1)\n        quickSort(array, left, index - 1)\n    if (index < right)\n        quickSort(array, index, right)\n}\n\n// ULongArray
================================================================================\n@ExperimentalUnsignedTypes\nprivate fun partition(\n    array: ULongArray, left: Int, right: Int): Int {\n    var i = left\n    var j = right\n    val pivot = array[(left + right) / 2]\n    while (i <= j) {\n        while (array[i] < pivot)\n            i++\n        while (array[j] > pivot)\n            j--\n        if (i <=
    j) {\n            val tmp = array[i]\n            array[i] = array[j]\n            array[j] = tmp\n            i++\n            j--\n        }\n    }\n    return i\n}\n\n@ExperimentalUnsignedTypes\nprivate fun quickSort(\n    array: ULongArray, left: Int, right: Int) {\n    val index = partition(array, left, right)\n    if (left < index - 1)\n        quickSort(array, left, index - 1)\n    if (index < right)\n        quickSort(array, index, right)\n}\n\n\n// Interfaces
================================================================================\n/**\n * Sorts the given array using qsort algorithm.\n */\n@ExperimentalUnsignedTypes\ninternal fun sortArray(array: UByteArray, fromIndex: Int, toIndex: Int) = quickSort(array, fromIndex, toIndex - 1)\n@ExperimentalUnsignedTypes\ninternal fun sortArray(array: UShortArray, fromIndex: Int, toIndex: Int) =
```

quickSort(array, fromIndex, toIndex - 1)\n@ExperimentalUnsignedTypes\ninternal fun sortArray(array: UIntArray, fromIndex: Int, toIndex:
Int) = quickSort(array, fromIndex, toIndex - 1)\n@ExperimentalUnsignedTypes\ninternal fun sortArray(array: ULongArray, fromIndex: Int, toIndex: Int) = quickSort(array, fromIndex, toIndex - 1)","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\nimport kotlin.internal.InlineOnly\n\n/**\n * Compares this object with the specified object for order. Returns zero if this object is equal\n * to the specified [other] object, a negative number if it's less than [other], or a positive number\n * if it's greater than [other].\n *\n * This function delegates to [Comparable.compareTo] and allows to call it in infix form.\n */\n@InlineOnly\n@SinceKotlin(\"1.6\")\npublic inline infix fun <T> Comparable<T>.compareTo(other: T): Int =\n this.compareTo(other)\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin
Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.contracts\n\nimport kotlin.internal.ContractsDsl\nimport kotlin.internal.InlineOnly\n\n/**\n * This marker distinguishes the experimental contract declaration API and is used to opt-in for that feature\n * when declaring contracts of user functions.\n *\n * Any usage of a declaration annotated with `@ExperimentalContracts` must be accepted either by\n * annotating that usage with the [OptIn] annotation, e.g. `@OptIn(ExperimentalContracts::class)`,\n * or by using the compiler argument `-opt-in=kotlin.contracts.ExperimentalContracts`.\n */\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin(\"1.3\")\n@RequiresOptIn\n@MustBeDocumented\npublic annotation class ExperimentalContracts\n\n/**\n * Provides a scope, where the functions of the contract DSL, such as [returns], [callsInPlace], etc.,\n * can be used to
describe the contract of a function.\n *\n * This type is used as a receiver type of the lambda function passed to the [contract] function.\n *\n * @see contract\n */\n@ContractsDsl\n@ExperimentalContracts\n@SinceKotlin(\"1.3\")\npublic interface ContractBuilder {\n /**\n * Describes a situation when a function returns normally, without any exceptions thrown.\n *\n * Use [SimpleEffect.implies] function to describe a conditional effect that happens in such case.\n *\n */\n // @sample samples.contracts.returnsContract\n @ContractsDsl public fun returns(): Returns\n\n /**\n * Describes a situation when a function returns normally with the specified return [value].\n *\n * The possible values of [value] are limited to `true`, `false` or `null`.\n *\n * Use [SimpleEffect.implies] function to describe a conditional effect that happens in such case.\n *\n */\n // @sample samples.contracts.returnsTrueContract\n // @sample samples.contracts.returnsFalseContract\n
// @sample samples.contracts.returnsNullContract\n @ContractsDsl public fun returns(value: Any?): Returns\n\n /**\n * Describes a situation when a function returns normally with any value that is not `null`.\n *\n * Use [SimpleEffect.implies] function to describe a conditional effect that happens in such case.\n *\n */\n // @sample samples.contracts.returnsNotNullContract\n @ContractsDsl public fun returnsNotNull(): ReturnsNotNull\n\n /**\n * Specifies that the function parameter [lambda] is invoked in place.\n *\n * This contract specifies that:\n * 1. the function [lambda] can only be invoked during the call of the owner function,\n * and it won't be invoked after that owner function call is completed;\n * 2. _(optionally)_ the function [lambda] is invoked the amount of times specified by the [kind] parameter,\n * see the [InvocationKind] enum for possible values.\n *\n * A
function declaring the `callsInPlace` effect must be _inline_.\n *\n */\n /* @sample samples.contracts.callsInPlaceAtMostOnceContract\n * @sample samples.contracts.callsInPlaceAtLeastOnceContract\n * @sample samples.contracts.callsInPlaceExactlyOnceContract\n * @sample samples.contracts.callsInPlaceUnknownContract\n */\n @ContractsDsl public fun <R> callsInPlace(lambda: Function<R>, kind: InvocationKind = InvocationKind.UNKNOWN): CallsInPlace\n}\n\n/**\n * Specifies how many times a function invokes its function parameter in place.\n *\n * See [ContractBuilder.callsInPlace] for the details of the call-in-place function contract.\n

*/\n@ContractsDsl\n@ExperimentalContracts\n@SinceKotlin(\"1.3\")\npublic enum class InvocationKind {\n /**\n * A function parameter will be invoked one time or not invoked at all.\n */\n // @sample samples.contracts.callsInPlaceAtMostOnceContract\n @ContractsDsl AT_MOST_ONCE,\n\n /**\n * A function parameter
will be invoked one or more times.\n *\n */\n // @sample samples.contracts.callsInPlaceAtLeastOnceContract\n @ContractsDsl AT_LEAST_ONCE,\n\n /**\n * A function parameter will be invoked exactly one time.\n *\n */\n // @sample samples.contracts.callsInPlaceExactlyOnceContract\n @ContractsDsl EXACTLY_ONCE,\n\n /**\n * A function parameter is called in place, but it's unknown how many times it can be called.\n *\n */\n // @sample samples.contracts.callsInPlaceUnknownContract\n @ContractsDsl UNKNOWN\n}\n\n/**\n * Specifies the contract of a function.\n *\n * The contract description must be at the beginning of a function and have at least one effect.\n *\n * Only the top-level functions can have a contract for now.\n *\n * @param builder the lambda where the contract of a function is described with the help of the [ContractBuilder] members.\n *\n */\n/* @sample samples.contracts.returnsContract\n* @sample samples.contracts.returnsTrueContract\n* @sample samples.contracts.returnsFalseContract\n* @sample samples.contracts.returnsNullContract\n* @sample samples.contracts.returnsNotNullContract\n* @sample samples.contracts.callsInPlaceAtMostOnceContract\n* @sample samples.contracts.callsInPlaceAtLeastOnceContract\n* @sample samples.contracts.callsInPlaceExactlyOnceContract\n* @sample samples.contracts.callsInPlaceUnknownContract\n*/\n@ContractsDsl\n@ExperimentalContracts\n@InlineOnly\n@SinceKotlin(\"1.3\")\n@Suppress(\"UNUSED_PARAMETER\")\npublic inline fun contract(builder: ContractBuilder.() -> Unit) { }\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines\n\n/**\n * Marks coroutine context element that intercepts coroutine continuations.\n * The coroutines framework uses [ContinuationInterceptor.Key] to retrieve the interceptor
and\n * intercepts all coroutine continuations with [interceptContinuation] invocations.\n *\n * [ContinuationInterceptor] behaves like a [polymorphic element][AbstractCoroutineContextKey], meaning that\n * its implementation delegates [get][CoroutineContext.Element.get] and [minusKey][CoroutineContext.Element.minusKey]\n * to [getPolymorphicElement] and [minusPolymorphicKey] respectively.\n * [ContinuationInterceptor] subtypes can be extracted from the coroutine context using either [ContinuationInterceptor.Key]\n * or subtype key if it extends [AbstractCoroutineContextKey].\n */\n@SinceKotlin(\"1.3\")\npublic interface ContinuationInterceptor : CoroutineContext.Element {\n /**\n * The key that defines *the* context interceptor.\n */\n companion object Key : CoroutineContext.Key<ContinuationInterceptor>\n\n /**\n * Returns continuation that wraps the original [continuation], thus intercepting all resumptions.\n * This function is invoked by coroutines framework
when needed and the resulting continuations are\n * cached internally per each instance of the original [continuation].\n *\n * This function may simply return original [continuation] if it does not want to intercept this particular continuation.\n *\n * When the original [continuation] completes, coroutine framework invokes [releaseInterceptedContinuation]\n * with the resulting continuation if it was intercepted, that is if `interceptContinuation` had previously\n * returned a different continuation instance.\n */\n public fun <T> interceptContinuation(continuation: Continuation<T>): Continuation<T>\n\n /**\n * Invoked for the continuation instance returned by [interceptContinuation] when the original\n * continuation completes and will not be used anymore. This function is invoked only if [interceptContinuation]\n * had returned a different continuation instance from the one it was invoked with.\n *\n * Default implementation
does nothing.\n *\n * @param continuation Continuation instance returned by this interceptor's [interceptContinuation] invocation.\n */\n public fun releaseInterceptedContinuation(continuation: Continuation<*>) {\n /* do nothing by default */\n }\n\n public override operator fun <E : CoroutineContext.Element> get(key: CoroutineContext.Key<E>): E? {\n // getPolymorphicKey specialized for

ContinuationInterceptor key\n @OptIn(ExperimentalStdlibApi::class)\n if (key is AbstractCoroutineContextKey<*, *>) {\n @Suppress(\"UNCHECKED_CAST\")\n return if (key.isSubKey(this.key)) key.tryCast(this) as? E else null\n }\n @Suppress(\"UNCHECKED_CAST\")\n return if (ContinuationInterceptor === key) this as E else null\n }\n\n\n public override fun minusKey(key: CoroutineContext.Key<*>): CoroutineContext {\n // minusPolymorphicKey specialized for ContinuationInterceptor key\n @OptIn(ExperimentalStdlibApi::class)\n if (key is AbstractCoroutineContextKey<*, *>) {\n return if (key.isSubKey(this.key) && key.tryCast(this) != null) EmptyCoroutineContext else this\n }\n return if (ContinuationInterceptor === key) EmptyCoroutineContext else this\n }\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines\n\n/**\n * Persistent context for the coroutine. It is an indexed set of [Element] instances.\n * An indexed set is a mix between a set and a map.\n * Every element in this set has a unique [Key].\n */\n@SinceKotlin(\"1.3\")\npublic interface CoroutineContext {\n /**\n * Returns the element with the given [key] from this context or `null`.\n */\n public operator fun <E : Element> get(key: Key<E>): E?\n\n /**\n * Accumulates entries of this context starting with [initial] value and applying [operation]\n * from left to right to current accumulator value and each element of this context.\n */\n public fun <R> fold(initial: R, operation: (R, Element) -> R): R\n\n /**\n * Returns a context containing elements from this context and elements from other [context].\n * The elements from this context with the same key as in the other one are dropped.\n */\n public operator fun plus(context: CoroutineContext): CoroutineContext =\n if (context === EmptyCoroutineContext) this else // fast path -- avoid lambda creation\n context.fold(this) { acc, element ->\n val removed = acc.minusKey(element.key)\n if (removed === EmptyCoroutineContext) element else {\n // make sure interceptor is always last in the context (and thus is fast to get when present)\n val interceptor = removed[ContinuationInterceptor]\n if (interceptor == null) CombinedContext(removed, element) else {\n val left = removed.minusKey(ContinuationInterceptor)\n if (left === EmptyCoroutineContext) CombinedContext(element, interceptor) else\n CombinedContext(CombinedContext(left, element), interceptor)\n }\n }\n }\n\n /**\n * Returns a context containing elements from this context, but without an element with\n * the specified [key].\n */\n public fun minusKey(key: Key<*>): CoroutineContext\n\n /**\n * Key for the elements of [CoroutineContext]. [E] is a type of element with this key.\n */\n public interface Key<E : Element>\n\n /**\n * An element of the [CoroutineContext]. An element of the coroutine context is a singleton context by itself.\n */\n public interface Element : CoroutineContext {\n /**\n * A key of this coroutine context element.\n */\n public val key: Key<*>\n\n public override operator fun <E : Element> get(key: Key<E>): E? =\n @Suppress(\"UNCHECKED_CAST\")\n if (this.key == key) this as E else null\n\n public override fun <R> fold(initial: R, operation: (R, Element) -> R): R =\n operation(initial, this)\n\n public override fun minusKey(key: Key<*>): CoroutineContext =\n if (this.key == key) EmptyCoroutineContext else this\n }\n}\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.coroutines\n\nimport kotlin.coroutines.CoroutineContext.Element\nimport kotlin.coroutines.CoroutineContext.Key\n\n/**\n * Base class for [CoroutineContext.Element] implementations.\n */\n@SinceKotlin(\"1.3\")\npublic abstract class AbstractCoroutineContextElement(public override val key: Key<*>) : Element\n\n/**\n * Base class for [CoroutineContext.Key] associated with polymorphic [CoroutineContext.Element] implementation.\n * Polymorphic element implementation implies delegating its [get][Element.get] and [minusKey][Element.minusKey]\n * to [getPolymorphicElement] and [minusPolymorphicKey] respectively.\n *\n * Polymorphic elements can be extracted from the coroutine context using both element key and its supertype key.\n * Example of polymorphic elements:\n * ```\n * open class BaseElement : CoroutineContext.Element {\n * companion object Key : CoroutineContext.Key<BaseElement>\n

```
*     override val key: CoroutineContext.Key<*> get() = Key\n *     // It is important to use getPolymorphicKey and minusPolymorphicKey\n *     override fun <E : CoroutineContext.Element> get(key: CoroutineContext.Key<E>): E? = getPolymorphicElement(key)\n *     override fun minusKey(key: CoroutineContext.Key<*>): CoroutineContext = minusPolymorphicKey(key)\n * }\n *\n * class DerivedElement : BaseElement() {\n *     companion object Key : AbstractCoroutineContextKey<BaseElement, DerivedElement>(BaseElement, { it as? DerivedElement })\n * }\n * // Now it is possible to query both `BaseElement` and `DerivedElement`\n * someContext[BaseElement] // Returns BaseElement?, non-null both for BaseElement and DerivedElement instances\n * someContext[DerivedElement] // Returns DerivedElement?, non-null only for DerivedElement instance\n * ```\n * @param B base class of a polymorphic element\n * @param baseKey an instance of base key\n * @param E element type associated with the current key\n * @param safeCast a function that can safely cast abstract [CoroutineContext.Element] to the concrete [E] type\n *                 and return the element if it is a subtype of [E] or `null` otherwise.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalStdlibApi\npublic abstract class AbstractCoroutineContextKey<B : Element, E : B>(\n    baseKey: Key<B>,\n    private val safeCast: (element: Element) -> E?\n) : Key<E> {\n    private val topmostKey: Key<*> = if (baseKey is AbstractCoroutineContextKey<*, *>) baseKey.topmostKey else baseKey\n\n    internal fun tryCast(element: Element): E? = safeCast(element)\n    internal fun isSubKey(key: Key<*>): Boolean = key === this || topmostKey === key\n}\n\n/**\n * Returns the current element if it is associated with the given [key] in a polymorphic manner or `null` otherwise.\n * This method returns non-null value if either [Element.key] is equal to the given [key] or if the [key] is associated\n * with [Element.key] via [AbstractCoroutineContextKey].\n * See [AbstractCoroutineContextKey] for the example of usage.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalStdlibApi\npublic fun <E : Element> Element.getPolymorphicElement(key: Key<E>): E? {\n    if (key is AbstractCoroutineContextKey<*, *>) {\n        @Suppress(\"UNCHECKED_CAST\")\n        return if (key.isSubKey(this.key)) key.tryCast(this) as? E else null\n    }\n    @Suppress(\"UNCHECKED_CAST\")\n    return if (this.key === key) this as E else null\n}\n\n/**\n * Returns empty coroutine context if the element is associated with the given [key] in a polymorphic manner\n * or `null` otherwise.\n * This method returns empty context if either [Element.key] is equal to the given [key] or if the [key] is associated\n * with [Element.key] via [AbstractCoroutineContextKey].\n * See [AbstractCoroutineContextKey] for the example of usage.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalStdlibApi\npublic fun Element.minusPolymorphicKey(key: Key<*>): CoroutineContext {\n    if (key is AbstractCoroutineContextKey<*, *>) {\n        return if (key.isSubKey(this.key) && key.tryCast(this) != null) EmptyCoroutineContext else this\n    }\n    return if (this.key === key) EmptyCoroutineContext else this\n}\n\n/**\n * An empty coroutine context.\n */\n@SinceKotlin(\"1.3\")\npublic object EmptyCoroutineContext : CoroutineContext, Serializable {\n    private const val serialVersionUID: Long = 0\n    private fun readResolve(): Any = EmptyCoroutineContext\n\n    public override fun <E : Element> get(key: Key<E>): E? = null\n    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R = initial\n    public override fun plus(context: CoroutineContext): CoroutineContext = context\n    public override fun minusKey(key: Key<*>): CoroutineContext = this\n    public override fun hashCode(): Int = 0\n    public override fun toString(): String = \"EmptyCoroutineContext\"\n}\n\n//---------- internal impl ----------\n\n// this class is not exposed, but is hidden inside implementations\n// this is a left-biased list, so that `plus` works naturally\n@SinceKotlin(\"1.3\")\ninternal class CombinedContext(\n    private val left: CoroutineContext,\n    private val element: Element\n) : CoroutineContext, Serializable {\n\n    override fun <E : Element> get(key: Key<E>): E? {\n        var cur = this\n        while (true) {\n            cur.element[key]?.let { return it }\n            val next = cur.left\n            if (next is CombinedContext) {\n                cur = next\n            } else {\n                return next[key]\n            }\n        }\n    }\n\n    public override fun <R> fold(initial: R, operation: (R, Element) -> R): R =\n        operation(left.fold(initial, operation), element)\n\n    public override fun minusKey(key: Key<*>): CoroutineContext {\n        element[key]?.let { return left }\n        val newLeft = left.minusKey(key)\n        return when {\n            newLeft === left -> this\n            newLeft ===
```

```
EmptyCoroutineContext -> element\n            else -> CombinedContext(newLeft, element)\n        }\n    }\n\n    private fun size(): Int {\n        var cur = this\n        var size = 2\n        while (true) {\n            cur = cur.left as? CombinedContext ?: return size\n            size++\n        }\n    }\n\n    private fun contains(element: Element): Boolean =\n        get(element.key) == element\n\n    private fun containsAll(context: CombinedContext): Boolean {\n        var cur = context\n        while (true) {\n            if (!contains(cur.element)) return false\n            val next = cur.left\n            if (next is CombinedContext) {\n                cur = next\n            } else {\n                return contains(next as Element)\n            }\n        }\n    }\n\n    override fun equals(other: Any?): Boolean =\n        this === other || other is CombinedContext && other.size() == size() && other.containsAll(this)\n\n    override fun hashCode(): Int = left.hashCode() + element.hashCode()\n\n    override fun toString(): String =\n        \"[\" + fold(\"\") { acc, element ->\n            if (acc.isEmpty()) element.toString() else \"$acc, $element\"\n        } + \"]\"\n\n    private fun writeReplace(): Any {\n        val n = size()\n        val elements = arrayOfNulls<CoroutineContext>(n)\n        var index = 0\n        fold(Unit) { _, element -> elements[index++] = element }\n        check(index == n)\n        @Suppress(\"UNCHECKED_CAST\")\n        return Serialized(elements as Array<CoroutineContext>)\n    }\n\n    private class Serialized(val elements: Array<CoroutineContext>) : Serializable {\n        companion object {\n            private const val serialVersionUID: Long = 0L\n        }\n\n        private fun readResolve(): Any = elements.fold(EmptyCoroutineContext, CoroutineContext::plus)\n    }\n}\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmName(\"IntrinsicsKt\")\n@file:kotlin.jvm.JvmMultifileClass\n\npackage kotlin.coroutines.intrinsics\n\nimport kotlin.contracts.*\nimport kotlin.coroutines.*\nimport kotlin.internal.InlineOnly\n\n/**\n * Obtains the current continuation instance inside suspend functions and either suspends\n * currently running coroutine or returns result immediately without suspension.\n *\n * If the [block] returns the special [COROUTINE_SUSPENDED] value, it means that suspend function did suspend the execution and will\n * not return any result immediately. In this case, the [Continuation] provided to the [block] shall be\n * resumed by invoking [Continuation.resumeWith] at some moment in the\n * future when the result becomes available to resume the computation.\n *\n * Otherwise, the return value of the [block] must have a type assignable to [T] and represents the result of this suspend function.\n * It means that the execution was not suspended and the [Continuation] provided to the [block] shall not be invoked.\n * As the result type of the [block] is declared as `Any?` and cannot be correctly type-checked,\n * its proper return type remains on the conscience of the suspend function's author.\n *\n * Invocation of [Continuation.resumeWith] resumes coroutine directly in the invoker's thread without going through the\n * [ContinuationInterceptor] that might be present in the coroutine's [CoroutineContext].\n * It is the invoker's responsibility to ensure that a proper invocation context is established.\n * [Continuation.intercepted] can be used to acquire the intercepted continuation.\n *\n * Note that it is not recommended to call either [Continuation.resume] nor [Continuation.resumeWithException] functions synchronously\n * in the same stackframe where suspension function is run. Use [suspendCoroutine] as a safer way to obtain current\n * continuation instance.\n */\n@SinceKotlin(\"1.3\")\n@InlineOnly\n@Suppress(\"UNUSED_PARAMETER\", \"RedundantSuspendModifier\")\npublic suspend inline fun <T> suspendCoroutineUninterceptedOrReturn(crossinline block: (Continuation<T>) -> Any?): T {\n    contract { callsInPlace(block, InvocationKind.EXACTLY_ONCE) }\n    throw NotImplementedError(\"Implementation of suspendCoroutineUninterceptedOrReturn is intrinsic\")\n}\n\n/**\n * This value is used as a return value of [suspendCoroutineUninterceptedOrReturn] `block` argument to state that\n * the execution was suspended and will not return any result immediately.\n *\n * **Note: this value should not be used in general code.** Using it outside of the context of\n * `suspendCoroutineUninterceptedOrReturn` function return value (including, but not limited to,\n * storing this value in other properties, returning it from other functions, etc)\n * can lead to unspecified behavior of the code.\n */\n// It is implemented as property with getter to avoid ProGuard
```

<clinit> problem with multifile IntrinsicsKt class\n@SinceKotlin(\"1.3\")\npublic val COROUTINE_SUSPENDED: Any get() = CoroutineSingletons.COROUTINE_SUSPENDED\n\n// Using enum here ensures two important properties:\n// 1. It makes SafeContinuation serializable with all kinds of serialization frameworks (since all of them natively support enums)\n// 2. It improves debugging experience, since you clearly see toString() value of those objects and what package they come from\n@SinceKotlin(\"1.3\")\n@PublishedApi // This class is Published API via serialized representation of SafeContinuation, don't rename/move\ninternal enum class CoroutineSingletons { COROUTINE_SUSPENDED, UNDECIDED, RESUMED }\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.enums\n\nimport kotlin.jvm.Volatile\n\n/**\n * A specialized immutable implementation of [List] interface that\n * contains all enum entries of the specified enum type [E].\n * [EnumEntries] contains all enum entries in the order they are declared in the source code,\n * consistently with the corresponding [Enum.ordinal] values.\n *\n * An instance of this interface can only be obtained from `EnumClass.entries` property.\n */\n@ExperimentalStdlibApi\n@SinceKotlin(\"1.8\")\npublic sealed interface EnumEntries<E : Enum<E>> : List<E>\n\n@PublishedApi\n@ExperimentalStdlibApi\n@SinceKotlin(\"1.8\") // Used by JVM compiler\ninternal fun <E : Enum<E>> enumEntries(entriesProvider: () -> Array<E>): EnumEntries<E> = EnumEntriesList(entriesProvider)\n\n@PublishedApi\n@ExperimentalStdlibApi\n@SinceKotlin(\"1.8\") // Used by Native/JS compilers and Java serialization\ninternal fun <E : Enum<E>> enumEntries(entries: Array<E>): EnumEntries<E> = EnumEntriesList { entries }.also {\n   /*\n   * Here we are enforcing initialization of _entries property.\n   * It is required because of two reasons.\n   * 1. In old Native mm the object will be frozen after creation, so it must be immutable\n   * 2. Native doesn't support @Volatile for now, so this initialization is not generally safe, if\n   *   done after object is published.\n   *\n   * This is very implementation-dependent hack, and it should be removed when/if both reasons above are gone.\n   */\n   it.size\n}\n\n/*\n * For enum class E, this class is instantiated in the following manner (NB it's pseudocode that does not\n * reflect code generation strategy precisely):\n * ```\n * class E extends Enum<E> {\n *   private static final E[] $VALUES\n *   private static final EnumEntries[] $ENTRIES\n *\n *   static {\n * $VALUES = $values();\n *   val supplier = #invokedynamic ..args.. values;\n * $ENTRIES = new EnumEntriesList(supplier);\n *   }\n *\n *   public static EnumEntries<MyEnum> getEntries() {\n *   return $ENTRIES;\n *   }\n *\n *   private synthetic static E[] $values() {\n *   return new E[] { ... };\n *   }\n * }\n * ```\n *\n * This machinery is required as a workaround for a long-standing issue when people do reflectively change `$VALUES` of\n * enums in order to workaround project-specific issues.\n * We allow racy initialization (e.g. entriesProvider can be invoked multiple times), but the resulting array is safely\n * published, preventing any read races after the initialization.\n */\n@SinceKotlin(\"1.8\")\n@ExperimentalStdlibApi\nprivate class EnumEntriesList<T : Enum<T>>(private val entriesProvider: () -> Array<T>) : EnumEntries<T>, AbstractList<T>(), Serializable {\n// WA for JS IR bug:\n// class type parameter MUST be different form E (AbstractList<E> type parameter),\n// otherwise the bridge names for contains() and indexOf() will be clashed with the original method names,\n// and produced JS code will not contain type checks and will not work correctly.\n\n   @Volatile // Volatile is required for safe publication of the array. It doesn't incur any real-world penalties\n   private var _entries: Array<T>? = null\n   private val entries: Array<T>\n   get() {\n   var e = _entries\n   if (e != null) return e\n   e = entriesProvider()\n   _entries = e\n   return e\n   }\n\n   override val size: Int\n   get() = entries.size\n\n   override fun get(index: Int): T {\n   val entries = entries\n   checkElementIndex(index, entries.size)\n   return entries[index]\n   }\n\n   // By definition, EnumEntries contains **all** enums in declaration order,\n   // thus we are able to short-circuit the implementation here\n\n   override fun contains(element: T): Boolean {\n   @Suppress(\"SENSELESS_COMPARISON\")\n   if (element === null) return false // WA for JS IR bug\n   // Check identity due to UnsafeVariance\n   val target = entries.getOrNull(element.ordinal)\n   return target === element\n   }\n\n   override fun indexOf(element: T): Int {\n   @Suppress(\"SENSELESS_COMPARISON\")\n

if (element === null) return -1 // WA for JS IR bug\n // Check identity due to UnsafeVariance\n val ordinal = element.ordinal\n val target = entries.getOrNull(ordinal)\n return if (target === element) ordinal else -1\n }\n\n override fun lastIndexOf(element: T): Int = indexOf(element)\n\n @Suppress(\"unused\") // Used for Java serialization\n private fun writeReplace(): Any {\n return EnumEntriesSerializationProxy(entries)\n }\n}\n\ninternal expect class EnumEntriesSerializationProxy<E : Enum<E>>(entries: Array<E>)\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.experimental\n\n/**\n * This annotation marks the experimental [ObjCName][kotlin.native.ObjCName] annotation.\n */\n@RequiresOptIn\n@Target(AnnotationTarget.ANNOTATION_CLASS)\n@Retention(AnnotationRetention.BINARY)\n@MustBeDocumented\npublic annotation class ExperimentalObjCName\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.experimental\n\n/**\n * This annotation marks the experimental Objective-C export refinement annotations.\n */\n@RequiresOptIn\n@Target(AnnotationTarget.ANNOTATION_CLASS)\n@Retention(AnnotationRetention.BINARY)\n@MustBeDocumented\npublic annotation class ExperimentalObjCRefinement\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.experimental\n\n/** Performs a bitwise AND operation between the two values. */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline infix fun Byte.and(other: Byte): Byte = (this.toInt() and other.toInt()).toByte()\n\n/** Performs a bitwise OR operation between the two values. */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline infix fun Byte.or(other: Byte): Byte = (this.toInt() or other.toInt()).toByte()\n\n/** Performs a bitwise XOR operation between the two values. */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline infix fun Byte.xor(other: Byte): Byte = (this.toInt() xor other.toInt()).toByte()\n\n/** Inverts the bits in this value. */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline fun Byte.inv(): Byte = (this.toInt().inv()).toByte()\n\n\n/** Performs a bitwise AND operation between the two values. */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.and(other: Short): Short = (this.toInt() and other.toInt()).toShort()\n\n/** Performs a bitwise OR operation between the two values. */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.or(other: Short): Short = (this.toInt() or other.toInt()).toShort()\n\n/** Performs a bitwise XOR operation between the two values. */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline infix fun Short.xor(other: Short): Short = (this.toInt() xor other.toInt()).toShort()\n\n/** Inverts the bits in this value. */\n@SinceKotlin(\"1.1\")\n@kotlin.internal.InlineOnly\npublic inline fun Short.inv(): Short = (this.toInt().inv()).toShort()\n\n\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.experimental\n\n/**\n * The experimental marker for type inference augmenting annotations.\n *\n * Any usage of a declaration annotated with `@ExperimentalTypeInference` must be accepted either by\n * annotating that usage with the [OptIn] annotation, e.g. `@OptIn(ExperimentalTypeInference::class)`,\n * or by using the compiler argument `-opt-in=kotlin.experimental.ExperimentalTypeInference`.\n */\n@RequiresOptIn(level = RequiresOptIn.Level.ERROR)\n@MustBeDocumented\n@Retention(AnnotationRetention.BINARY)\n@Target(AnnotationTarget.ANNOTATION_CLASS)\n@SinceKotlin(\"1.3\")\npublic annotation class ExperimentalTypeInference\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.internal\n\n/**\n * Specifies that the corresponding type should be ignored during type inference.\n

*/\n@Target(AnnotationTarget.TYPE)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class NoInfer\n\n/**\n * Specifies that the constraint built for the type during type inference should be an equality one.\n */\n@Target(AnnotationTarget.TYPE)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class Exact\n\n/**\n * Specifies that a corresponding member has the lowest priority in overload resolution.\n */\n@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY, AnnotationTarget.CONSTRUCTOR)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class LowPriorityInOverloadResolution\n\n/**\n * Specifies that the corresponding member has the highest priority in overload resolution. Effectively this means that\n * an extension annotated with this annotation will win in overload resolution over a member with the same signature.\n */\n@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class HidesMembers\n\n/**\n * The value of this type parameter should be mentioned in input types (argument types, receiver type or expected type).\n */\n@Target(AnnotationTarget.TYPE_PARAMETER)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class OnlyInputTypes\n\n/**\n * Specifies that this function should not be called directly without inlining\n */\n@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY, AnnotationTarget.PROPERTY_GETTER, AnnotationTarget.PROPERTY_SETTER)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class InlineOnly\n\n/**\n * Specifies that this declaration can have dynamic receiver type.\n */\n@Target(AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY)\n@Retention(AnnotationRetention.BINARY)\ninternal annotation class DynamicExtension\n\n/**\n * The value of this parameter should be a property reference expression (`this::foo`), referencing a `lateinit` property,\n * the backing field of which is accessible at the point where the corresponding argument is passed.\n */\n@Target(AnnotationTarget.VALUE_PARAMETER)\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin(\"1.2\")\ninternal annotation class AccessibleLateinitPropertyLiteral\n\n/**\n * Specifies that this declaration is only completely supported since the specified version.\n *\n * The Kotlin compiler of an earlier version is going to report a diagnostic on usages of this declaration.\n * The diagnostic message can be specified with [message], or via [errorCode] (takes less space, but might not be immediately clear\n * to the user). The diagnostic severity can be specified with [level]: WARNING/ERROR mean that either a warning or an error\n * is going to be reported, HIDDEN means that the declaration is going to be removed from resolution completely.\n *\n * [versionKind] specifies which version should be compared with the [version] value, when compiling the usage of the annotated declaration.\n * Note that prior to 1.2, only [RequireKotlinVersionKind.LANGUAGE_VERSION] was supported, so the Kotlin compiler before 1.2 is going to\n * treat any [RequireKotlin] as if it requires the language version. Since 1.2, the Kotlin compiler supports\n * [RequireKotlinVersionKind.LANGUAGE_VERSION], [RequireKotlinVersionKind.COMPILER_VERSION] and [RequireKotlinVersionKind.API_VERSION].\n * If the actual value of [versionKind] is something different (e.g. a new version kind, added in future versions of Kotlin),\n * Kotlin 1.2 is going to ignore this [RequireKotlin] altogether, where as Kotlin before 1.2 is going to treat this as a requirement\n * on the language version.\n *\n * This annotation is erased at compile time; its arguments are stored in a more compact form in the Kotlin metadata.\n */\n@Target(AnnotationTarget.CLASS, AnnotationTarget.FUNCTION, AnnotationTarget.PROPERTY, AnnotationTarget.CONSTRUCTOR, AnnotationTarget.TYPEALIAS)\n@Retention(AnnotationRetention.SOURCE)\n@Repeatable\n@SinceKotlin(\"1.2\")\ninternal annotation class RequireKotlin(\n   val version: String,\n   val message: String = \"\",\n   val level: DeprecationLevel = DeprecationLevel.ERROR,\n   val versionKind: RequireKotlinVersionKind = RequireKotlinVersionKind.LANGUAGE_VERSION,\n   val errorCode: Int = -1\n)\n\n/**\n * The kind of the version that is required by [RequireKotlin].\n */\n@SinceKotlin(\"1.2\")\ninternal enum class

RequireKotlinVersionKind {\n LANGUAGE_VERSION,\n COMPILER_VERSION,\n API_VERSION,\n}\n\n/**\n * Specifies that this declaration is a part of special DSL, used for constructing function's contract.\n */\n@Retention(AnnotationRetention.BINARY)\n@SinceKotlin(\"1.2\")\ninternal annotation class ContractsDsl\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.internal\n\n// a mod b (in arithmetical sense)\nprivate fun mod(a: Int, b: Int): Int {\n val mod = a % b\n return if (mod >= 0) mod else mod + b\n}\n\nprivate fun mod(a: Long, b: Long): Long {\n val mod = a % b\n return if (mod >= 0) mod else mod + b\n}\n\n// (a - b) mod c\nprivate fun differenceModulo(a: Int, b: Int, c: Int): Int {\n return mod(mod(a, c) - mod(b, c), c)\n}\n\nprivate fun differenceModulo(a: Long, b: Long, c: Long): Long {\n return mod(mod(a, c) - mod(b, c), c)\n}\n\n/**\n * Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in the range\n * from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative\n * [step].\n *\n * No validation on passed parameters is performed. The given parameters should satisfy the condition:\n *\n * - either `step > 0` and `start <= end`,\n * - or `step < 0` and `start >= end`.\n *\n * @param start first element of the progression\n * @param end ending bound for the progression\n * @param step increment, or difference of successive elements in the progression\n * @return the final element of the progression\n * @suppress\n */\n@PublishedApi\ninternal fun getProgressionLastElement(start: Int, end: Int, step: Int): Int = when {\n step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step)\n step < 0 -> if (start <= end) end else end + differenceModulo(start, end, -step)\n else -> throw kotlin.IllegalArgumentException(\"Step is zero.\")\n}\n\n/**\n * Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in the range\n * from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative\n * [step].\n *\n * No validation on passed parameters is performed. The given parameters should satisfy the condition:\n *\n * - either `step > 0` and `start <= end`,\n * - or `step < 0` and `start >= end`.\n *\n * @param start first element of the progression\n * @param end ending bound for the progression\n * @param step increment, or difference of successive elements in the progression\n * @return the final element of the progression\n * @suppress\n */\n@PublishedApi\ninternal fun getProgressionLastElement(start: Long, end: Long, step: Long): Long = when {\n step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step)\n step < 0 -> if (start <= end) end else end + differenceModulo(start, end, -step)\n else -> throw kotlin.IllegalArgumentException(\"Step is zero.\")\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.properties\n\nimport kotlin.reflect.KProperty\n\n/**\n * Standard property delegates.\n */\npublic object Delegates {\n /**\n * Returns a property delegate for a read/write property with a non-`null` value that is initialized not during\n * object construction time but at a later time. Trying to read the property before the initial value has been\n * assigned results in an exception.\n *\n * @sample samples.properties.Delegates.notNullDelegate\n */\n public fun <T : Any> notNull(): ReadWriteProperty<Any?, T> = NotNullVar()\n\n /**\n * Returns a property delegate for a read/write property that calls a specified callback function when changed.\n * @param initialValue the initial value of the property.\n * @param onChange the callback which is called after the change of the property is made. The value of the property\n * has already been changed when this callback is invoked.\n *\n * @sample samples.properties.Delegates.observableDelegate\n */\n public inline fun <T> observable(initialValue: T, crossinline onChange: (property: KProperty<*>, oldValue: T, newValue: T) -> Unit):\n ReadWriteProperty<Any?, T> =\n object : ObservableProperty<T>(initialValue) {\n override fun afterChange(property: KProperty<*>, oldValue: T, newValue: T) = onChange(property, oldValue, newValue)\n }\n\n /**\n * Returns a property delegate for a read/write property that calls a specified callback function when changed,\n * allowing the callback to veto the modification.\n * @param initialValue the initial value of the property.\n * @param onChange the callback which is called before a change to the property value is attempted.\n

```
 *  The value of the property hasn't been changed yet, when this callback is invoked.\n     *  If the callback returns `true` the value of the property is being set to the new value,\n     *  and if the callback returns `false` the new value is discarded and the property remains its old value.\n     *\n     *  @sample samples.properties.Delegates.vetoableDelegate\n     *  @sample samples.properties.Delegates.throwVetoableDelegate\n     */\n    public inline fun <T> vetoable(initialValue: T, crossinline onChange: (property: KProperty<*>, oldValue: T, newValue: T) -> Boolean):\n ReadWriteProperty<Any?, T> =\n        object : ObservableProperty<T>(initialValue) {\n            override fun beforeChange(property: KProperty<*>, oldValue: T, newValue: T): Boolean = onChange(property, oldValue, newValue)\n        }\n\n}\n\n\nprivate class NotNullVar<T : Any>() : ReadWriteProperty<Any?, T> {\n    private var value: T? = null\n\n    public override fun getValue(thisRef: Any?, property: KProperty<*>): T {\n        return value ?: throw IllegalStateException(\"Property ${property.name} should be initialized before get.\")\n    }\n\n    public override fun setValue(thisRef: Any?, property: KProperty<*>, value: T) {\n        this.value = value\n    }\n}\n\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.properties\n\nimport kotlin.reflect.KProperty\n\n/**\n * Base interface that can be used for implementing property delegates of read-only properties.\n *\n * This is provided only for convenience; you don't have to extend this interface\n * as long as your property delegate has methods with the same signatures.\n *\n * @param T the type of object which owns the delegated property.\n * @param V the type of the property value.\n */\npublic fun interface ReadOnlyProperty<in T, out V> {\n    /**\n     * Returns the value of the property for the given object.\n     * @param thisRef the object for which the value is requested.\n     * @param property the metadata for the property.\n     * @return the property value.\n     */\n    public operator fun getValue(thisRef: T, property: KProperty<*>): V\n}\n\n/**\n * Base interface that can be used for implementing property delegates of read-write properties.\n *\n * This is provided only for convenience; you don't have to extend this interface\n * as long as your property delegate has methods with the same signatures.\n *\n * @param T the type of object which owns the delegated property.\n * @param V the type of the property value.\n */\npublic interface ReadWriteProperty<in T, V> : ReadOnlyProperty<T, V> {\n    /**\n     * Returns the value of the property for the given object.\n     * @param thisRef the object for which the value is requested.\n     * @param property the metadata for the property.\n     * @return the property value.\n     */\n    public override operator fun getValue(thisRef: T, property: KProperty<*>): V\n\n    /**\n     * Sets the value of the property for the given object.\n     * @param thisRef the object for which the value is requested.\n     * @param property the metadata for the property.\n     * @param value the value to set.\n     */\n    public operator fun setValue(thisRef: T, property: KProperty<*>, value: V)\n}\n\n/**\n * Base interface that can be used for implementing property delegate providers.\n *\n * This is provided only for convenience; you don't have to extend this interface\n * as long as your delegate provider has a method with the same signature.\n *\n * @param T the type of object which owns the delegated property.\n * @param D the type of property delegates this provider provides.\n */\n@SinceKotlin(\"1.4\")\npublic fun interface PropertyDelegateProvider<in T, out D> {\n    /**\n     * Returns the delegate of the property for the given object.\n     *\n     * This function can be used to extend the logic of creating the object (e.g. perform validation checks)\n     * to which the property implementation is delegated.\n     *\n     * @param thisRef the object for which property delegate is requested.\n     * @param property the metadata for the property.\n     * @return the property delegate.\n     */\n    public operator fun provideDelegate(thisRef: T, property: KProperty<*>): D\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.properties\n\nimport kotlin.reflect.KProperty\n\n/**\n * Implements the core logic of a property delegate for a read/write property that calls callback functions when changed.\n * @param initialValue the initial value of the property.\n */\npublic abstract class ObservableProperty<V>(initialValue: V) : ReadWriteProperty<Any?, V> {\n    private var value = initialValue\n\n    /**\n     *  The callback which is called before a change to the property value
```

```
is attempted.\n     *  The value of the property hasn't been changed yet, when this callback is invoked.\n     *  If the callback returns `true` the value of the property is being set to the new value,\n     *  and if the callback returns `false` the new value is discarded and the property remains its old value.\n     */\n    protected open fun beforeChange(property: KProperty<*>, oldValue: V, newValue: V): Boolean = true\n\n    /**\n     * The callback which is called after the change of the property is made. The value of the property\n     * has already been changed when this callback is invoked.\n     */\n    protected open fun afterChange(property: KProperty<*>, oldValue: V, newValue: V): Unit {}\n\n    public override fun getValue(thisRef: Any?, property: KProperty<*>): V {\n        return value\n    }\n\n    public override fun setValue(thisRef: Any?, property: KProperty<*>, value: V) {\n        val oldValue = this.value\n        if (!beforeChange(property, oldValue, value)) {\n            return\n        }\n        this.value = value\n        afterChange(property, oldValue, value)\n    }\n}","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:Suppress(\"PackageDirectoryMismatch\")\npackage kotlin\n\nimport kotlin.reflect.*\n\n/**\n * An extension operator that allows delegating a read-only property of type [V]\n * to a property reference to a property of type [V] or its subtype.\n *\n * @receiver A property reference to a read-only or mutable property of type [V] or its subtype.\n * The reference is without a receiver, i.e. it either references a top-level property or\n * has the receiver bound to it.\n *\n * Example:\n *\n * ```\n * class Login(val username: String)\n * val defaultLogin = Login(\"Admin\")\n * val defaultUsername by defaultLogin::username\n * // equivalent to\n * val defaultUserName
    get() = defaultLogin.username\n * ```\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline operator fun <V> KProperty0<V>.getValue(thisRef: Any?, property: KProperty<*>): V {\n    return get()\n}\n\n/**\n * An extension operator that allows delegating a mutable property of type [V]\n * to a property reference to a mutable property of the same type [V].\n *\n * @receiver A property reference to a mutable property of type [V].\n * The reference is without a receiver, i.e. it either references a top-level property or\n * has the receiver bound to it.\n *\n * Example:\n *\n * ```\n * class Login(val username: String, var incorrectAttemptCounter: Int = 0)\n * val defaultLogin = Login(\"Admin\")\n * var defaultLoginAttempts by defaultLogin::incorrectAttemptCounter\n * // equivalent to\n * var defaultLoginAttempts: Int\n *     get() = defaultLogin.incorrectAttemptCounter\n *     set(value) { defaultLogin.incorrectAttemptCounter = value }\n * ```\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic
 inline operator fun <V> KMutableProperty0<V>.setValue(thisRef: Any?, property: KProperty<*>, value: V) {\n    set(value)\n}\n\n\n/**\n * An extension operator that allows delegating a read-only member or extension property of type [V]\n * to a property reference to a member or extension property of type [V] or its subtype.\n *\n * @receiver A property reference to a read-only or mutable property of type [V] or its subtype.\n * The reference has an unbound receiver of type [T].\n *\n * Example:\n *\n * ```\n * class Login(val username: String)\n * val Login.user by Login::username\n * // equivalent to\n * val Login.user get() = this.username\n * ```\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline operator fun <T, V> KProperty1<T, V>.getValue(thisRef: T, property: KProperty<*>): V {\n    return get(thisRef)\n}\n\n/**\n * An extension operator that allows delegating a mutable member or extension property of type [V]\n *
 to a property reference to a member or extension mutable property of the same type [V].\n *\n * @receiver A property reference to a read-only or mutable property of type [V] or its subtype.\n * The reference has an unbound receiver of type [T].\n *\n * Example:\n *\n * ```\n * class Login(val username: String, var incorrectAttemptCounter: Int)\n * var Login.attempts by Login::incorrectAttemptCounter\n * // equivalent to\n * var Login.attempts: Int\n *     get() = this.incorrectAttemptCounter\n *     set(value) { this.incorrectAttemptCounter = value }\n * ```\n */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline operator fun <T, V> KMutableProperty1<T, V>.setValue(thisRef: T, property: KProperty<*>, value: V) {\n    set(thisRef, value)\n}","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.random\n\nimport
```

```
kotlin.math.nextDown\n\n/**\n * An abstract class that is implemented by random number generator algorithms.\n *\n * The companion object [Random.Default] is the default instance of [Random].\n *\n * To get a seeded instance of random generator use [Random] function.\n *\n * @sample samples.random.Randoms.defaultRandom\n */\n@SinceKotlin(\"1.3\")\npublic abstract class Random {\n\n    /**\n     * Gets the next random [bitCount] number of bits.\n     *\n     * Generates an `Int` whose lower [bitCount] bits are filled with random values and the remaining upper bits are zero.\n     *\n     * @param bitCount number of bits to generate, must be in range 0..32, otherwise the behavior is unspecified.\n     *\n     * @sample samples.random.Randoms.nextBits\n     */\n    public abstract fun nextBits(bitCount: Int): Int\n\n    /**\n     * Gets the next random `Int` from the random number generator.\n     *\n     * Generates an `Int` random value uniformly distributed between `Int.MIN_VALUE` and `Int.MAX_VALUE` (inclusive).\n     *\n     * @sample samples.random.Randoms.nextInt\n     */\n    public open fun nextInt(): Int = nextBits(32)\n\n    /**\n     * Gets the next random non-negative `Int` from the random number generator less than the specified [until] bound.\n     *\n     * Generates an `Int` random value uniformly distributed between `0` (inclusive) and the specified [until] bound (exclusive).\n     *\n     * @param until must be positive.\n     *\n     * @throws IllegalArgumentException if [until] is negative or zero.\n     *\n     * @sample samples.random.Randoms.nextIntFromUntil\n     */\n    public open fun nextInt(until: Int): Int = nextInt(0, until)\n\n    /**\n     * Gets the next random `Int` from the random number generator in the specified range.\n     *\n     * Generates an `Int` random value uniformly distributed between the specified [from] (inclusive) and [until] (exclusive) bounds.\n     *\n     * @throws IllegalArgumentException if [from] is greater than or equal to [until].\n     *\n     * @sample samples.random.Randoms.nextIntFromUntil\n     */\n    public open fun nextInt(from: Int, until: Int): Int {\n        checkRangeBounds(from, until)\n        val n = until - from\n        if (n > 0 || n == Int.MIN_VALUE) {\n            val rnd = if (n and -n == n) {\n                val bitCount = fastLog2(n)\n                nextBits(bitCount)\n            } else {\n                var v: Int\n                do {\n                    val bits = nextInt().ushr(1)\n                    v = bits % n\n                } while (bits - v + (n - 1) < 0)\n                v\n            }\n            return from + rnd\n        } else {\n            while (true) {\n                val rnd = nextInt()\n                if (rnd in from until until) return rnd\n            }\n        }\n    }\n\n    /**\n     * Gets the next random `Long` from the random number generator.\n     *\n     * Generates a `Long` random value uniformly distributed between `Long.MIN_VALUE` and `Long.MAX_VALUE` (inclusive).\n     *\n     * @sample samples.random.Randoms.nextLong\n     */\n    public open fun nextLong(): Long = nextInt().toLong().shl(32) + nextInt()\n\n    /**\n     * Gets the next random non-negative `Long` from the random number generator less than the specified [until] bound.\n     *\n     * Generates a `Long` random value uniformly distributed between `0` (inclusive) and the specified [until] bound (exclusive).\n     *\n     * @param until must be positive.\n     *\n     * @throws IllegalArgumentException if [until] is negative or zero.\n     *\n     * @sample samples.random.Randoms.nextLongFromUntil\n     */\n    public open fun nextLong(until: Long): Long = nextLong(0, until)\n\n    /**\n     * Gets the next random `Long` from the random number generator in the specified range.\n     *\n     * Generates a `Long` random value uniformly distributed between the specified [from] (inclusive) and [until] (exclusive) bounds.\n     *\n     * @throws IllegalArgumentException if [from] is greater than or equal to [until].\n     *\n     * @sample samples.random.Randoms.nextLongFromUntil\n     */\n    public open fun nextLong(from: Long, until: Long): Long {\n        checkRangeBounds(from, until)\n        val n = until - from\n        if (n > 0) {\n            val rnd: Long\n            if (n and -n == n) {\n                val nLow = n.toInt()\n                val nHigh = (n ushr 32).toInt()\n                rnd = when {\n                    nLow != 0 -> {\n                        val bitCount = fastLog2(nLow)\n                        // toUInt().toLong()\n                        nextBits(bitCount).toLong() and 0xFFFF_FFFF\n                    }\n                    nHigh == 1 ->\n                        // toUInt().toLong()\n                        nextInt().toLong() and 0xFFFF_FFFF\n                    else -> {\n                        val bitCount = fastLog2(nHigh)\n                        nextBits(bitCount).toLong().shl(32) + (nextInt().toLong() and 0xFFFF_FFFF)\n                    }\n                }\n            } else {\n                var v: Long\n                do {\n                    val bits = nextLong().ushr(1)\n                    v = bits % n\n                } while (bits - v + (n - 1) < 0)\n                rnd = v\n            }\n            return from + rnd\n        } else {\n            while (true) {\n                val rnd = nextLong()\n                if (rnd in from until until) return rnd\n            }\n        }\n    }\n\n    /**\n     * Gets the
```

```
next random [Boolean] value.\n     *\n     * @sample samples.random.Randoms.nextBoolean\n     */\n    public open fun nextBoolean(): Boolean = nextBits(1) != 0\n\n    /**\n     * Gets the next random [Double] value uniformly distributed between 0 (inclusive) and 1 (exclusive).\n     *\n     * @sample samples.random.Randoms.nextDouble\n     */\n    public open fun nextDouble(): Double = doubleFromParts(nextBits(26), nextBits(27))\n\n    /**\n     * Gets the next random non-negative `Double` from the random number generator less than the specified [until] bound.\n     *\n     * Generates a `Double` random value uniformly distributed between 0 (inclusive) and [until] (exclusive).\n     *\n     * @throws IllegalArgumentException if [until] is negative or zero.\n     *\n     * @sample samples.random.Randoms.nextDoubleFromUntil\n     */\n    public open fun nextDouble(until: Double): Double = nextDouble(0.0, until)\n\n    /**\n     * Gets the next random `Double` from the random number generator in the specified range.\n     *\n     * Generates a `Double` random value uniformly distributed between the specified [from] (inclusive) and [until] (exclusive) bounds.\n     *\n     * [from] and [until] must be finite otherwise the behavior is unspecified.\n     *\n     * @throws IllegalArgumentException if [from] is greater than or equal to [until].\n     *\n     * @sample samples.random.Randoms.nextDoubleFromUntil\n     */\n    public open fun nextDouble(from: Double, until: Double): Double {\n        checkRangeBounds(from, until)\n        val size = until - from\n        val r = if (size.isInfinite() && from.isFinite() && until.isFinite()) {\n            val r1 = nextDouble() * (until / 2 - from / 2)\n            from + r1 + r1\n        } else {\n            from + nextDouble() * size\n        }\n        return if (r >= until) until.nextDown() else r\n    }\n\n    /**\n     * Gets the next random [Float] value uniformly distributed between 0 (inclusive) and 1 (exclusive).\n     *\n     * @sample samples.random.Randoms.nextFloat\n     */\n    public open fun nextFloat(): Float = nextBits(24) / (1 shl 24).toFloat()\n\n    /**\n     * Fills a subrange of the specified byte [array] starting from [fromIndex] inclusive and ending [toIndex] exclusive\n     * with random bytes.\n     *\n     * @return [array] with the subrange filled with random bytes.\n     *\n     * @sample samples.random.Randoms.nextBytes\n     */\n    public open fun nextBytes(array: ByteArray, fromIndex: Int = 0, toIndex: Int = array.size): ByteArray {\n        require(fromIndex in 0..array.size && toIndex in 0..array.size) { \"fromIndex ($fromIndex) or toIndex ($toIndex) are out of range: 0..${array.size}.\" }\n        require(fromIndex <= toIndex) { \"fromIndex ($fromIndex) must be not greater than toIndex ($toIndex).\" }\n\n        val steps = (toIndex - fromIndex) / 4\n\n        var position = fromIndex\n        repeat(steps) {\n            val v = nextInt()\n            array[position] = v.toByte()\n            array[position + 1] = v.ushr(8).toByte()\n            array[position + 2] = v.ushr(16).toByte()\n            array[position + 3] = v.ushr(24).toByte()\n            position += 4\n        }\n\n        val remainder = toIndex - position\n        val vr = nextBits(remainder * 8)\n        for (i in 0 until remainder) {\n            array[position + i] = vr.ushr(i * 8).toByte()\n        }\n\n        return array\n    }\n\n    /**\n     * Fills the specified byte [array] with random bytes and returns it.\n     *\n     * @return [array] filled with random bytes.\n     *\n     * @sample samples.random.Randoms.nextBytes\n     */\n    public open fun nextBytes(array: ByteArray): ByteArray = nextBytes(array, 0, array.size)\n\n    /**\n     * Creates a byte array of the specified [size], filled with random bytes.\n     *\n     * @sample samples.random.Randoms.nextBytes\n     */\n    public open fun nextBytes(size: Int): ByteArray = nextBytes(ByteArray(size))\n\n\n    /**\n     * The default random number generator.\n     *\n     * On JVM this generator is thread-safe, its methods can be invoked from multiple threads.\n     *\n     * @sample samples.random.Randoms.defaultRandom\n     */\n    companion object Default : Random(), Serializable {\n        private val defaultRandom: Random = defaultPlatformRandom()\n\n        private object Serialized : Serializable {\n            private const val serialVersionUID = 0L\n\n            private fun readResolve(): Any = Random\n        }\n\n        private fun writeReplace(): Any = Serialized\n\n        override fun nextBits(bitCount: Int): Int = defaultRandom.nextBits(bitCount)\n        override fun nextInt(): Int = defaultRandom.nextInt()\n        override fun nextInt(until: Int): Int = defaultRandom.nextInt(until)\n        override fun nextInt(from: Int, until: Int): Int = defaultRandom.nextInt(from, until)\n\n        override fun nextLong(): Long = defaultRandom.nextLong()\n        override fun nextLong(until: Long): Long = defaultRandom.nextLong(until)\n        override fun nextLong(from: Long, until: Long): Long = defaultRandom.nextLong(from, until)\n\n        override fun nextBoolean(): Boolean = defaultRandom.nextBoolean()\n\n        override fun nextDouble(): Double = defaultRandom.nextDouble()\n
```

override fun nextDouble(until: Double): Double = defaultRandom.nextDouble(until)\n override fun nextDouble(from:
Double, until: Double): Double = defaultRandom.nextDouble(from, until)\n\n override fun nextFloat(): Float = defaultRandom.nextFloat()\n\n override fun nextBytes(array: ByteArray): ByteArray = defaultRandom.nextBytes(array)\n override fun nextBytes(size: Int): ByteArray = defaultRandom.nextBytes(size)\n override fun nextBytes(array: ByteArray, fromIndex: Int, toIndex: Int): ByteArray =\n defaultRandom.nextBytes(array, fromIndex, toIndex)\n }\n}\n\n/**\n * Returns a repeatable random number generator seeded with the given [seed] `Int` value.\n *\n * Two generators with the same seed produce the same sequence of values within the same version of Kotlin runtime.\n *\n * *Note:* Future versions of Kotlin may change the algorithm of this seeded number generator so that it will return\n * a sequence of values different from the current one for a given seed.\n *\n * On JVM the returned generator is NOT thread-safe. Do not invoke it from multiple threads
without proper synchronization.\n *\n * @sample samples.random.Randoms.seededRandom\n */\n@SinceKotlin(\"1.3\")\npublic fun Random(seed: Int): Random = XorWowRandom(seed, seed.shr(31))\n\n/**\n * Returns a repeatable random number generator seeded with the given [seed] `Long` value.\n *\n * Two generators with the same seed produce the same sequence of values within the same version of Kotlin runtime.\n *\n * *Note:* Future versions of Kotlin may change the algorithm of this seeded number generator so that it will return\n * a sequence of values different from the current one for a given seed.\n *\n * On JVM the returned generator is NOT thread-safe. Do not invoke it from multiple threads without proper synchronization.\n *\n * @sample samples.random.Randoms.seededRandom\n */\n@SinceKotlin(\"1.3\")\npublic fun Random(seed: Long): Random = XorWowRandom(seed.toInt(), seed.shr(32).toInt())\n\n\n/**\n * Gets the next random `Int` from the random number generator in the specified [range].\n
*\n * Generates an `Int` random value uniformly distributed in the specified [range]:\n * from `range.start` inclusive to `range.endInclusive` inclusive.\n *\n * @throws IllegalArgumentException if [range] is empty.\n */\n@SinceKotlin(\"1.3\")\npublic fun Random.nextInt(range: IntRange): Int = when {\n range.isEmpty() -> throw IllegalArgumentException(\"Cannot get random in empty range: $range\")\n range.last < Int.MAX_VALUE -> nextInt(range.first, range.last + 1)\n range.first > Int.MIN_VALUE -> nextInt(range.first - 1, range.last) + 1\n else -> nextInt()\n}\n\n/**\n * Gets the next random `Long` from the random number generator in the specified [range].\n *\n * Generates a `Long` random value uniformly distributed in the specified [range]:\n * from `range.start` inclusive to `range.endInclusive` inclusive.\n *\n * @throws IllegalArgumentException if [range] is empty.\n */\n@SinceKotlin(\"1.3\")\npublic fun Random.nextLong(range: LongRange): Long = when {\n range.isEmpty()
-> throw IllegalArgumentException(\"Cannot get random in empty range: $range\")\n range.last < Long.MAX_VALUE -> nextLong(range.first, range.last + 1)\n range.first > Long.MIN_VALUE -> nextLong(range.first - 1, range.last) + 1\n else -> nextLong()\n}\n\n\ninternal expect fun defaultPlatformRandom(): Random\ninternal expect fun doubleFromParts(hi26: Int, low27: Int): Double\n\ninternal fun fastLog2(value: Int): Int = 31 - value.countLeadingZeroBits()\n\n/** Takes upper [bitCount] bits (0..32) from this number. */\ninternal fun Int.takeUpperBits(bitCount: Int): Int =\n this.ushr(32 - bitCount) and (-bitCount).shr(31)\n\ninternal fun checkRangeBounds(from: Int, until: Int) = require(until > from) { boundsErrorMessage(from, until) }\ninternal fun checkRangeBounds(from: Long, until: Long) = require(until > from) { boundsErrorMessage(from, until) }\ninternal fun checkRangeBounds(from: Double, until: Double) = require(until > from) { boundsErrorMessage(from, until) }\n\ninternal
fun boundsErrorMessage(from: Any, until: Any) = \"Random range is empty: [$from, $until).\"\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.random\n\n\n/**\n * Gets the next random [UInt] from the random number generator.\n *\n * Generates a [UInt] random value uniformly distributed between [UInt.MIN_VALUE] and [UInt.MAX_VALUE] (inclusive).\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun

Random.nextUInt(): UInt = nextInt().toUInt()\n\n/**\n * Gets the next random [UInt] from the random number generator less than the specified [until] bound.\n *\n * Generates a [UInt] random value uniformly distributed between `0` (inclusive) and the specified [until] bound (exclusive).\n *\n * @throws IllegalArgumentException if [until] is zero.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Random.nextUInt(until: UInt): UInt = nextUInt(0u, until)\n\n/**\n * Gets the next random [UInt] from the random number generator in the specified range.\n *\n * Generates a [UInt] random value uniformly distributed between the specified [from] (inclusive) and [until] (exclusive) bounds.\n *\n * @throws IllegalArgumentException if [from] is greater than or equal to [until].\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Random.nextUInt(from: UInt, until: UInt): UInt {\n checkUIntRangeBounds(from, until)\n\n val signedFrom = from.toInt() xor Int.MIN_VALUE\n val signedUntil = until.toInt() xor Int.MIN_VALUE\n\n val signedResult = nextInt(signedFrom, signedUntil) xor Int.MIN_VALUE\n return signedResult.toUInt()\n}\n\n/**\n * Gets the next random [UInt] from the random number generator in the specified [range].\n *\n * Generates a [UInt] random value uniformly distributed
in the specified [range]:\n * from `range.start` inclusive to `range.endInclusive` inclusive.\n *\n * @throws IllegalArgumentException if [range] is empty.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Random.nextUInt(range: UIntRange): UInt = when {\n range.isEmpty() -> throw IllegalArgumentException(\"Cannot get random in empty range: $range\")\n range.last < UInt.MAX_VALUE -> nextUInt(range.first, range.last + 1u)\n range.first > UInt.MIN_VALUE -> nextUInt(range.first - 1u, range.last) + 1u\n else -> nextUInt()\n}\n\n/**\n * Gets the next random [ULong] from the random number generator.\n *\n * Generates a [ULong] random value uniformly distributed between [ULong.MIN_VALUE] and [ULong.MAX_VALUE] (inclusive).\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Random.nextULong(): ULong = nextLong().toULong()\n\n/**\n * Gets the next random [ULong] from the random number generator
less than the specified [until] bound.\n *\n * Generates a [ULong] random value uniformly distributed between `0` (inclusive) and the specified [until] bound (exclusive).\n *\n * @throws IllegalArgumentException if [until] is zero.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Random.nextULong(until: ULong): ULong = nextULong(0uL, until)\n\n/**\n * Gets the next random [ULong] from the random number generator in the specified range.\n *\n * Generates a [ULong] random value uniformly distributed between the specified [from] (inclusive) and [until] (exclusive) bounds.\n *\n * @throws IllegalArgumentException if [from] is greater than or equal to [until].\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Random.nextULong(from: ULong, until: ULong): ULong {\n checkULongRangeBounds(from, until)\n\n val signedFrom = from.toLong() xor Long.MIN_VALUE\n val signedUntil = until.toLong() xor Long.MIN_VALUE\n\n
val signedResult = nextLong(signedFrom, signedUntil) xor Long.MIN_VALUE\n return signedResult.toULong()\n}\n\n/**\n * Gets the next random [ULong] from the random number generator in the specified [range].\n *\n * Generates a [ULong] random value uniformly distributed in the specified [range]:\n * from `range.start` inclusive to `range.endInclusive` inclusive.\n *\n * @throws IllegalArgumentException if [range] is empty.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun Random.nextULong(range: ULongRange): ULong = when {\n range.isEmpty() -> throw IllegalArgumentException(\"Cannot get random in empty range: $range\")\n range.last < ULong.MAX_VALUE -> nextULong(range.first, range.last + 1u)\n range.first > ULong.MIN_VALUE -> nextULong(range.first - 1u, range.last) + 1u\n else -> nextULong()\n}\n\n/**\n * Fills the specified unsigned byte [array] with random bytes and returns it.\n *\n * @return [array] filled with

random bytes.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun Random.nextUBytes(array: UByteArray): UByteArray {\n nextBytes(array.asByteArray())\n return array\n}\n\n/**\n * Creates an unsigned byte array of the specified [size], filled with random bytes.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun Random.nextUBytes(size: Int): UByteArray = nextBytes(size).asUByteArray()\n\n/**\n * Fills a subrange of the specified `UByte` [array] starting from [fromIndex] inclusive and ending [toIndex] exclusive with random UBytes.\n *\n * @return [array] with the subrange filled with random bytes.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalUnsignedTypes\npublic fun Random.nextUBytes(array: UByteArray, fromIndex: Int = 0, toIndex: Int = array.size): UByteArray {\n nextBytes(array.asByteArray(), fromIndex, toIndex)\n return array\n}\n\n\ninternal fun checkUIntRangeBounds(from: UInt, until: UInt) = require(until > from) { boundsErrorMessage(from, until) }\ninternal fun checkULongRangeBounds(from: ULong, until: ULong) = require(until > from) { boundsErrorMessage(from, until) }\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.random\n\n/**\n * Random number generator, using Marsaglia's \"xorwow\" algorithm\n *\n * Cycles after 2^192 - 2^32 repetitions.\n *\n * For more details, see Marsaglia, George (July 2003). \"Xorshift RNGs\". Journal of Statistical Software. 8 (14). doi:10.18637/jss.v008.i14\n *\n * Available at https://www.jstatsoft.org/v08/i14/paper\n *\n */\ninternal class XorWowRandom internal constructor(\n private var x: Int,\n private var y: Int,\n private var z: Int,\n private var w: Int,\n private var v: Int,\n private var addend: Int\n) : Random(), Serializable {\n\n internal constructor(seed1: Int, seed2: Int) :\n this(seed1, seed2, 0, 0, seed1.inv(), (seed1 shl 10) xor (seed2 ushr 4))\n\n init {\n require((x or y or z or w or v) != 0) { \"Initial state must have at least one non-zero element.\" }\n\n // some trivial seeds can produce several values with zeroes in upper bits, so we discard first 64\n repeat(64) { nextInt() }\n }\n\n override fun nextInt(): Int {\n // Equivalent to the xorxow algorithm\n // From Marsaglia, G. 2003. Xorshift RNGs. J. Statis. Soft. 8, 14, p. 5\n var t = x\n t = t xor (t ushr 2)\n x = y\n y = z\n z = w\n val v0 = v\n w = v0\n t = (t xor (t shl 1)) xor v0 xor (v0 shl 4)\n v = t\n addend += 362437\n return t + addend\n }\n\n override fun nextBits(bitCount: Int): Int =\n nextInt().takeUpperBits(bitCount)\n\n private companion object {\n private const val serialVersionUID: Long = 0L\n }\n}\n","/*\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin.ranges\n\n/**\n * An iterator over a progression of values of type `Char`.\n * @property step the number by which the value is incremented on each step.\n */\ninternal class CharProgressionIterator(first: Char, last: Char, val step: Int) : CharIterator() {\n private val finalElement: Int = last.code\n private var hasNext: Boolean = if (step > 0) first <= last else first >= last\n private var next: Int = if (hasNext) first.code else finalElement\n\n override fun hasNext(): Boolean = hasNext\n\n override fun nextChar(): Char {\n val value = next\n if (value == finalElement) {\n if (!hasNext) throw kotlin.NoSuchElementException()\n hasNext = false\n }\n else {\n next += step\n }\n return value.toChar()\n }\n}\n\n/**\n * An iterator over a progression of values of type `Int`.\n * @property step the number by which the value is incremented on each step.\n */\ninternal class IntProgressionIterator(first: Int, last: Int, val step: Int) : IntIterator() {\n private val finalElement: Int = last\n private var hasNext: Boolean = if (step > 0) first <= last else first >= last\n private var next: Int = if (hasNext) first else finalElement\n\n override fun hasNext(): Boolean = hasNext\n\n override fun nextInt(): Int {\n val value = next\n if (value == finalElement) {\n if (!hasNext) throw kotlin.NoSuchElementException()\n hasNext = false\n }\n else {\n next += step\n }\n return value\n }\n}\n\n/**\n * An iterator over a progression of values of type `Long`.\n * @property step the number by which the value is incremented on each step.\n */\ninternal class LongProgressionIterator(first: Long, last: Long, val step: Long) : LongIterator() {\n private val finalElement: Long = last\n private var hasNext: Boolean = if (step > 0) first <= last else first >= last\n private var next: Long

= if (hasNext) first else finalElement\n\n   override fun hasNext(): Boolean = hasNext\n\n   override fun nextLong(): Long {\n   val value = next\n   if (value == finalElement) {\n   if (!hasNext) throw kotlin.NoSuchElementException()\n   hasNext = false\n   }\n   else {\n   next += step\n   }\n   return value\n   }\n}\n\n","/*\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin.ranges\n\nimport kotlin.internal.getProgressionLastElement\n\n/**\n * A progression of values of type `Char`.\n */\npublic open class CharProgression\n   internal constructor\n   (\n   start: Char,\n   endInclusive: Char,\n   step: Int\n   ) : Iterable<Char> {\n   init {\n   if (step == 0) throw kotlin.IllegalArgumentException(\"Step must be non-zero.\")\n   if (step == Int.MIN_VALUE) throw kotlin.IllegalArgumentException(\"Step must be greater than Int.MIN_VALUE to avoid overflow on negation.\")\n   }\n\n   /**\n   * The first element in the progression.\n   */\n   public val first: Char = start\n\n   /**\n   * The last element in the progression.\n   */\n   public val last: Char = getProgressionLastElement(start.code, endInclusive.code, step).toChar()\n\n   /**\n   * The step of the progression.\n   */\n   public val step: Int = step\n\n   override fun iterator(): CharIterator = CharProgressionIterator(first, last, step)\n\n   /**\n   * Checks if the progression is empty.\n   *\n   * Progression with a positive step is empty if its first element is greater than the last element.\n   * Progression with a negative step is empty if its first element is less than the last element.\n   */\n   public open fun isEmpty(): Boolean = if (step > 0) first > last else first < last\n\n   override fun equals(other: Any?): Boolean =\n   other is CharProgression && (isEmpty() && other.isEmpty() ||\n   first == other.first && last == other.last && step == other.step)\n\n   override fun hashCode(): Int =\n   if (isEmpty()) -1 else (31 * (31 * first.code + last.code) + step)\n\n   override fun toString(): String = if (step > 0) \"$first..$last step $step\" else \"$first downTo $last step ${-step}\"\n\n   companion object {\n   /**\n   * Creates CharProgression within the specified bounds of a closed range.\n   *\n   * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].\n   * In order to go backwards the [step] must be negative.\n   *\n   * [step] must be greater than `Int.MIN_VALUE` and not equal to zero.\n   */\n   public fun fromClosedRange(rangeStart: Char, rangeEnd: Char, step: Int): CharProgression = CharProgression(rangeStart, rangeEnd, step)\n   }\n}\n\n/**\n * A progression of values of type `Int`.\n */\npublic open class IntProgression\n   internal constructor\n   (\n   start: Int,\n   endInclusive: Int,\n   step: Int\n   ) : Iterable<Int> {\n   init {\n   if (step == 0) throw kotlin.IllegalArgumentException(\"Step must be non-zero.\")\n   if (step == Int.MIN_VALUE) throw kotlin.IllegalArgumentException(\"Step must be greater than Int.MIN_VALUE to avoid overflow on negation.\")\n   }\n\n   /**\n   * The first element in the progression.\n   */\n   public val first: Int = start\n\n   /**\n   * The last element in the progression.\n   */\n   public val last: Int = getProgressionLastElement(start, endInclusive, step)\n\n   /**\n   * The step of the progression.\n   */\n   public val step: Int = step\n\n   override fun iterator(): IntIterator = IntProgressionIterator(first, last, step)\n\n   /**\n   * Checks if the progression is empty.\n   *\n   * Progression with a positive step is empty if its first element is greater than the last element.\n   * Progression with a negative step is empty if its first element is less than the last element.\n   */\n   public open fun isEmpty(): Boolean = if (step > 0) first > last else first < last\n\n   override fun equals(other: Any?): Boolean =\n   other is IntProgression && (isEmpty() && other.isEmpty() ||\n   first == other.first && last == other.last && step == other.step)\n\n   override fun hashCode(): Int =\n   if (isEmpty()) -1 else (31 * (31 * first + last) + step)\n\n   override fun toString(): String = if (step > 0) \"$first..$last step $step\" else \"$first downTo $last step ${-step}\"\n\n

    companion object {\n   /**\n   * Creates IntProgression within the specified bounds of a closed range.\n   *\n   * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].\n   * In order to go backwards the [step] must be negative.\n   *\n   * [step] must be greater than `Int.MIN_VALUE` and not equal to zero.\n   */\n   public fun fromClosedRange(rangeStart: Int, rangeEnd: Int, step: Int): IntProgression = IntProgression(rangeStart, rangeEnd, step)\n   }\n}\n\n/**\n * A

progression of values of type `Long`.\n */\npublic open class LongProgression\n internal constructor\n (\n start: Long,\n endInclusive: Long,\n step: Long\n) : Iterable<Long> {\n init {\n if (step == 0L) throw kotlin.IllegalArgumentException(\"Step must be non-zero.\")\n if (step == Long.MIN_VALUE) throw kotlin.IllegalArgumentException(\"Step must be greater than Long.MIN_VALUE to avoid overflow on negation.\")\n }\n\n /**\n * The first element in the progression.\n */\n public val first: Long = start\n\n /**\n * The last element in the progression.\n */\n public val last: Long = getProgressionLastElement(start, endInclusive, step)\n\n /**\n * The step of the progression.\n */\n public val step: Long = step\n\n override fun iterator(): LongIterator = LongProgressionIterator(first, last, step)\n\n /**\n * Checks if the progression is empty.\n *\n * Progression with a positive step is empty if its first element is greater than the last element.\n * Progression with a negative step is empty if its first element is less than the last element.\n */\n public open fun isEmpty(): Boolean = if (step > 0) first > last else first < last\n\n override fun equals(other: Any?): Boolean =\n other is LongProgression && (isEmpty() && other.isEmpty()\n ||\n first == other.first && last == other.last && step == other.step)\n\n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 * (31 * (first xor (first ushr 32)) + (last xor (last ushr 32))) + (step xor (step ushr 32))).toInt()\n\n override fun toString(): String = if (step > 0) \"$first..$last step $step\" else \"$first downTo $last step ${-step}\"\n\n companion object {\n /**\n * Creates LongProgression within the specified bounds of a closed range.\n *\n * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].\n * In order to go backwards the [step] must be negative.\n *\n * [step] must be greater than `Long.MIN_VALUE` and not equal to zero.\n */\n public fun fromClosedRange(rangeStart: Long, rangeEnd: Long, step: Long): LongProgression = LongProgression(rangeStart, rangeEnd, step)\n }\n}\n\n","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.ranges\n\n/**\n * Represents a range of values (for example, numbers or characters) where both the lower and upper bounds are included in the range.\n * See the [Kotlin language documentation](https://kotlinlang.org/docs/reference/ranges.html) for more information.\n */\npublic interface ClosedRange<T : Comparable<T>> {\n /**\n * The minimum value in the range.\n */\n public val start: T\n\n /**\n * The maximum value in the range (inclusive).\n */\n public val endInclusive: T\n\n /**\n * Checks whether the specified [value] belongs to the range.\n *\n * A value belongs to the closed range if it is greater than or equal to the [start] bound and less than or equal to the [endInclusive] bound.\n */\n public operator fun contains(value: T): Boolean = value >= start && value <= endInclusive\n\n /**\n * Checks whether the range is empty.\n *\n * The range is empty if its start value is greater than the end value.\n */\n public fun isEmpty(): Boolean = start > endInclusive\n}\n\n/**\n * Represents a range of values (for example, numbers or characters) where the upper bound is not included in the range.\n * See the [Kotlin language documentation](https://kotlinlang.org/docs/reference/ranges.html) for more information.\n */\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic interface OpenEndRange<T : Comparable<T>> {\n /**\n * The minimum value in the range.\n */\n public val start: T\n\n /**\n * The maximum value in the range (exclusive).\n *\n * @throws IllegalStateException can be thrown if the exclusive end bound cannot be represented\n * with a value of type [T].\n */\n public val endExclusive: T\n\n /**\n * Checks whether the specified [value] belongs to the range.\n *\n * A value belongs to the open-ended range if it is greater than or equal to the [start] bound and strictly less than the [endExclusive] bound.\n */\n public operator fun contains(value: T): Boolean = value >= start && value < endExclusive\n\n /**\n * Checks whether the range is empty.\n *\n * The open-ended range is empty if its start value is greater than or equal to the end value.\n */\n public fun isEmpty(): Boolean = start >= endExclusive\n}","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n

```
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"RangesKt\")\n\npackage kotlin.ranges\n\n/**\n * Represents a range of [Comparable] values.\n */\nprivate open class ComparableRange<T : Comparable<T>>(\n    override val start: T,\n    override val endInclusive: T\n) : ClosedRange<T> {\n\n    override fun equals(other: Any?): Boolean {\n        return other is ComparableRange<*> && (isEmpty() && other.isEmpty() ||\n                start == other.start && endInclusive == other.endInclusive)\n    }\n\n    override fun hashCode(): Int {\n        return if (isEmpty()) -1 else 31 * start.hashCode() + endInclusive.hashCode()\n    }\n\n    override fun toString(): String = \"$start..$endInclusive\"\n}\n\n/**\n * Creates a range from this [Comparable] value to the specified [that] value.\n *\n * This value needs to be smaller than or equal to [that] value, otherwise the returned range will be empty.\n * @sample samples.ranges.Ranges.rangeFromComparable\n */\npublic operator fun <T : Comparable<T>> T.rangeTo(that: T): ClosedRange<T> = ComparableRange(this, that)\n\n/**\n * Represents a range of [Comparable] values.\n */\n@OptIn(ExperimentalStdlibApi::class)\nprivate open class ComparableOpenEndRange<T : Comparable<T>>(\n    override val start: T,\n    override val endExclusive: T\n) : OpenEndRange<T> {\n\n    override fun equals(other: Any?): Boolean {\n        return other is ComparableOpenEndRange<*> && (isEmpty() && other.isEmpty() ||\n                start == other.start && endExclusive == other.endExclusive)\n    }\n\n    override fun hashCode(): Int {\n        return if (isEmpty()) -1 else 31 * start.hashCode() + endExclusive.hashCode()\n    }\n\n    override fun toString(): String = \"$start..<$endExclusive\"\n}\n\n/**\n * Creates an open-ended range from this [Comparable] value to the specified [that] value.\n *\n * This value needs to be smaller than [that] value, otherwise the returned range will be empty.\n * @sample samples.ranges.Ranges.rangeFromComparable\n */\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun <T : Comparable<T>> T.rangeUntil(that: T): OpenEndRange<T> = ComparableOpenEndRange(this, that)\n\n\n/**\n * Represents a range of floating point numbers.\n * Extends
 [ClosedRange] interface providing custom operation [lessThanOrEquals] for comparing values of range domain type.\n *\n * This interface is implemented by floating point ranges returned by [Float.rangeTo] and [Double.rangeTo] operators to\n * achieve IEEE-754 comparison order instead of total order of floating point numbers.\n */\n@SinceKotlin(\"1.1\")\npublic interface ClosedFloatingPointRange<T : Comparable<T>> : ClosedRange<T> {\n    override fun contains(value: T): Boolean = lessThanOrEquals(start, value) && lessThanOrEquals(value, endInclusive)\n    override fun isEmpty(): Boolean = !lessThanOrEquals(start, endInclusive)\n\n    /**\n     * Compares two values of range domain type and returns true if first is less than or equal to second.\n     */\n    fun lessThanOrEquals(a: T, b: T): Boolean\n}\n\n\n/**\n * A closed range of values of type `Double`.\n *\n * Numbers are compared with the ends of this range according to IEEE-754.\n */\nprivate class ClosedDoubleRange(\n    start:
 Double,\n    endInclusive: Double\n) : ClosedFloatingPointRange<Double> {\n    private val _start = start\n    private val _endInclusive = endInclusive\n    override val start: Double get() = _start\n    override val endInclusive: Double get() = _endInclusive\n\n    override fun lessThanOrEquals(a: Double, b: Double): Boolean = a <= b\n\n    override fun contains(value: Double): Boolean = value >= _start && value <= _endInclusive\n    override fun isEmpty(): Boolean = !(_start <= _endInclusive)\n\n    override fun equals(other: Any?): Boolean {\n        return other is ClosedDoubleRange && (isEmpty() && other.isEmpty() ||\n                _start == other._start && _endInclusive == other._endInclusive)\n    }\n\n    override fun hashCode(): Int {\n        return if (isEmpty()) -1 else 31 * _start.hashCode() + _endInclusive.hashCode()\n    }\n\n    override fun toString(): String = \"$_start..$_endInclusive\"\n}\n\n/**\n * Creates a range from this [Double] value to the specified [that]
 value.\n *\n * Numbers are compared with the ends of this range according to IEEE-754.\n * @sample samples.ranges.Ranges.rangeFromDouble\n */\n@SinceKotlin(\"1.1\")\npublic operator fun Double.rangeTo(that: Double): ClosedFloatingPointRange<Double> = ClosedDoubleRange(this, that)\n\n/**\n * An open-ended range of values of type `Double`.\n *\n * Numbers are compared with the ends of this range according to IEEE-754.\n */\n@OptIn(ExperimentalStdlibApi::class)\nprivate class OpenEndDoubleRange(\n    start: Double,\n    endExclusive: Double\n) : OpenEndRange<Double> {\n    private val _start = start\n    private val _endExclusive =
```

```
endExclusive\n    override val start: Double get() = _start\n    override val endExclusive: Double get() = _endExclusive\n\n    private fun lessThanOrEquals(a: Double, b: Double): Boolean = a <= b\n\n    override fun contains(value: Double): Boolean = value >= _start && value < _endExclusive\n    override fun isEmpty(): Boolean = !(_start < _endExclusive)\n\n
    override fun equals(other: Any?): Boolean {\n        return other is OpenEndDoubleRange && (isEmpty() && other.isEmpty() ||\n                _start == other._start && _endExclusive == other._endExclusive)\n    }\n\n    override fun hashCode(): Int {\n        return if (isEmpty()) -1 else 31 * _start.hashCode() + _endExclusive.hashCode()\n    }\n\n    override fun toString(): String = \"$_start..<$_endExclusive\"\n}\n\n/**\n * Creates an open-ended range from this [Double] value to the specified [that] value.\n *\n * Numbers are compared with the ends of this range according to IEEE-754.\n */\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun Double.rangeUntil(that: Double): OpenEndRange<Double> = OpenEndDoubleRange(this, that)\n\n\n/**\n * A closed range of values of type `Float`.\n *\n * Numbers are compared with the ends of this range according to IEEE-754.\n */\nprivate class ClosedFloatRange(\n    start: Float,\n    endInclusive: Float\n) : ClosedFloatingPointRange<Float>
 {\n    private val _start = start\n    private val _endInclusive = endInclusive\n    override val start: Float get() = _start\n    override val endInclusive: Float get() = _endInclusive\n\n    override fun lessThanOrEquals(a: Float, b: Float): Boolean = a <= b\n\n    override fun contains(value: Float): Boolean = value >= _start && value <= _endInclusive\n    override fun isEmpty(): Boolean = !(_start <= _endInclusive)\n\n    override fun equals(other: Any?): Boolean {\n        return other is ClosedFloatRange && (isEmpty() && other.isEmpty() ||\n                _start == other._start && _endInclusive == other._endInclusive)\n    }\n\n    override fun hashCode(): Int {\n        return if (isEmpty()) -1 else 31 * _start.hashCode() + _endInclusive.hashCode()\n    }\n\n    override fun toString(): String = \"$_start..$_endInclusive\"\n}\n\n/**\n * Creates a range from this [Float] value to the specified [that] value.\n *\n * Numbers are compared with the ends of this range according
 to IEEE-754.\n * @sample samples.ranges.Ranges.rangeFromFloat\n */\n@SinceKotlin(\"1.1\")\npublic operator fun Float.rangeTo(that: Float): ClosedFloatingPointRange<Float> = ClosedFloatRange(this, that)\n\n\n/**\n * An open-ended range of values of type `Float`.\n *\n * Numbers are compared with the ends of this range according to IEEE-754.\n */\n@OptIn(ExperimentalStdlibApi::class)\nprivate class OpenEndFloatRange(\n    start: Float,\n    endExclusive: Float\n) : OpenEndRange<Float> {\n    private val _start = start\n    private val _endExclusive = endExclusive\n    override val start: Float get() = _start\n    override val endExclusive: Float get() = _endExclusive\n\n    private fun lessThanOrEquals(a: Float, b: Float): Boolean = a <= b\n\n    override fun contains(value: Float): Boolean = value >= _start && value < _endExclusive\n    override fun isEmpty(): Boolean = !(_start < _endExclusive)\n\n    override fun equals(other: Any?): Boolean {\n        return other is OpenEndFloatRange
 && (isEmpty() && other.isEmpty() ||\n                _start == other._start && _endExclusive == other._endExclusive)\n    }\n\n    override fun hashCode(): Int {\n        return if (isEmpty()) -1 else 31 * _start.hashCode() + _endExclusive.hashCode()\n    }\n\n    override fun toString(): String = \"$_start..<$_endExclusive\"\n}\n\n/**\n * Creates an open-ended range from this [Float] value to the specified [that] value.\n *\n * Numbers are compared with the ends of this range according to IEEE-754.\n */\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\npublic operator fun Float.rangeUntil(that: Float): OpenEndRange<Float> = OpenEndFloatRange(this, that)\n\n\n/**\n * Returns `true` if this iterable range contains the specified [element].\n *\n * Always returns `false` if the [element] is `null`.\n */\n@SinceKotlin(\"1.3\")\n@kotlin.internal.InlineOnly\npublic inline operator fun <T, R> R.contains(element: T?): Boolean where T : Any, R : ClosedRange<T>, R : Iterable<T> =\n    element
 != null && contains(element)\n\n/**\n * Returns `true` if this iterable range contains the specified [element].\n *\n * Always returns `false` if the [element] is `null`.\n */\n@SinceKotlin(\"1.7\")\n@ExperimentalStdlibApi\n@kotlin.internal.InlineOnly\npublic inline operator fun <T, R> R.contains(element: T?): Boolean where T : Any, R : OpenEndRange<T>, R : Iterable<T> =\n    element != null && contains(element)\n\ninternal fun checkStepIsPositive(isPositive: Boolean, step: Number) {\n    if (!isPositive)
```

throw IllegalArgumentException(\"Step must be positive, was: $step.\")\n}\n","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmName(\"KClasses\")\n@file:Suppress(\"UNCHECKED_CAST\")\n\npackage kotlin.reflect\n\nimport kotlin.internal.LowPriorityInOverloadResolution\n\n/**\n * Casts the given [value] to the class represented by this [KClass] object.\n * Throws an exception if the value is `null` or if it is not an instance of this class.\n *\n * This is an experimental function that behaves as a similar function from kotlin.reflect.full on JVM.\n *\n * @see [KClass.isInstance]\n * @see [KClass.safeCast]\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@LowPriorityInOverloadResolution\nfun <T : Any> KClass<T>.cast(value: Any?): T {\n if (!isInstance(value)) throw ClassCastException(\"Value cannot be cast to $qualifiedOrSimpleName\")\n return value as T\n}\n\n// TODO: replace with qualifiedName when it is fully supported in K/JS\ninternal expect val KClass<*>.qualifiedOrSimpleName: String?\n\n/**\n * Casts the given [value] to the class represented by this [KClass] object.\n * Returns `null` if the value is `null` or if it is not an instance of this class.\n *\n * This is an experimental function that behaves as a similar function from kotlin.reflect.full on JVM.\n *\n * @see [KClass.isInstance]\n * @see [KClass.cast]\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@LowPriorityInOverloadResolution\nfun <T : Any> KClass<T>.safeCast(value: Any?): T? {\n return if (isInstance(value)) value as T else null\n}\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.reflect\n\nimport kotlin.jvm.JvmField\nimport kotlin.jvm.JvmStatic\n\n\n\n/**\n * Represents a type projection. Type projection is usually the argument to another type in a type usage.\n * For example, in the type `Array<out Number>`, `out Number` is the covariant projection of the type represented by the class `Number`.\n *\n * Type projection is either the star projection, or an entity consisting of a specific type plus optional variance.\n *\n * See the [Kotlin language documentation](https://kotlinlang.org/docs/reference/generics.html#type-projections)\n * for more information.\n */\n@SinceKotlin(\"1.1\")\npublic data class KTypeProjection constructor(\n /**\n * The use-site variance specified in the projection, or `null` if this is a star projection.\n */\n public val variance: KVariance?,\n /**\n * The type specified in the projection, or `null` if this is a star projection.\n */\n public val type: KType?\n) {\n\n init {\n require((variance == null) == (type == null)) {\n if (variance == null)\n \"Star projection must have no type specified.\"\n else\n \"The projection variance $variance requires type to be specified.\"\n }\n }\n\n override fun toString(): String = when (variance) {\n null -> \"*\"\n KVariance.INVARIANT -> type.toString()\n KVariance.IN -> \"in $type\"\n KVariance.OUT -> \"out $type\"\n }\n\n\n public companion object {\n // provided for compiler access\n @JvmField\n @PublishedApi\n internal val star: KTypeProjection = KTypeProjection(null, null)\n\n /**\n * Star projection, denoted by the `*` character.\n * For example, in the type `KClass<*>`, `*` is the star projection.\n * See the [Kotlin language documentation](https://kotlinlang.org/docs/reference/generics.html#star-projections)\n * for more information.\n */\n public val STAR: KTypeProjection get() = star\n\n /**\n * Creates an invariant projection of a given type. Invariant projection is just the type itself,\n * without any use-site variance modifiers applied to it.\n * For example, in the type `Set<String>`, `String` is an invariant projection of the type represented by the class `String`.\n */\n @JvmStatic\n public fun invariant(type: KType): KTypeProjection =\n KTypeProjection(KVariance.INVARIANT, type)\n\n /**\n * Creates a contravariant projection of a given type, denoted by the `in` modifier applied to a type.\n * For example, in the type `MutableList<in Number>`, `in Number` is a contravariant projection of the type of class `Number`.\n */\n @JvmStatic\n public fun contravariant(type: KType): KTypeProjection =\n KTypeProjection(KVariance.IN, type)\n\n /**\n * Creates a covariant projection of a given type, denoted by the `out` modifier applied to a type.\n * For example, in the type

`Array<out Number>`, `out Number` is a covariant projection of the type of class `Number`.\n */\n @JvmStatic\n public fun covariant(type: KType): KTypeProjection =\n KTypeProjection(KVariance.OUT, type)\n }\n}","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.reflect\n\n/**\n * Represents variance applied to a type parameter on the declaration site (*declaration-site variance*),\n * or to a type in a projection (*use-site variance*).\n *\n * See the [Kotlin language documentation](https://kotlinlang.org/docs/reference/generics.html#variance)\n * for more information.\n *\n * @see [KTypeParameter.variance]\n * @see [KTypeProjection]\n */\n@SinceKotlin(\"1.1\")\nenum class KVariance {\n /**\n * The affected type parameter or type is *invariant*, which means it has no variance applied to it.\n */\n INVARIANT,\n\n /**\n * The affected type parameter or type is *contravariant*. Denoted by the `in` modifier in the source code.\n */\n IN,\n\n /**\n * The affected type parameter or type is *covariant*. Denoted by the `out` modifier in the source code.\n */\n OUT,\n}","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.reflect\n\n/**\n * Returns a runtime representation of the given reified type [T] as an instance of [KType].\n *\n * Note that on JVM, the created type has no annotations ([KType.annotations] returns an empty list)\n * even if the type in the source code is annotated. Support for type annotations might be added in a future version.\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic inline fun <reified T> typeOf(): KType =\n throw UnsupportedOperationException(\"This function is implemented as an intrinsic on all supported platforms.\")\n","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"StringsKt\")\n\npackage kotlin.text\n\n/**\n * An object to which char sequences and values can be appended.\n */\nexpect interface Appendable {\n /**\n * Appends the specified character [value] to this Appendable and returns this instance.\n *\n * @param value the character to append.\n */\n fun append(value: Char): Appendable\n\n /**\n * Appends the specified character sequence [value] to this Appendable and returns this instance.\n *\n * @param value the character sequence to append. If [value] is `null`, then the four characters `\"null\"` are appended to this Appendable.\n */\n fun append(value: CharSequence?): Appendable\n\n /**\n * Appends a subsequence of the specified character sequence [value] to this Appendable and returns this instance.\n *\n * @param value the character sequence from which a subsequence is appended. If [value] is `null`,\n * then characters are appended as if [value] contained the four characters `\"null\"`.\n * @param startIndex the beginning (inclusive) of the subsequence to append.\n * @param endIndex the end (exclusive) of the subsequence to append.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.\n */\n fun append(value: CharSequence?, startIndex: Int, endIndex: Int): Appendable\n}\n\n/**\n * Appends a subsequence of the specified character sequence [value] to this Appendable and returns this instance.\n *\n * @param value the character sequence from which a subsequence is appended.\n * @param startIndex the beginning (inclusive) of the subsequence to append.\n * @param endIndex the end (exclusive) of the subsequence to append.\n *\n * @throws IndexOutOfBoundsException or [IllegalArgumentException] when [startIndex] or [endIndex] is out of range of the [value] character sequence indices or when `startIndex > endIndex`.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun <T : Appendable> T.appendRange(value: CharSequence, startIndex: Int, endIndex: Int): T {\n @Suppress(\"UNCHECKED_CAST\")\n return append(value, startIndex, endIndex) as T\n}\n\n/**\n * Appends

```
all arguments to the given [Appendable].\n */\npublic fun <T : Appendable> T.append(vararg value: CharSequence?): T {\n    for (item in value)\n        append(item)\n    return this\n}\n\n/** Appends a line feed character (`\\n`) to this Appendable. */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun Appendable.appendLine(): Appendable = append('\\n')\n\n/** Appends value to the given Appendable and a line feed character (`\\n`) after it. */\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun Appendable.appendLine(value: CharSequence?): Appendable = append(value).appendLine()\n\n/** Appends value to the given
 Appendable and a line feed character (`\\n`) after it.
*/\n@SinceKotlin(\"1.4\")\n@kotlin.internal.InlineOnly\npublic inline fun Appendable.appendLine(value: Char): Appendable = append(value).appendLine()\n\n\ninternal fun <T> Appendable.appendElement(element: T, transform: ((T) -> CharSequence)?) {\n    when {\n        transform != null -> append(transform(element))\n        element is CharSequence? -> append(element)\n        element is Char -> append(element)\n        else -> append(element.toString())\n    }\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"StringsKt\")\n\npackage kotlin.text\n\n/**\n * Trims leading whitespace characters followed by [marginPrefix] from every line of a source string and removes\n * the first and the last lines if they
 are blank (notice difference blank vs empty).\n *\n * Doesn't affect a line if it doesn't contain [marginPrefix] except the first and the last blank lines.\n *\n * Doesn't preserve the original line endings.\n *\n * @param marginPrefix non-blank string, which is used as a margin delimiter. Default is `|` (pipe character).\n *\n * @sample samples.text.Strings.trimMargin\n * @see trimIndent\n * @see kotlin.text.isWhitespace\n
*/\n@kotlin.internal.IntrinsicConstEvaluation\npublic fun String.trimMargin(marginPrefix: String = \"|\"): String =\n    replaceIndentByMargin(\"\", marginPrefix)\n\n/**\n * Detects indent by [marginPrefix] as it does [trimMargin] and replace it with [newIndent].\n *\n * @param marginPrefix non-blank string, which is used as a margin delimiter. Default is `|` (pipe character).\n */\npublic fun String.replaceIndentByMargin(newIndent: String = \"\", marginPrefix: String = \"|\"): String {\n    require(marginPrefix.isNotBlank()) { \"marginPrefix must be non-blank string.\"
 }\n    val lines = lines()\n\n    return lines.reindent(length + newIndent.length * lines.size, getIndentFunction(newIndent), { line ->\n        val firstNonWhitespaceIndex = line.indexOfFirst { !it.isWhitespace() }\n\n        when {\n            firstNonWhitespaceIndex == -1 -> null\n            line.startsWith(marginPrefix, firstNonWhitespaceIndex) -> line.substring(firstNonWhitespaceIndex + marginPrefix.length)\n            else -> null\n        }\n    })\n}\n\n/**\n * Detects a common minimal indent of all the input lines, removes it from every line and also removes the first and the last\n * lines if they are blank (notice difference blank vs empty).\n *\n * Note that blank lines do not affect the detected indent level.\n *\n * In case if there are non-blank lines with no leading whitespace characters (no indent at all) then the\n * common indent is 0, and therefore this function doesn't change the indentation.\n *\n * Doesn't preserve the original line endings.\n *\n * @sample
 samples.text.Strings.trimIndent\n * @see trimMargin\n * @see kotlin.text.isBlank\n
*/\n@kotlin.internal.IntrinsicConstEvaluation\npublic fun String.trimIndent(): String = replaceIndent(\"\")\n\n/**\n * Detects a common minimal indent like it does [trimIndent] and replaces it with the specified [newIndent].\n
*/\npublic fun String.replaceIndent(newIndent: String = \"\"): String {\n    val lines = lines()\n\n    val minCommonIndent = lines\n        .filter(String::isNotBlank)\n        .map(String::indentWidth)\n        .minOrNull() ?: 0\n\n    return lines.reindent(length + newIndent.length * lines.size, getIndentFunction(newIndent), { line -> line.drop(minCommonIndent) })\n}\n\n/**\n * Prepends [indent] to every line of the original string.\n *\n * Doesn't preserve the original line endings.\n */\npublic fun String.prependIndent(indent: String = \"    \"): String =\n    lineSequence()\n        .map {\n            when {\n                it.isBlank() -> {\n                    when {\n                        it.length < indent.length -> indent\n                        else -> it\n                    }\n                }\n                else -> indent + it\n            }\n        }\n        .joinToString(\"\\n\")\n\nprivate fun String.indentWidth(): Int =
```

indexOfFirst { !it.isWhitespace() }.let { if (it == -1) length else it }\n\nprivate fun getIndentFunction(indent: String) = when {\n   indent.isEmpty() -> { line: String -> line }\n   else -> { line: String -> indent + line }\n}\n\nprivate inline fun List<String>.reindent(\n   resultSizeEstimate: Int,\n   indentAddFunction: (String) -> String,\n indentCutFunction: (String) -> String?\n): String {\n   val lastIndex = lastIndex\n   return mapIndexedNotNull { index, value ->\n   if ((index == 0 || index == lastIndex) && value.isBlank())\n   null\n   else\n indentCutFunction(value)?.let(indentAddFunction) ?: value\n   }\n   .joinTo(StringBuilder(resultSizeEstimate), \"\\n\")\n

        .toString()\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/**\n * Defines names for Unicode symbols used in proper Typography.\n */\npublic object Typography {\n   /** The character " \u2013 quotation mark */\n   public const val quote: Char = '\\u0022'\n   /** The character $ \u2013 dollar sign */\n   public const val dollar: Char = '\\u0024'\n   /** The character & \u2013 ampersand */\n   public const val amp: Char = '\\u0026'\n   /** The character < \u2013 less-than sign */\n   public const val less: Char = '\\u003C'\n   /** The character > \u2013 greater-than sign */\n   public const val greater: Char = '\\u003E'\n   /** The non-breaking space character */\n   public const val nbsp: Char = '\\u00A0'\n   /** The character × */\n   public const val times: Char = '\\u00D7'\n   /** The character ¢ */\n   public const val cent: Char = '\\u00A2'\n   /** The character £ */\n   public const val pound: Char = '\\u00A3'\n   /** The character § */\n   public const val section: Char = '\\u00A7'\n   /** The character © */\n   public const val copyright: Char = '\\u00A9'\n   /** The character « */\n   @SinceKotlin(\"1.6\")\n   public const val leftGuillemet: Char = '\\u00AB'\n   /** The character » */\n   @SinceKotlin(\"1.6\")\n   public const val rightGuillemet: Char = '\\u00BB'\n   /** The character ® */\n   public const val registered: Char = '\\u00AE'\n   /** The character ° */\n   public const val degree: Char = '\\u00B0'\n   /** The character ± */\n   public const val plusMinus: Char = '\\u00B1'\n   /** The character ¶ */\n   public const val paragraph: Char = '\\u00B6'\n   /** The character · */\n   public const val middleDot: Char = '\\u00B7'\n

    /** The character ½ */\n   public const val half: Char = '\\u00BD'\n   /** The character – */\n   public const val ndash: Char = '\\u2013'\n   /** The character — */\n   public const val mdash: Char = '\\u2014'\n   /** The character ‘ */\n   public const val leftSingleQuote: Char = '\\u2018'\n   /** The character ’ */\n   public const val rightSingleQuote: Char = '\\u2019'\n   /** The character ‚ */\n   public const val lowSingleQuote: Char = '\\u201A'\n   /** The character “ */\n   public const val leftDoubleQuote: Char = '\\u201C'\n   /** The character ” */\n   public const val rightDoubleQuote: Char = '\\u201D'\n   /** The character „ */\n   public const val lowDoubleQuote: Char = '\\u201E'\n   /** The character † */\n   public const val dagger: Char = '\\u2020'\n   /** The character ‡ */\n   public const val doubleDagger: Char = '\\u2021'\n   /** The character • */\n   public const val bullet: Char = '\\u2022'\n   /** The character … */\n   public const val ellipsis: Char = '\\u2026'\n   /** The character ′ */\n   public const val prime: Char = '\\u2032'\n   /** The character ″ */\n   public const val doublePrime: Char = '\\u2033'\n   /** The character € */\n   public const val euro: Char = '\\u20AC'\n   /** The character ™ */\n   public const val tm: Char = '\\u2122'\n   /** The character ≈ */\n   public const val almostEqual: Char = '\\u2248'\n   /** The character ≠ */\n   public const val notEqual: Char = '\\u2260'\n   /** The character ≤ */\n   public const val lessOrEqual: Char = '\\u2264'\n   /** The character ≥ */\n   public const val greaterOrEqual: Char = '\\u2265'\n\n   /** The character « */\n   @Deprecated(\"This constant has a typo in the name. Use leftGuillemet instead.\", ReplaceWith(\"Typography.leftGuillemet\"))\n   @DeprecatedSinceKotlin(\"1.6\")\n

    public const val leftGuillemete: Char = '\\u00AB'\n\n   /** The character » */\n   @Deprecated(\"This constant has a typo in the name. Use rightGuillemet instead.\", ReplaceWith(\"Typography.rightGuillemet\"))\n@DeprecatedSinceKotlin(\"1.6\")\n   public const val rightGuillemete: Char = '\\u00BB'\n}","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.text\n\n/**\n *

Represents a collection of captured groups in a single match of a regular expression.\n *\n * This collection has size of `groupCount + 1` where `groupCount` is the count of groups in the regular expression.\n * Groups are indexed from 1 to `groupCount` and group with the index 0 corresponds to the entire match.\n *\n * An element of the collection at the particular index can be `null`,\n * if the corresponding group in the regular expression is optional and\n * there was no match captured by that group.\n */\npublic interface MatchGroupCollection : Collection<MatchGroup?> {\n\n /** Returns a group with the specified [index].\n *\n * @return An instance of [MatchGroup] if the group with the specified [index] was matched or `null` otherwise.\n *\n * Groups are indexed from 1 to the count of groups in the regular expression. A group with the index 0\n * corresponds to the entire match.\n */\n public operator fun get(index: Int): MatchGroup?\n}\n\n/**\n * Extends [MatchGroupCollection] by introducing a way to get matched groups by name, when regex supports it.\n */\n@SinceKotlin(\"1.1\")\npublic interface MatchNamedGroupCollection : MatchGroupCollection {\n /**\n * Returns a named group with the specified [name].\n * @return An instance of [MatchGroup] if the group with the specified [name] was matched or `null` otherwise.\n * @throws IllegalArgumentException if there is no group with the specified [name] defined in the regex pattern.\n * @throws UnsupportedOperationException if this match group collection doesn't support getting match groups by name,\n * for example, when it's not supported by the current platform.\n */\n public operator fun get(name: String): MatchGroup?\n}\n\n/**\n * Represents the results from a single regular expression match.\n */\npublic interface MatchResult {\n /** The range of indices in the original string where match was captured. */\n public val range: IntRange\n /** The substring from the input string captured by this match. */\n public val value: String\n /**\n * A collection of groups matched by the regular expression.\n *\n * This collection has size of `groupCount + 1` where `groupCount` is the count of groups in the regular expression.\n * Groups are indexed from 1 to `groupCount` and group with the index 0 corresponds to the entire match.\n */\n public val groups: MatchGroupCollection\n /**\n * A list of matched indexed group values.\n *\n * This list has size of `groupCount + 1` where `groupCount` is the count of groups in the regular expression.\n * Groups are indexed from 1 to `groupCount` and group with the index 0 corresponds to the entire match.\n *\n * If the group in the regular expression is optional and there were no match captured by that group,\n * corresponding item in [groupValues] is an empty string.\n *\n * @sample samples.text.Regexps.matchDestructuringToGroupValues\n */\n public val groupValues: List<String>\n\n /**\n * An instance of [MatchResult.Destructured] wrapper providing components for destructuring assignment of group values.\n *\n * component1 corresponds to the value of the first group, component2 \u2014 of the second, and so on.\n *\n * @sample samples.text.Regexps.matchDestructuringToGroupValues\n */\n public val destructured: Destructured\n get() = Destructured(this)\n\n /** Returns a new [MatchResult] with the results for the next match, starting at the position\n * at which the last match ended (at the character after the last matched character).\n */\n public fun next(): MatchResult?\n\n /**\n * Provides components for destructuring assignment of group values.\n *\n * [component1] corresponds to the value of the first group, [component2] \u2014 of the second, and so on.\n *\n * If the group in the regular expression is optional and there were no match captured by that group,\n * corresponding component value is an empty string.\n *\n * @sample samples.text.Regexps.matchDestructuringToGroupValues\n */\n public class Destructured internal constructor(public val match: MatchResult) {\n @kotlin.internal.InlineOnly\n public operator inline fun component1(): String = match.groupValues[1]\n @kotlin.internal.InlineOnly\n public operator inline fun component2(): String = match.groupValues[2]\n @kotlin.internal.InlineOnly\n public operator inline fun component3(): String = match.groupValues[3]\n @kotlin.internal.InlineOnly\n public operator inline fun component4(): String = match.groupValues[4]\n @kotlin.internal.InlineOnly\n public operator inline fun component5(): String = match.groupValues[5]\n @kotlin.internal.InlineOnly\n public operator inline fun component6(): String = match.groupValues[6]\n @kotlin.internal.InlineOnly\n public operator inline fun component7(): String = match.groupValues[7]\n @kotlin.internal.InlineOnly\n public operator inline fun component8(): String = match.groupValues[8]\n @kotlin.internal.InlineOnly\n public operator inline

fun component9(): String = match.groupValues[9]\n @kotlin.internal.InlineOnly\n public operator inline fun component10(): String = match.groupValues[10]\n\n /**\n * Returns destructured group values as a list of strings.\n * First value in the returned list corresponds to the value of the first group, and so on.\n *\n * @sample samples.text.Regexps.matchDestructuringToGroupValues\n */\n public fun toList(): List<String> = match.groupValues.subList(1, match.groupValues.size)\n }\n}","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass()\n@file:kotlin.jvm.JvmName(\"DurationUnitKt\")\n\npackage kotlin.time\n\n\n/**\n * The list of possible time measurement units, in which a duration can be expressed.\n *\n * The smallest time unit is [NANOSECONDS] and the largest is [DAYS], which corresponds to exactly 24 [HOURS].\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalTime::class)\npublic expect enum class DurationUnit {\n /**\n * Time unit representing one nanosecond, which is 1/1000 of a microsecond.\n */\n NANOSECONDS,\n /**\n * Time unit representing one microsecond, which is 1/1000 of a millisecond.\n */\n MICROSECONDS,\n /**\n * Time unit representing one millisecond, which is 1/1000 of a second.\n */\n MILLISECONDS,\n /**\n * Time unit representing one second.\n */\n SECONDS,\n /**\n * Time unit representing one minute.\n */\n MINUTES,\n /**\n * Time unit representing one hour.\n */\n HOURS,\n /**\n * Time unit representing one day, which is always equal to 24 hours.\n */\n DAYS;\n}\n\n/** Converts the given time duration [value] expressed in the specified [sourceUnit] into the specified [targetUnit]. */\n@SinceKotlin(\"1.3\")\ninternal expect fun convertDurationUnit(value: Double, sourceUnit: DurationUnit, targetUnit: DurationUnit): Double\n\n// overflown result is unspecified\n@SinceKotlin(\"1.5\")\ninternal expect fun convertDurationUnitOverflow(value: Long, sourceUnit: DurationUnit, targetUnit: DurationUnit): Long\n\n// overflown result is coerced in the Long range boundaries\n@SinceKotlin(\"1.5\")\ninternal expect fun convertDurationUnit(value: Long, sourceUnit: DurationUnit, targetUnit: DurationUnit): Long\n\n\n@SinceKotlin(\"1.3\")\n@Suppress(\"REDUNDANT_ELSE_IN_WHEN\")\ninternal fun DurationUnit.shortName(): String = when (this) {\n DurationUnit.NANOSECONDS -> \"ns\"\n DurationUnit.MICROSECONDS -> \"us\"\n DurationUnit.MILLISECONDS -> \"ms\"\n DurationUnit.SECONDS -> \"s\"\n DurationUnit.MINUTES -> \"m\"\n DurationUnit.HOURS -> \"h\"\n DurationUnit.DAYS -> \"d\"\n else -> error(\"Unknown unit: $this\")\n}\n\n@SinceKotlin(\"1.5\")\ninternal fun durationUnitByShortName(shortName: String): DurationUnit = when (shortName) {\n \"ns\" -> DurationUnit.NANOSECONDS\n \"us\" -> DurationUnit.MICROSECONDS\n \"ms\" -> DurationUnit.MILLISECONDS\n \"s\" -> DurationUnit.SECONDS\n \"m\" -> DurationUnit.MINUTES\n \"h\" -> DurationUnit.HOURS\n \"d\" -> DurationUnit.DAYS\n else -> throw IllegalArgumentException(\"Unknown duration unit short name: $shortName\")\n}\n\n@SinceKotlin(\"1.5\")\ninternal fun durationUnitByIsoChar(isoChar: Char, isTimeComponent: Boolean): DurationUnit =\n when {\n !isTimeComponent -> {\n when (isoChar) {\n 'D' -> DurationUnit.DAYS\n else -> throw IllegalArgumentException(\"Invalid or unsupported duration ISO non-time unit: $isoChar\")\n }\n }\n else -> {\n when (isoChar) {\n 'H' -> DurationUnit.HOURS\n 'M' -> DurationUnit.MINUTES\n 'S' -> DurationUnit.SECONDS\n else -> throw IllegalArgumentException(\"Invalid duration ISO time unit: $isoChar\")\n }\n }\n}","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport kotlin.annotation.AnnotationTarget.*\n\n/**\n * This annotation marks the experimental preview of the standard library API for measuring time and working with durations.\n *\n * > Note that this API is in a preview state and has a very high chance of being changed in the future.\n * Do not use it if you develop a library since your library will

become binary incompatible\n * with the future versions of the standard library.\n *\n * Any usage of a declaration annotated with `@ExperimentalTime` must be accepted either by\n * annotating that usage with the [OptIn] annotation, e.g. `@OptIn(ExperimentalTime::class)`,\n * or by using the compiler argument `-opt-in=kotlin.time.ExperimentalTime`.\n */\n@RequiresOptIn(level = RequiresOptIn.Level.ERROR)\n@MustBeDocumented\n@Retention(AnnotationRetention.BINARY)\n@Target(\n CLASS,\n ANNOTATION_CLASS,\n PROPERTY,\n FIELD,\n LOCAL_VARIABLE,\n VALUE_PARAMETER,\n CONSTRUCTOR,\n FUNCTION,\n PROPERTY_GETTER,\n PROPERTY_SETTER,\n TYPEALIAS\n)\n@SinceKotlin(\"1.3\")\npublic annotation class ExperimentalTime\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport kotlin.jvm.JvmInline\n\n/**\n * A source of time for measuring time intervals.\n *\n * The only operation provided by the time source is [markNow]. It returns a [TimeMark], which can be used to query the elapsed time later.\n *\n * @see [measureTime]\n * @see [measureTimedValue]\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\npublic interface TimeSource {\n /**\n * Marks a point in time on this time source.\n *\n * The returned [TimeMark] instance encapsulates the captured time point and allows querying\n * the duration of time interval [elapsed][TimeMark.elapsedNow] from that point.\n */\n public fun markNow(): TimeMark\n\n /**\n * A [TimeSource] that returns [time marks][ComparableTimeMark] that can be compared for difference with each other.\n */\n @SinceKotlin(\"1.8\")\n @ExperimentalTime\n public interface WithComparableMarks : TimeSource {\n override fun markNow(): ComparableTimeMark\n }\n\n /**\n * The most precise time source available in the platform.\n *\n * This time source returns its readings from a source of monotonic time when it is available in a target platform,\n * and resorts to a non-monotonic time source otherwise.\n *\n * The function [markNow] of this time source returns the specialized [ValueTimeMark] that is an inline value class\n * wrapping a platform-dependent time reading value.\n */\n public object Monotonic : TimeSource.WithComparableMarks {\n override fun markNow(): ValueTimeMark = MonotonicTimeSource.markNow()\n override fun toString(): String = MonotonicTimeSource.toString()\n\n /**\n * A specialized [kotlin.time.TimeMark] returned by [TimeSource.Monotonic] time source.\n *\n * This time mark is implemented as an inline value class wrapping a platform-dependent\n * time reading value of the default monotonic time source, thus allowing to avoid additional boxing\n * of that value.\n *\n * The operations [plus] and [minus] are also specialized to return [ValueTimeMark] type.\n *\n * This time mark implements [ComparableTimeMark] and therefore is comparable with other time marks\n * obtained from the same [TimeSource.Monotonic] time source.\n */\n @ExperimentalTime\n @SinceKotlin(\"1.7\")\n @JvmInline\n public value class ValueTimeMark internal constructor(internal val reading: ValueTimeMarkReading) : ComparableTimeMark {\n override fun elapsedNow(): Duration = MonotonicTimeSource.elapsedFrom(this)\n override fun plus(duration: Duration): ValueTimeMark = MonotonicTimeSource.adjustReading(this, duration)\n override fun minus(duration: Duration): ValueTimeMark = MonotonicTimeSource.adjustReading(this, -duration)\n override fun hasPassedNow(): Boolean = !elapsedNow().isNegative()\n override fun hasNotPassedNow(): Boolean = elapsedNow().isNegative()\n\n override fun minus(other: ComparableTimeMark): Duration {\n if (other !is ValueTimeMark)\n throw IllegalArgumentException(\"Subtracting or comparing time marks from different time sources is not possible: $this and $other\")\n return this.minus(other)\n }\n\n /**\n * Returns the duration elapsed between the [other] time mark obtained from the same [TimeSource.Monotonic] time source and `this` time mark.\n *\n * The returned duration can be infinite if the time marks are far away from each other and\n * the result doesn't fit into [Duration] type,\n * or if one time mark is infinitely distant, or if both `this` and [other] time marks\n * lie infinitely distant on the opposite sides of the time scale.\n *\n * Two infinitely distant time marks on the same side of the time scale are considered equal and\n * the duration between them is [Duration.ZERO].\n */\n

public operator fun minus(other: ValueTimeMark): Duration = MonotonicTimeSource.differenceBetween(this, other)\n\n /**\n * Compares this time mark with the [other] time mark for order.\n *\n * - Returns zero if this time mark represents *the same moment* of time as the [other] time mark.\n * - Returns a negative number if this time mark is *earlier* than the [other] time mark.\n * - Returns a positive number if this time mark is *later* than the [other] time mark.\n */\n public operator fun compareTo(other: ValueTimeMark): Int =\n (this - other).compareTo(Duration.ZERO)\n }\n }\n\n public companion object {\n\n }\n}\n\n/** A platform-specific reading type that is wrapped by [TimeSource.Monotonic.ValueTimeMark] inline class. */\ninternal expect class ValueTimeMarkReading\n\n\n/**\n * Represents a time point notched on a particular [TimeSource]. Remains bound to the time source it was taken from\n * and allows querying for the duration of time elapsed from that point (see the function [elapsedNow]).\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\npublic interface TimeMark {\n /**\n * Returns the amount of time passed from this mark measured with the time source from which this mark was taken.\n *\n * Note that the value returned by this function can change on subsequent invocations.\n *\n * @throws IllegalArgumentException an implementation may throw if calculating the elapsed time involves\n * adding a positive infinite duration to an infinitely distant past time mark or\n * a negative infinite duration to an infinitely distant future time mark.\n */\n public abstract fun elapsedNow(): Duration\n\n /**\n * Returns a time mark on the same time source that is ahead of this time mark by the specified [duration].\n *\n * The returned time mark is more _late_ when the [duration] is positive, and more _early_ when the [duration] is negative.\n *\n * If the time mark is adjusted too far in the past or in the future, it may saturate to an infinitely distant time mark.\n * In that case, [elapsedNow] will return an infinite duration elapsed from such infinitely distant mark.\n *\n * @throws IllegalArgumentException an implementation may throw if a positive infinite duration is added to an infinitely distant past time mark or\n * a negative infinite duration is added to an infinitely distant future time mark.\n */\n public operator fun plus(duration: Duration): TimeMark = AdjustedTimeMark(this, duration)\n\n /**\n * Returns a time mark on the same time source that is behind this time mark by the specified [duration].\n *\n * The returned time mark is more _early_ when the [duration] is positive, and more _late_ when the [duration] is negative.\n *\n * If the time mark is adjusted too far in the past or in the future, it may saturate to an infinitely distant time mark.\n * In that case, [elapsedNow] will return an infinite duration elapsed from such infinitely distant mark.\n *\n * @throws IllegalArgumentException an implementation may throw if a positive infinite duration is subtracted from an infinitely distant future time mark or\n * a negative infinite duration is subtracted from an infinitely distant past time mark.\n */\n public open operator fun minus(duration: Duration): TimeMark = plus(-duration)\n\n\n /**\n * Returns true if this time mark has passed according to the time source from which this mark was taken.\n *\n * Note that the value returned by this function can change on subsequent invocations.\n * If the time source is monotonic, it can change only from `false` to `true`, namely, when the time mark becomes behind the current point of the time source.\n */\n public fun hasPassedNow(): Boolean = !elapsedNow().isNegative()\n\n /**\n * Returns false if this time mark has not passed according to the time source from which this mark was taken.\n *\n * Note that the value returned by this function can change on subsequent invocations.\n * If the time source is monotonic, it can change only from `true` to `false`, namely, when the time mark becomes behind the current point of the time source.\n */\n public fun hasNotPassedNow(): Boolean = elapsedNow().isNegative()\n}\n\n/**\n * A [TimeMark] that can be compared for difference with other time marks obtained from the same [TimeSource.WithComparableMarks] time source.\n */\n@SinceKotlin(\"1.8\")\n@ExperimentalTime\npublic interface ComparableTimeMark : TimeMark, Comparable<ComparableTimeMark> {\n public abstract override operator fun plus(duration: Duration): ComparableTimeMark\n public open override operator fun minus(duration: Duration): ComparableTimeMark = plus(-duration)\n\n /**\n * Returns the duration elapsed between the [other] time mark and `this` time mark.\n *\n * The returned duration can be infinite if the time marks are far away from each other and\n * the result doesn't fit into [Duration] type,\n * or if one time mark is infinitely distant, or if both

```
`this` and [other] time marks\n     * lie infinitely distant on the opposite sides of the time scale.\n     *\n     * Two infinitely
 distant time marks on the same side of the time scale are considered equal and\n     * the duration between them is [Duration.ZERO].\n     *\n     * Note that the other time mark must be obtained from the same time source as this one.\n     *\n     * @throws IllegalArgumentException if time marks were obtained from different time sources.\n     */\n    public operator fun minus(other: ComparableTimeMark): Duration\n\n    /**\n     * Compares this time mark with the [other] time mark for order.\n     *\n     * - Returns zero if this time mark represents *the same moment* of time as the [other] time mark.\n     * - Returns a negative number if this time mark is *earlier* than the [other] time mark.\n     * - Returns a positive number if this time mark is *later* than the [other] time mark.\n     *\n     * Note that the other time mark must be obtained from the same time source as this one.\n     *\n     * @throws IllegalArgumentException if time marks were obtained from different
 time sources.\n     */\n    public override operator fun compareTo(other: ComparableTimeMark): Int =\n        (this - other).compareTo(Duration.ZERO)\n\n    /**\n     * Returns `true` if two time marks from the same time source represent the same moment of time, and `false` otherwise,\n     * including the situation when the time marks were obtained from different time sources.\n     */\n    override fun equals(other: Any?): Boolean\n    override fun hashCode(): Int\n}\n\n\n@ExperimentalTime\nprivate class AdjustedTimeMark(val mark: TimeMark, val adjustment: Duration) : TimeMark {\n    override fun elapsedNow(): Duration = mark.elapsedNow() - adjustment\n\n    override fun plus(duration: Duration): TimeMark = AdjustedTimeMark(mark, adjustment + duration)\n}\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport
 kotlin.time.Duration.Companion.milliseconds\nimport kotlin.time.Duration.Companion.nanoseconds\nimport kotlin.time.Duration.Companion.seconds\n\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\ninternal expect object MonotonicTimeSource : TimeSource.WithComparableMarks {\n    override fun markNow(): TimeSource.Monotonic.ValueTimeMark\n    fun elapsedFrom(timeMark: TimeSource.Monotonic.ValueTimeMark): Duration\n    fun differenceBetween(one: TimeSource.Monotonic.ValueTimeMark, another: TimeSource.Monotonic.ValueTimeMark): Duration\n    fun adjustReading(timeMark: TimeSource.Monotonic.ValueTimeMark, duration: Duration): TimeSource.Monotonic.ValueTimeMark\n}\n\n/**\n * An abstract class used to implement time sources that return their readings as [Long] values in the specified [unit].\n *\n * @property unit The unit in which this time source's readings are expressed.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\npublic abstract class AbstractLongTimeSource(protected val unit:
 DurationUnit) : TimeSource.WithComparableMarks {\n    /**\n     * This protected method should be overridden to return the current reading of the time source expressed as a [Long] number\n     * in the unit specified by the [unit] property.\n     */\n    protected abstract fun read(): Long\n\n    private class LongTimeMark(private val startedAt: Long, private val timeSource: AbstractLongTimeSource, private val offset: Duration) : ComparableTimeMark {\n
 override fun elapsedNow(): Duration = if (offset.isInfinite()) -offset else (timeSource.read() - startedAt).toDuration(timeSource.unit) - offset\n        override fun plus(duration: Duration): ComparableTimeMark = LongTimeMark(startedAt, timeSource, offset + duration)\n        override fun minus(other: ComparableTimeMark): Duration {\n            if (other !is LongTimeMark || this.timeSource != other.timeSource)\n                throw IllegalArgumentException(\"Subtracting or comparing time marks from different time sources is
 not possible: $this and $other\")\n\n//            val thisValue = this.effectiveDuration()\n//            val otherValue = other.effectiveDuration()\n//            if (thisValue == otherValue && thisValue.isInfinite()) return Duration.ZERO\n//            return thisValue - otherValue\n            if (this.offset == other.offset && this.offset.isInfinite()) return Duration.ZERO\n            val offsetDiff = this.offset - other.offset\n            val startedAtDiff = (this.startedAt - other.startedAt).toDuration(timeSource.unit)\n//            println(\"$startedAtDiff, $offsetDiff\")\n            return if (startedAtDiff == -offsetDiff) Duration.ZERO else startedAtDiff + offsetDiff\n        }\n\n        override fun equals(other: Any?): Boolean =\n            other is LongTimeMark && this.timeSource ==
```

```
other.timeSource && (this - other) == Duration.ZERO\n\n        internal fun effectiveDuration(): Duration {\n
if (offset.isInfinite()) return offset\n            val unit = timeSource.unit\n
            if (unit >= DurationUnit.MILLISECONDS) {\n                return startedAt.toDuration(unit) + offset\n
}\n            val scale = convertDurationUnit(1L, DurationUnit.MILLISECONDS, unit)\n            val startedAtMillis =
startedAt / scale\n            val startedAtRem = startedAt % scale\n\n            return offset.toComponents {
offsetSeconds, offsetNanoseconds ->\n                val offsetMillis = offsetNanoseconds / NANOS_IN_MILLIS\n
        val offsetRemNanos = offsetNanoseconds % NANOS_IN_MILLIS\n\n                // add component-wise\n
    (startedAtRem.toDuration(unit) + offsetRemNanos.nanoseconds) +\n                    (startedAtMillis +
offsetMillis).milliseconds +\n                    offsetSeconds.seconds\n            }\n\n        }\n\n        override fun
hashCode(): Int = effectiveDuration().hashCode()\n\n        override fun toString(): String =
\"LongTimeMark($startedAt${timeSource.unit.shortName()} + $offset
 (=${effectiveDuration()}), $timeSource)\"\n    }\n\n    override fun markNow(): ComparableTimeMark =
LongTimeMark(read(), this, Duration.ZERO)\n}\n\n/**\n * An abstract class used to implement time sources that
return their readings as [Double] values in the specified [unit].\n *\n * @property unit The unit in which this time
source's readings are expressed.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\n@Deprecated(\"Using
AbstractDoubleTimeSource is no longer recommended, use AbstractLongTimeSource instead.\")\npublic abstract
class AbstractDoubleTimeSource(protected val unit: DurationUnit) : TimeSource.WithComparableMarks {\n
/**\n     * This protected method should be overridden to return the current reading of the time source expressed as a
[Double] number\n     * in the unit specified by the [unit] property.\n     */\n    protected abstract fun read():
Double\n\n    @Suppress(\"DEPRECATION\")\n    private class DoubleTimeMark(private val startedAt: Double,
private val timeSource:
 AbstractDoubleTimeSource, private val offset: Duration) : ComparableTimeMark {\n        override fun
elapsedNow(): Duration = (timeSource.read() - startedAt).toDuration(timeSource.unit) - offset\n        override fun
plus(duration: Duration): ComparableTimeMark = DoubleTimeMark(startedAt, timeSource, offset + duration)\n\n
    override fun minus(other: ComparableTimeMark): Duration {\n            if (other !is DoubleTimeMark ||
this.timeSource != other.timeSource)\n                throw IllegalArgumentException(\"Subtracting or comparing time
marks from different time sources is not possible: $this and $other\")\n\n            if (this.offset == other.offset &&
this.offset.isInfinite()) return Duration.ZERO\n            val offsetDiff = this.offset - other.offset\n            val
startedAtDiff = (this.startedAt - other.startedAt).toDuration(timeSource.unit)\n            return if (startedAtDiff == -
offsetDiff) Duration.ZERO else startedAtDiff + offsetDiff\n        }\n\n
        override fun equals(other: Any?): Boolean {\n            return other is DoubleTimeMark && this.timeSource ==
other.timeSource && (this - other) == Duration.ZERO\n        }\n\n        override fun hashCode(): Int {\n
return (startedAt.toDuration(timeSource.unit) + offset).hashCode()\n        }\n\n        override fun toString(): String =
\"DoubleTimeMark($startedAt${timeSource.unit.shortName()} + $offset, $timeSource)\"\n    }\n\n    override fun
markNow(): ComparableTimeMark = DoubleTimeMark(read(), this, Duration.ZERO)\n}\n\n/**\n * A time source
that has programmatically updatable readings. It is useful as a predictable source of time in tests.\n *\n * The current
reading value can be advanced by the specified duration amount with the operator [plusAssign]:\n *\n * ```\n * val
timeSource = TestTimeSource()\n * timeSource += 10.seconds\n * ```\n *\n * Implementation note: the current
reading value is stored as a [Long] number of nanoseconds,\n * thus it's capable to
 represent a time range of approximately \u00b1292 years.\n * Should the reading value overflow as the result of
[plusAssign] operation, an [IllegalStateException] is thrown.\n
*/\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\npublic class TestTimeSource : AbstractLongTimeSource(unit =
DurationUnit.NANOSECONDS) {\n    private var reading: Long = 0L\n\n    override fun read(): Long = reading\n\n
  /**\n     * Advances the current reading value of this time source by the specified [duration].\n     *\n     * [duration]
value is rounded down towards zero when converting it to a [Long] number of nanoseconds.\n     * For example, if
the duration being added is `0.6.nanoseconds`, the reading doesn't advance because\n     * the duration value is
rounded to zero nanoseconds.\n     *\n     * @throws IllegalStateException when the reading value overflows as the
result of this operation.\n     */\n    public operator fun plusAssign(duration: Duration) {\n        val longDelta =
```

duration.toLong(unit)\n
reading = if (longDelta != Long.MIN_VALUE && longDelta != Long.MAX_VALUE) {\n // when delta fits in long, add it as long\n val newReading = reading + longDelta\n if (reading xor longDelta >= 0 && reading xor newReading < 0) overflow(duration)\n newReading\n } else {\n val delta = duration.toDouble(unit)\n // when delta is greater than long, add it as double\n val newReading = reading + delta\n if (newReading > Long.MAX_VALUE || newReading < Long.MIN_VALUE) overflow(duration)\n newReading.toLong()\n }\n }\n\n private fun overflow(duration: Duration) {\n throw IllegalStateException(\"TestTimeSource will overflow if its reading ${reading}${unit.shortName()} is advanced by $duration.\")\n }\n}\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport kotlin.time.Duration.Companion.milliseconds\nimport kotlin.time.Duration.Companion.nanoseconds\n\n// Long time reading saturation math, shared between JVM and Native\n\ninternal fun saturatingAdd(longNs: Long, duration: Duration): Long {\n val durationNs = duration.inWholeNanoseconds\n if (longNs.isSaturated()) { // MIN_VALUE or MAX_VALUE - the reading is infinite\n return checkInfiniteSumDefined(longNs, duration, durationNs)\n }\n if (durationNs.isSaturated()) { // duration doesn't fit in Long nanos\n return saturatingAddInHalves(longNs, duration)\n }\n\n val result = longNs + durationNs\n if (((longNs xor result) and (durationNs xor result)) < 0) {\n return if (longNs < 0) Long.MIN_VALUE else Long.MAX_VALUE\n }\n return result\n}\n\nprivate fun checkInfiniteSumDefined(longNs: Long, duration: Duration, durationNs: Long): Long {\n if (duration.isInfinite() && (longNs xor durationNs < 0)) throw IllegalArgumentException(\"Summing infinities of different signs\")\n return longNs\n}\n\nprivate fun saturatingAddInHalves(longNs: Long, duration: Duration): Long {\n val half = duration / 2\n if (half.inWholeNanoseconds.isSaturated()) {\n // this will definitely saturate\n return (longNs + duration.toDouble(DurationUnit.NANOSECONDS)).toLong()\n } else {\n return saturatingAdd(saturatingAdd(longNs, half), duration - half)\n }\n}\n\ninternal fun saturatingDiff(valueNs: Long, originNs: Long): Duration {\n if (originNs.isSaturated()) { // MIN_VALUE or MAX_VALUE\n return -(originNs.toDuration(DurationUnit.DAYS)) // saturate to infinity\n }\n return saturatingFiniteDiff(valueNs, originNs)\n}\n\ninternal fun saturatingOriginsDiff(origin1Ns: Long, origin2Ns: Long): Duration {\n if (origin2Ns.isSaturated()) { // MIN_VALUE or MAX_VALUE\n if (origin1Ns == origin2Ns) return Duration.ZERO // saturated values of the same sign are considered equal\n return -(origin2Ns.toDuration(DurationUnit.DAYS)) // saturate to infinity\n }\n if (origin1Ns.isSaturated()) {\n return origin1Ns.toDuration(DurationUnit.DAYS)\n }\n return saturatingFiniteDiff(origin1Ns, origin2Ns)\n}\n\nprivate fun saturatingFiniteDiff(value1Ns: Long, value2Ns: Long): Duration {\n val result = value1Ns - value2Ns\n if ((result xor value1Ns) and (result xor value2Ns).inv() < 0) {\n val resultMs = value1Ns / NANOS_IN_MILLIS - value2Ns / NANOS_IN_MILLIS\n val resultNs = value1Ns % NANOS_IN_MILLIS - value2Ns % NANOS_IN_MILLIS\n return resultMs.milliseconds + resultNs.nanoseconds\n }\n return result.nanoseconds\n}\n\n@Suppress(\"NOTHING_TO_INLINE\")\nprivate inline fun Long.isSaturated(): Boolean =\n (this - 1) or 1 == Long.MAX_VALUE // == either MAX_VALUE or MIN_VALUE\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.time\n\nimport kotlin.contracts.*\n\n/**\n * Executes the given function [block] and returns the duration of elapsed time interval.\n *\n * The elapsed time is measured with [TimeSource.Monotonic].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\npublic inline fun measureTime(block: () -> Unit): Duration {\n contract {\n callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n }\n return TimeSource.Monotonic.measureTime(block)\n}\n\n\n/**\n * Executes the given function [block] and returns the duration of elapsed time interval.\n *\n * The elapsed time is measured with the specified `this` [TimeSource] instance.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\npublic inline fun TimeSource.measureTime(block: ()

-> Unit): Duration {\n   contract {\n   callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n   }\n\n   val mark = markNow()\n
    block()\n   return mark.elapsedNow()\n}\n\n/**\n * Executes the given function [block] and returns the duration of elapsed time interval.\n *\n * The elapsed time is measured with the specified `this` [TimeSource.Monotonic] instance.\n */\n@SinceKotlin(\"1.7\")\n@ExperimentalTime\npublic inline fun TimeSource.Monotonic.measureTime(block: () -> Unit): Duration {\n   contract {\n   callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n   }\n\n   val mark = markNow()\n   block()\n   return mark.elapsedNow()\n}\n\n\n/**\n * Data class representing a result of executing an action, along with the duration of elapsed time interval.\n *\n * @property value the result of the action.\n * @property duration the time elapsed to execute the action.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\npublic data class TimedValue<T>(val value: T, val duration: Duration)\n\n/**\n * Executes the given function [block] and returns an instance of [TimedValue] class, containing both\n * the result
 of the function execution and the duration of elapsed time interval.\n *\n * The elapsed time is measured with [TimeSource.Monotonic].\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\npublic inline fun <T> measureTimedValue(block: () -> T): TimedValue<T> {\n   contract {\n   callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n   }\n\n   return TimeSource.Monotonic.measureTimedValue(block)\n}\n\n/**\n * Executes the given [block] and returns an instance of [TimedValue] class, containing both\n * the result of function execution and the duration of elapsed time interval.\n *\n * The elapsed time is measured with the specified `this` [TimeSource] instance.\n */\n@SinceKotlin(\"1.3\")\n@ExperimentalTime\npublic inline fun <T> TimeSource.measureTimedValue(block: () -> T): TimedValue<T> {\n   contract {\n   callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n   }\n\n   val mark = markNow()\n   val result = block()\n   return TimedValue(result, mark.elapsedNow())\n}\n\n/**\n
 * Executes the given [block] and returns an instance of [TimedValue] class, containing both\n * the result of function execution and the duration of elapsed time interval.\n *\n * The elapsed time is measured with the specified `this` [TimeSource.Monotonic] instance.\n */\n@SinceKotlin(\"1.7\")\n@ExperimentalTime\npublic inline fun <T> TimeSource.Monotonic.measureTimedValue(block: () -> T): TimedValue<T> {\n   contract {\n   callsInPlace(block, InvocationKind.EXACTLY_ONCE)\n   }\n\n   val mark = markNow()\n   val result = block()\n   return TimedValue(result, mark.elapsedNow())\n}\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\nimport kotlin.coroutines.*\nimport kotlin.coroutines.intrinsics.*\nimport kotlin.native.concurrent.SharedImmutable\n\n/**\n * Defines deep recursive function that
 keeps its stack on the heap,\n * which allows very deep recursive computations that do not use the actual call stack.\n * To initiate a call to this deep recursive function use its [invoke] function.\n * As a rule of thumb, it should be used if recursion goes deeper than a thousand calls.\n *\n * The [DeepRecursiveFunction] takes one parameter of type [T] and returns a result of type [R].\n * The [block] of code defines the body of a recursive function. In this block\n * [callRecursive][DeepRecursiveScope.callRecursive] function can be used to make a recursive call\n * to the declared function. Other instances of [DeepRecursiveFunction] can be called\n * in this scope with `callRecursive` extension, too.\n *\n * For example, take a look at the following recursive tree class and a deeply\n * recursive instance of this tree with 100K nodes:\n *\n * ```\n * class Tree(val left: Tree? = null, val right: Tree? = null)\n * val deepTree = generateSequence(Tree()) { Tree(it) }.take(100_000).last()\n
 * ```\n *\n * A regular recursive function can be defined to compute a depth of a tree:\n *\n * ```\n * fun depth(t: Tree?): Int =\n *   if (t == null) 0 else max(depth(t.left), depth(t.right)) + 1\n * println(depth(deepTree)) // StackOverflowError\n * ```\n *\n * If this `depth` function is called for a `deepTree` it produces `StackOverflowError` because of deep recursion.\n * However, the `depth` function can be rewritten using `DeepRecursiveFunction` in the following way, and then\n * it successfully computes [`depth(deepTree)`][DeepRecursiveFunction.invoke] expression:\n *\n * ```\n * val depth = DeepRecursiveFunction<Tree?, Int> { t ->\n *   if (t == null) 0 else max(callRecursive(t.left),

```
callRecursive(t.right)) + 1\n * }\n * println(depth(deepTree)) // Ok\n * ```\n *\n * Deep recursive functions can also mutually call each other using a heap for the stack via\n * [callRecursive][DeepRecursiveScope.callRecursive] extension. For example, the\n * following pair of mutually
 recursive functions computes the number of tree nodes at even depth in the tree.\n *\n * ```\n * val mutualRecursion = object {\n *     val even: DeepRecursiveFunction<Tree?, Int> = DeepRecursiveFunction { t ->\n *         if (t == null) 0 else odd.callRecursive(t.left) + odd.callRecursive(t.right) + 1\n *     }\n *     val odd: DeepRecursiveFunction<Tree?, Int> = DeepRecursiveFunction { t ->\n *         if (t == null) 0 else even.callRecursive(t.left) + even.callRecursive(t.right)\n *     }\n * }\n * ```\n *\n * @param [T] the function parameter type.\n * @param [R] the function result type.\n * @param block the function body.\n */\n@SinceKotlin(\"1.7\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic class DeepRecursiveFunction<T, R>(\n    internal val block: suspend DeepRecursiveScope<T, R>.(T) -> R\n)\n\n/**\n * Initiates a call to this deep recursive function, forming a root of the call tree.\n *\n * This operator should not be used from inside of [DeepRecursiveScope] as
 it uses the call stack slot for\n * initial recursive invocation. From inside of [DeepRecursiveScope] use\n * [callRecursive][DeepRecursiveScope.callRecursive].\n */\n@SinceKotlin(\"1.7\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic operator fun <T, R> DeepRecursiveFunction<T, R>.invoke(value: T): R =\n    DeepRecursiveScopeImpl<T, R>(block, value).runCallLoop()\n\n/**\n * A scope class for [DeepRecursiveFunction] function declaration that defines [callRecursive] methods to\n * recursively call this function or another [DeepRecursiveFunction] putting the call activation frame on the heap.\n *\n * @param [T] function parameter type.\n * @param [R] function result type.\n */\n@RestrictsSuspension\n@SinceKotlin(\"1.7\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic sealed class DeepRecursiveScope<T, R> {\n    /**\n     * Makes recursive call to this [DeepRecursiveFunction] function putting the call activation frame on the heap,\n     * as opposed to the actual call
 stack that is used by a regular recursive call.\n     */\n    public abstract suspend fun callRecursive(value: T): R\n\n    /**\n     * Makes call to the specified [DeepRecursiveFunction] function putting the call activation frame on the heap,\n     * as opposed to the actual call stack that is used by a regular call.\n     */\n    public abstract suspend fun <U, S> DeepRecursiveFunction<U, S>.callRecursive(value: U): S\n\n    @Deprecated(\n        level = DeprecationLevel.ERROR,\n        message =\n        \"'invoke' should not be called from DeepRecursiveScope. \" +\n                \"Use 'callRecursive' to do recursion in the heap instead of the call stack.\",\n        replaceWith = ReplaceWith(\"this.callRecursive(value)\")\n    )\n    @Suppress(\"UNUSED_PARAMETER\")\n    public operator fun DeepRecursiveFunction<*, *>.invoke(value: Any?): Nothing =\n        throw UnsupportedOperationException(\"Should not be called from DeepRecursiveScope\")\n}\n\n// ================== Implementation
 ==================\n\nprivate typealias DeepRecursiveFunctionBlock = suspend DeepRecursiveScope<*, *>.(Any?) -> Any?\n\n@SharedImmutable\nprivate val UNDEFINED_RESULT = Result.success(COROUTINE_SUSPENDED)\n\n@Suppress(\"UNCHECKED_CAST\")\nprivate class DeepRecursiveScopeImpl<T, R>(\n    block: suspend DeepRecursiveScope<T, R>.(T) -> R,\n    value: T\n) : DeepRecursiveScope<T, R>(), Continuation<R> {\n    // Active function block\n    private var function: DeepRecursiveFunctionBlock = block as DeepRecursiveFunctionBlock\n\n    // Value to call function with\n    private var value: Any? = value\n\n    // Continuation of the current call\n    private var cont: Continuation<Any?>? = this as Continuation<Any?>\n\n    // Completion result (completion of the whole call stack)\n    private var result: Result<Any?> = UNDEFINED_RESULT\n\n    override val context: CoroutineContext\n        get() = EmptyCoroutineContext\n\n    override fun resumeWith(result: Result<R>) {\n        this.cont = null\n        this.result = result\n    }\n\n    override suspend fun callRecursive(value: T): R =
 suspendCoroutineUninterceptedOrReturn { cont ->\n        // calling the same function that is currently active\n        this.cont = cont as Continuation<Any?>\n        this.value = value\n        COROUTINE_SUSPENDED\n    }\n\n    override suspend fun <U, S> DeepRecursiveFunction<U, S>.callRecursive(value: U): S =
 suspendCoroutineUninterceptedOrReturn { cont ->\n        // calling another recursive function\n        val function =
```

block as DeepRecursiveFunctionBlock\n with(this@DeepRecursiveScopeImpl) {\n val currentFunction = this.function\n if (function !== currentFunction) {\n // calling a different function -- create a trampoline to restore function ref\n this.function = function\n this.cont = crossFunctionCompletion(currentFunction, cont as Continuation<Any?>)\n } else {\n // calling the same function -- direct\n this.cont = cont as Continuation<Any?>\n }\n this.value = value\n }\n COROUTINE_SUSPENDED\n }\n\n private fun crossFunctionCompletion(\n currentFunction: DeepRecursiveFunctionBlock,\n cont: Continuation<Any?>\n): Continuation<Any?> = Continuation(EmptyCoroutineContext) {\n this.function = currentFunction\n // When going back from a trampoline we cannot just call cont.resume (stack usage!)\n // We delegate the cont.resumeWith(it) call to runCallLoop\n this.cont = cont\n this.result = it\n }\n\n @Suppress(\"UNCHECKED_CAST\")\n fun runCallLoop(): R {\n while (true) {\n // Note: cont is set to null in DeepRecursiveScopeImpl.resumeWith when the whole computation completes\n val result = this.result\n val cont = this.cont\n ?: return (result as Result<R>).getOrThrow() // done -- final result\n // The order of comparison is important here for that case of rogue class with broken equals\n if (UNDEFINED_RESULT == result) {\n // call \"function\" with \"value\" using \"cont\" as completion\n val r = try {\n // This is block.startCoroutine(this, value, cont)\n function.startCoroutineUninterceptedOrReturn(this, value, cont)\n } catch (e: Throwable) {\n cont.resumeWithException(e)\n continue\n }\n // If the function returns without suspension -- calls its continuation immediately\n if (r !== COROUTINE_SUSPENDED)\n cont.resume(r as R)\n } else {\n // we returned from a crossFunctionCompletion trampoline -- call resume here\n this.result = UNDEFINED_RESULT // reset result back\n cont.resumeWith(result)\n }\n }\n }\n}\n","/*\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\n@file:kotlin.jvm.JvmName(\"NumbersKt\")\n@file:kotlin.jvm.JvmMultifileClass\npackage kotlin\n\nimport kotlin.math.sign\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Byte.floorDiv(other: Byte): Int = \n this.toInt().floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Byte.mod(other: Byte): Byte = \n this.toInt().mod(other.toInt()).toByte()\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Byte.floorDiv(other: Short): Int = \n this.toInt().floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Byte.mod(other: Short): Short = \n this.toInt().mod(other.toInt()).toShort()\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Byte.floorDiv(other: Int): Int = \n this.toInt().floorDiv(other)\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Byte.mod(other: Int): Int = \n this.toInt().mod(other)\n\n/** Divides this value by the other value, flooring the

result to an integer that is closer to negative infinity.
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Byte.floorDiv(other: Long): Long = \n   this.toLong().floorDiv(other)\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n
 * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Byte.mod(other: Long): Long = \n   this.toLong().mod(other)\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Short.floorDiv(other: Byte): Int = \n   this.toInt().floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Short.mod(other:
 Byte): Byte = \n   this.toInt().mod(other.toInt()).toByte()\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Short.floorDiv(other: Short): Int = \n   this.toInt().floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Short.mod(other: Short): Short = \n   this.toInt().mod(other.toInt()).toShort()\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic
 inline fun Short.floorDiv(other: Int): Int = \n   this.toInt().floorDiv(other)\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Short.mod(other: Int): Int = \n   this.toInt().mod(other)\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Short.floorDiv(other: Long): Long = \n   this.toLong().floorDiv(other)\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and
 has the absolute value less than the absolute value of the divisor.\n
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Short.mod(other: Long): Long = \n   this.toLong().mod(other)\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Int.floorDiv(other: Byte): Int = \n   this.floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Int.mod(other: Byte): Byte = \n   this.mod(other.toInt()).toByte()\n\n/** Divides this value
 by the other value, flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Int.floorDiv(other: Short): Int = \n   this.floorDiv(other.toInt())\n\n/**\n * Calculates the remainder of flooring

division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Int.mod(other: Short): Short = \n   this.mod(other.toInt()).toShort()\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Int.floorDiv(other: Int): Int {\n   var q = this / other\n
    if (this xor other < 0 && q * other != this) q-- \n   return q\n}\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Int.mod(other: Int): Int {\n   val r = this % other\n   return r + (other and (((r xor other) and (r or -r)) shr 31))\n}\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity. */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Int.floorDiv(other: Long): Long = \n   this.toLong().floorDiv(other)\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as
 the _divisor_ and has the absolute value less than the absolute value of the divisor.\n
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Int.mod(other: Long): Long = \n   this.toLong().mod(other)\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.floorDiv(other: Byte): Long = \n   this.floorDiv(other.toLong())\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.mod(other: Byte): Byte = \n   this.mod(other.toLong()).toByte()\n\n/**
 Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.floorDiv(other: Short): Long = \n   this.floorDiv(other.toLong())\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.mod(other: Short): Short = \n   this.mod(other.toLong()).toShort()\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.floorDiv(other: Int): Long = \n
    this.floorDiv(other.toLong())\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.mod(other: Int): Int = \n   this.mod(other.toLong()).toInt()\n\n/** Divides this value by the other value, flooring the result to an integer that is closer to negative infinity.
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Long.floorDiv(other: Long): Long {\n   var q = this / other\n   if (this xor other < 0 && q * other != this) q-- \n
 return q\n}\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the
 absolute value less than the absolute value of the divisor.\n
*/\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline

fun Long.mod(other: Long): Long {\n   val r = this % other\n   return r + (other and (((r xor other) and (r or -r)) shr 63))\n}\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n * \n * If the result cannot be represented exactly, it is rounded to the nearest representable number. In this case the absolute value of the result can be less than or _equal to_ the absolute value of the divisor.\n */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Float.mod(other: Float): Float {\n   val r = this % other\n   return if (r != 0.0.toFloat() && r.sign != other.sign) r +
 other else r\n}\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n * \n * If the result cannot be represented exactly, it is rounded to the nearest representable number. In this case the absolute value of the result can be less than or _equal to_ the absolute value of the divisor.\n */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Float.mod(other: Double): Double = \n   this.toDouble().mod(other)\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n * \n * If the result cannot be represented exactly, it is rounded to the nearest representable number. In
 this case the absolute value of the result can be less than or _equal to_ the absolute value of the divisor.\n */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Double.mod(other: Float): Double = \n   this.mod(other.toDouble())\n\n/**\n * Calculates the remainder of flooring division of this value by the other value.\n * \n * The result is either zero or has the same sign as the _divisor_ and has the absolute value less than the absolute value of the divisor.\n * \n * If the result cannot be represented exactly, it is rounded to the nearest representable number. In this case the absolute value of the result can be less than or _equal to_ the absolute value of the divisor.\n */\n@SinceKotlin(\"1.5\")\n@kotlin.internal.InlineOnly\n@kotlin.internal.IntrinsicConstEvaluation\npublic inline fun Double.mod(other: Double): Double {\n   val r = this % other\n   return if (r != 0.0 && r.sign != other.sign) r + other else r\n}\n\n","/*\n
 * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\nimport kotlin.internal.InlineOnly\n\n\n/**\n * Returns a hash code value for the object or zero if the object is `null`.\n *\n * @see Any.hashCode\n */\n@SinceKotlin(\"1.3\")\n@InlineOnly\npublic inline fun Any?.hashCode(): Int = this?.hashCode() ?: 0\n","/*\n * Copyright 2010-2020 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\n\n/**\n * Represents a version of the Kotlin standard library.\n *\n * [major], [minor] and [patch] are integer components of a version,\n * they must be non-negative and not greater than 255 ([MAX_COMPONENT_VALUE]).\n *\n * @constructor Creates a version from all three components.\n
 */\n@SinceKotlin(\"1.1\")\npublic class KotlinVersion(val major: Int, val minor: Int, val patch: Int) : Comparable<KotlinVersion> {\n   /**\n   * Creates a version from [major] and [minor] components, leaving [patch] component zero.\n   */\n   public constructor(major: Int, minor: Int) : this(major, minor, 0)\n\n   private val version = versionOf(major, minor, patch)\n\n   private fun versionOf(major: Int, minor: Int, patch: Int): Int {\n   require(major in 0..MAX_COMPONENT_VALUE && minor in 0..MAX_COMPONENT_VALUE && patch in 0..MAX_COMPONENT_VALUE) {\n   \"Version components are out of range: $major.$minor.$patch\"\n   }\n   return major.shl(16) + minor.shl(8) + patch\n   }\n\n   /**\n   * Returns the string representation of this version\n   */\n   override fun toString(): String = \"$major.$minor.$patch\"\n\n   override fun equals(other: Any?): Boolean {\n   if (this === other) return true\n   val otherVersion = (other as?
 KotlinVersion) ?: return false\n   return this.version == otherVersion.version\n   }\n\n   override fun hashCode(): Int = version\n\n   override fun compareTo(other: KotlinVersion): Int = version - other.version\n\n

/**\n * Returns `true` if this version is not less than the version specified\n * with the provided [major] and [minor] components.\n */\n public fun isAtLeast(major: Int, minor: Int): Boolean = // this.version >= versionOf(major, minor, 0)\n this.major > major || (this.major == major &&\n this.minor >= minor)\n\n /**\n * Returns `true` if this version is not less than the version specified\n * with the provided [major], [minor] and [patch] components.\n */\n public fun isAtLeast(major: Int, minor: Int, patch: Int): Boolean = // this.version >= versionOf(major, minor, patch)\n this.major > major || (this.major == major &&\n (this.minor > minor || this.minor == minor &&\n this.patch >= patch))\n\n companion object {\n /**\n * Maximum value a version component can have, a constant value 255.\n */\n // NOTE: Must be placed before CURRENT because its initialization requires this field being initialized in JS\n public const val MAX_COMPONENT_VALUE = 255\n\n /**\n * Returns the current version of the Kotlin standard library.\n */\n @kotlin.jvm.JvmField\n public val CURRENT: KotlinVersion = KotlinVersionCurrentValue.get()\n }\n}\n\n// this class is ignored during classpath normalization when considering whether to recompile dependencies in Kotlin build\nprivate object KotlinVersionCurrentValue {\n @kotlin.jvm.JvmStatic\n fun get(): KotlinVersion = KotlinVersion(1, 8, 10) // value is written here automatically during build\n}","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmName(\"LateinitKt\")\n@file:Suppress(\"unused\")\n\npackage kotlin\n\nimport kotlin.internal.InlineOnly\nimport kotlin.internal.AccessibleLateinitPropertyLiteral\nimport kotlin.reflect.KProperty0\n\n/**\n * Returns `true` if this lateinit property has been assigned a value, and `false` otherwise.\n *\n * Cannot be used in an inline function, to avoid binary compatibility issues.\n */\n@SinceKotlin(\"1.2\")\n@InlineOnly\ninline val @receiver:AccessibleLateinitPropertyLiteral KProperty0<*>.isInitialized: Boolean\n get() = throw NotImplementedError(\"Implementation is intrinsic\")\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmName(\"LazyKt\")\n@file:kotlin.jvm.JvmMultifileClass\n\npackage kotlin\n\nimport kotlin.reflect.KProperty\n\n/**\n * Represents a value with lazy initialization.\n *\n * To create an instance of [Lazy] use the [lazy] function.\n */\npublic interface Lazy<out T> {\n /**\n * Gets the lazily initialized value of the current Lazy instance.\n * Once the value was initialized it must not change during the rest of lifetime of this Lazy instance.\n */\n public val value: T\n\n /**\n * Returns `true` if a value for this Lazy instance has been already initialized, and `false` otherwise.\n * Once this function has returned `true` it stays `true` for the rest of lifetime of this Lazy instance.\n */\n public fun isInitialized(): Boolean\n}\n\n/**\n * Creates a new instance of the [Lazy] that is already initialized with the specified [value].\n */\npublic fun <T> lazyOf(value: T): Lazy<T> = InitializedLazyImpl(value)\n\n/**\n * An extension to delegate a read-only property of type [T] to an instance of [Lazy].\n *\n * This extension allows to use instances of Lazy for property delegation:\n * `val property: String by lazy { initializer }`\n */\n@kotlin.internal.InlineOnly\npublic inline operator fun <T> Lazy<T>.getValue(thisRef: Any?, property: KProperty<*>): T = value\n\n/**\n * Specifies how a [Lazy] instance synchronizes initialization among multiple threads.\n */\npublic enum class LazyThreadSafetyMode {\n\n /**\n * Locks are used to ensure that only a single thread can initialize the [Lazy] instance.\n */\n SYNCHRONIZED,\n\n /**\n * Initializer function can be called several times on concurrent access to uninitialized [Lazy] instance value,\n * but only the first returned value will be used as the value of [Lazy] instance.\n */\n PUBLICATION,\n\n /**\n * No locks are used to synchronize an access to the [Lazy] instance value; if the instance is accessed from multiple threads, its behavior is undefined.\n *\n * This mode should not be used unless the [Lazy] instance is guaranteed never to be initialized from more than one thread.\n */\n NONE,\n}\n\n\ninternal object UNINITIALIZED_VALUE\n\n// internal to be called from lazy in JS\ninternal class UnsafeLazyImpl<out T>(initializer: () -> T) : Lazy<T>, Serializable {\n private var initializer: (() -> T)? = initializer\n private var _value: Any? = UNINITIALIZED_VALUE\n\n override val value: T\n get() {\n if (_value ===

UNINITIALIZED_VALUE) {\n _value = initializer!!()\n initializer = null\n }\n @Suppress(\"UNCHECKED_CAST\")\n return _value as T\n }\n\n override fun isInitialized(): Boolean = _value !== UNINITIALIZED_VALUE\n\n override fun toString(): String = if (isInitialized()) value.toString() else \"Lazy value not initialized yet.\"\n\n private fun writeReplace(): Any = InitializedLazyImpl(value)\n}\n\ninternal class InitializedLazyImpl<out T>(override val value: T) : Lazy<T>, Serializable {\n\n override fun isInitialized(): Boolean = true\n\n override fun toString(): String = value.toString()\n\n}\n","/*\n * Copyright 2010-2019 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"NumbersKt\")\npackage kotlin\n\n/**\n * Counts the number of set bits in the binary representation of this [Int] number.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Int.countOneBits(): Int\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Int] number.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Int.countLeadingZeroBits(): Int\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the binary representation
of this [Int] number.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Int.countTrailingZeroBits(): Int\n\n/**\n * Returns a number having a single bit set in the position of the most significant set bit of this [Int] number,\n * or zero, if this number is zero.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Int.takeHighestOneBit(): Int\n\n/**\n * Returns a number having a single bit set in the position of the least significant set bit of this [Int] number,\n * or zero, if this number is zero.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Int.takeLowestOneBit(): Int\n\n/**\n * Rotates the binary representation of this [Int] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n *\n * Rotating the number left by a
negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n *\n * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 32)`\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Int.rotateLeft(bitCount: Int): Int\n\n\n/**\n * Rotates the binary representation of this [Int] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n *\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n *\n * Rotating by a multiple of [Int.SIZE_BITS] (32) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 32)`\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic
expect fun Int.rotateRight(bitCount: Int): Int\n\n\n/**\n * Counts the number of set bits in the binary representation of this [Long] number.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.countOneBits(): Int\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Long] number.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.countLeadingZeroBits(): Int\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Long] number.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.countTrailingZeroBits(): Int\n\n/**\n * Returns a number having a single bit set in the position of the most significant set bit of this [Long] number,\n * or zero, if this number

is zero.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.takeHighestOneBit(): Long\n\n/**\n * Returns a number having a single bit set in the position of the least significant set bit of this [Long] number,\n * or zero, if this number is zero.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.takeLowestOneBit(): Long\n\n/**\n * Rotates the binary representation of this [Long] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n *\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n *\n * Rotating by a multiple of [Long.SIZE_BITS] (64) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 64)`\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.rotateLeft(bitCount: Int): Long\n\n/**\n * Rotates the binary representation of this [Long] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n *\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n *\n * Rotating by a multiple of [Long.SIZE_BITS] (64) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 64)`\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun Long.rotateRight(bitCount: Int): Long\n\n/**\n * Counts the number of set bits in the binary representation of this [Byte] number.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.countOneBits(): Int = (toInt() and 0xFF).countOneBits()\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Byte] number.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.countLeadingZeroBits(): Int = (toInt() and 0xFF).countLeadingZeroBits() - (Int.SIZE_BITS - Byte.SIZE_BITS)\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Byte] number.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.countTrailingZeroBits(): Int = (toInt() or 0x100).countTrailingZeroBits()\n\n/**\n * Returns a number having a single bit set in the position of the most significant set bit of this [Byte] number,\n * or zero, if this number is zero.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.takeHighestOneBit(): Byte = (toInt() and 0xFF).takeHighestOneBit().toByte()\n\n/**\n * Returns a number having a single bit set in the position of the least significant set bit of this [Byte] number,\n * or zero, if this number is zero.\n */\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Byte.takeLowestOneBit(): Byte = toInt().takeLowestOneBit().toByte()\n\n/**\n * Rotates the binary representation of this [Byte] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n *\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n *\n * Rotating by a multiple of [Byte.SIZE_BITS] (8)
returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 8)`\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Byte.rotateLeft(bitCount: Int): Byte =\n   (toInt().shl(bitCount and 7) or (toInt() and 0xFF).ushr(8 - (bitCount and 7))).toByte()\n\n/**\n * Rotates the binary representation of this [Byte] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most

significant bits on the left side.\n *\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n *\n * Rotating by a multiple of [Byte.SIZE_BITS] (8) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 8)`\n
*/\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Byte.rotateRight(bitCount:
Int): Byte =\n   (toInt().shl(8 - (bitCount and 7)) or (toInt() and 0xFF).ushr(bitCount and 7)).toByte()\n\n/**\n * Counts the number of set bits in the binary representation of this [Short] number.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.countOneBits(): Int = (toInt() and 0xFFFF).countOneBits()\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [Short] number.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.countLeadingZeroBits(): Int =\n   (toInt() and 0xFFFF).countLeadingZeroBits() - (Int.SIZE_BITS - Short.SIZE_BITS)\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [Short] number.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun Short.countTrailingZeroBits(): Int = (toInt() or 0x10000).countTrailingZeroBits()\n\n/**\n * Returns a number having a single bit set in the position of the most significant set bit of this [Short] number,\n * or zero, if this number is zero.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.takeHighestOneBit(): Short = (toInt() and 0xFFFF).takeHighestOneBit().toShort()\n\n/**\n * Returns a number having a single bit set in the position of the least significant set bit of this [Short] number,\n * or zero, if this number is zero.\n
*/\n@SinceKotlin(\"1.4\")\n@WasExperimental(ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun Short.takeLowestOneBit(): Short = toInt().takeLowestOneBit().toShort()\n\n/**\n * Rotates the binary representation of this [Short] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from
the left side reenter the number as the least significant bits on the right side.\n *\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n *\n * Rotating by a multiple of [Short.SIZE_BITS] (16) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 16)`\n
*/\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Short.rotateLeft(bitCount: Int): Short =\n   (toInt().shl(bitCount and 15) or (toInt() and 0xFFFF).ushr(16 - (bitCount and 15))).toShort()\n\n/**\n * Rotates the binary representation of this [Short] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n *\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n
* `number.rotateRight(-n) == number.rotateLeft(n)`\n *\n * Rotating by a multiple of [Short.SIZE_BITS] (16) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 16)`\n
/\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic fun Short.rotateRight(bitCount: Int): Short =\n   (toInt().shl(16 - (bitCount and 15)) or (toInt() and 0xFFFF).ushr(bitCount and 15)).toShort()\n","/\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin\nimport kotlin.internal.RequireKotlin\nimport kotlin.internal.RequireKotlinVersionKind\n\n@kotlin.internal.InlineOnly\n@SinceKotlin(\"1.2\")\n@Suppress(\"INVISIBLE_MEMBER\", \"INVISIBLE_REFERENCE\")\npublic inline fun <R> suspend(noinline block: suspend () -> R): suspend () -> R = block\n","/*\n * Copyright 2010-2018 JetBrains

s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmName(\"TuplesKt\")\n\npackage kotlin\n\n\n/**\n * Represents a generic pair of two values.\n *\n * There is no meaning attached to values in this class, it can be used for any purpose.\n * Pair exhibits value semantics, i.e. two pairs are equal if both components are equal.\n *\n * An example of decomposing it into values:\n * @sample samples.misc.Tuples.pairDestructuring\n *\n * @param A type of the first value.\n * @param B type of the second value.\n * @property first First value.\n * @property second Second value.\n * @constructor Creates a new instance of Pair.\n */\npublic data class Pair<out A, out B>(\n public val first: A,\n public val second: B\n) : Serializable {\n\n /**\n * Returns string representation of the [Pair] including its [first] and [second] values.\n */\n public override fun toString(): String = \"($first, $second)\"\n}\n\n/**\n * Creates a tuple of type [Pair] from this and [that].\n *\n * This can be useful for creating [Map] literals with less noise, for example:\n * @sample samples.collections.Maps.Instantiation.mapFromPairs\n */\npublic infix fun <A, B> A.to(that: B): Pair<A, B> = Pair(this, that)\n\n/**\n * Converts this pair into a list.\n * @sample samples.misc.Tuples.pairToList\n */\npublic fun <T> Pair<T, T>.toList(): List<T> = listOf(first, second)\n\n/**\n * Represents a triad of values\n *\n * There is no meaning attached to values in this class, it can be used for any purpose.\n * Triple exhibits value semantics, i.e. two triples are equal if all three components are equal.\n * An example of decomposing it into values:\n * @sample samples.misc.Tuples.tripleDestructuring\n *\n * @param A type of the first value.\n * @param B type of the second value.\n * @param C type of the third value.\n * @property first First value.\n * @property second Second value.\n * @property third Third value.\n */\npublic data class Triple<out A, out B, out C>(\n public val first: A,\n public val second: B,\n public val third: C\n) : Serializable {\n\n /**\n * Returns string representation of the [Triple] including its [first], [second] and [third] values.\n */\n public override fun toString(): String = \"($first, $second, $third)\"\n}\n\n/**\n * Converts this triple into a list.\n * @sample samples.misc.Tuples.tripleToList\n */\npublic fun <T> Triple<T, T, T>.toList(): List<T> = listOf(first, second, third)\n","/*\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin.ranges\n\n\n\nimport kotlin.internal.*\n\n/**\n * A range of values of type `UInt`.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@OptIn(ExperimentalStdlibApi::class)\npublic
class UIntRange(start: UInt, endInclusive: UInt) : UIntProgression(start, endInclusive, 1), ClosedRange<UInt>, OpenEndRange<UInt> {\n override val start: UInt get() = first\n override val endInclusive: UInt get() = last\n\n @SinceKotlin(\"1.7\")\n @ExperimentalStdlibApi\n @Deprecated(\"Can throw an exception when it's impossible to represent the value with UInt type, for example, when the range includes MAX_VALUE. It's recommended to use 'endInclusive' property that doesn't throw.\")\n override val endExclusive: UInt get() {\n if (last == UInt.MAX_VALUE) error(\"Cannot return the exclusive upper bound of a range that includes MAX_VALUE.\")\n return last + 1u\n }\n\n override fun contains(value: UInt): Boolean = first <= value && value <= last\n\n /** \n * Checks if the range is empty.\n \n * The range is empty if its start value is greater than
the end value.\n */\n override fun isEmpty(): Boolean = first > last\n\n override fun equals(other: Any?): Boolean =\n other is UIntRange && (isEmpty() && other.isEmpty() ||\n first == other.first && last == other.last)\n\n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 * first.toInt() + last.toInt())\n\n override fun toString(): String = \"$first..$last\"\n\n companion object {\n /** An empty range of values of type UInt. */\n public val EMPTY: UIntRange = UIntRange(UInt.MAX_VALUE, UInt.MIN_VALUE)\n }\n}\n\n/**\n * A progression of values of type `UInt`.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic open class UIntProgression\ninternal constructor(\n start: UInt,\n endInclusive: UInt,\n step: Int\n) : Iterable<UInt> {\n init {\n if (step == 0.toInt()) throw kotlin.IllegalArgumentException(\"Step must be non-zero.\")\n if (step

== Int.MIN_VALUE)
 throw kotlin.IllegalArgumentException(\"Step must be greater than Int.MIN_VALUE to avoid overflow on negation.\")\n }\n\n /**\n * The first element in the progression.\n */\n public val first: UInt = start\n\n /**\n * The last element in the progression.\n */\n public val last: UInt = getProgressionLastElement(start, endInclusive, step)\n\n /**\n * The step of the progression.\n */\n public val step: Int = step\n\n final override fun iterator(): Iterator<UInt> = UIntProgressionIterator(first, last, step)\n\n /** \n * Checks if the progression is empty.\n \n * Progression with a positive step is empty if its first element is greater than the last element.\n * Progression with a negative step is empty if its first element is less than the last element.\n */\n public open fun isEmpty(): Boolean = if (step > 0) first > last else first < last\n\n override fun equals(other: Any?): Boolean =\n other is
 UIntProgression && (isEmpty() && other.isEmpty() ||\n first == other.first && last == other.last && step == other.step)\n\n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 * (31 * first.toInt() + last.toInt()) + step.toInt())\n\n override fun toString(): String = if (step > 0) \"$first..$last step $step\" else \"$first downTo $last step ${-step}\"\n\n companion object {\n /**\n * Creates UIntProgression within the specified bounds of a closed range.\n\n * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding it, with the specified [step].\n * In order to go backwards the [step] must be negative.\n *\n * [step] must be greater than `Int.MIN_VALUE` and not equal to zero.\n */\n public fun fromClosedRange(rangeStart: UInt, rangeEnd: UInt, step: Int): UIntProgression = UIntProgression(rangeStart, rangeEnd, step)\n }\n}\n\n\n/**\n * An
 iterator over a progression of values of type `UInt`.\n * @property step the number by which the value is incremented on each step.\n */\n@SinceKotlin(\"1.3\")\nprivate class UIntProgressionIterator(first: UInt, last: UInt, step: Int) : Iterator<UInt> {\n private val finalElement = last\n private var hasNext: Boolean = if (step > 0) first <= last else first >= last\n private val step = step.toUInt() // use 2-complement math for negative steps\n private var next = if (hasNext) first else finalElement\n\n override fun hasNext(): Boolean = hasNext\n\n override fun next(): UInt {\n val value = next\n if (value == finalElement) {\n if (!hasNext) throw kotlin.NoSuchElementException()\n hasNext = false\n } else {\n next += step\n }\n return value\n }\n}\n\n","/*\n * Copyright 2010-2023 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0
 license that can be found in the license/LICENSE.txt file.\n */\n\n// Auto-generated file. DO NOT EDIT!\n\npackage kotlin.ranges\n\n\n\nimport kotlin.internal.*\n\n/**\n * A range of values of type `ULong`.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@OptIn(ExperimentalStdlibApi::class)\npublic class ULongRange(start: ULong, endInclusive: ULong) : ULongProgression(start, endInclusive, 1), ClosedRange<ULong>, OpenEndRange<ULong> {\n override val start: ULong get() = first\n override val endInclusive: ULong get() = last\n \n @SinceKotlin(\"1.7\")\n @ExperimentalStdlibApi\n @Deprecated(\"Can throw an exception when it's impossible to represent the value with ULong type, for example, when the range includes MAX_VALUE. It's recommended to use 'endInclusive' property that doesn't throw.\")\n override val endExclusive: ULong get() {\n if (last == ULong.MAX_VALUE) error(\"Cannot return the exclusive upper bound of a range that includes
 MAX_VALUE.\")\n return last + 1u\n }\n\n override fun contains(value: ULong): Boolean = first <= value && value <= last\n\n /** \n * Checks if the range is empty.\n \n * The range is empty if its start value is greater than the end value.\n */\n override fun isEmpty(): Boolean = first > last\n\n override fun equals(other: Any?): Boolean =\n other is ULongRange && (isEmpty() && other.isEmpty() ||\n first == other.first && last == other.last)\n\n override fun hashCode(): Int =\n if (isEmpty()) -1 else (31 * (first xor (first shr 32)).toInt() + (last xor (last shr 32)).toInt())\n\n override fun toString(): String = \"$first..$last\"\n\n companion object {\n /** An empty range of values of type ULong. */\n public val EMPTY: ULongRange = ULongRange(ULong.MAX_VALUE, ULong.MIN_VALUE)\n }\n}\n\n/**\n * A progression of values of type `ULong`.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic

```
 open class ULongProgression\ninternal constructor(\n    start: ULong,\n    endInclusive: ULong,\n    step: Long\n) : Iterable<ULong> {\n    init {\n        if (step == 0.toLong()) throw kotlin.IllegalArgumentException(\"Step must be non-zero.\")\n        if (step == Long.MIN_VALUE) throw kotlin.IllegalArgumentException(\"Step must be greater than Long.MIN_VALUE to avoid overflow on negation.\")\n    }\n\n    /**\n     * The first element in the progression.\n     */\n    public val first: ULong = start\n\n    /**\n     * The last element in the progression.\n     */\n    public val last: ULong = getProgressionLastElement(start, endInclusive, step)\n\n    /**\n     * The step of the progression.\n     */\n    public val step: Long = step\n\n    final override fun iterator(): Iterator<ULong> = ULongProgressionIterator(first, last, step)\n\n    /** \n     * Checks if the progression is empty.\n     \n     * Progression with a positive step is empty if its
 first element is greater than the last element.\n     * Progression with a negative step is empty if its first element is less than the last element.\n     */\n    public open fun isEmpty(): Boolean = if (step > 0) first > last else first < last\n\n    override fun equals(other: Any?): Boolean =\n        other is ULongProgression && (isEmpty() && other.isEmpty() ||\n                first == other.first && last == other.last && step == other.step)\n\n    override fun hashCode(): Int =\n        if (isEmpty()) -1 else (31 * (31 * (first xor (first shr 32)).toInt() + (last xor (last shr 32)).toInt()) + (step xor (step ushr 32)).toInt())\n\n    override fun toString(): String = if (step > 0) \"$first..$last step $step\" else \"$first downTo $last step ${-step}\"\n\n    companion object {\n        /**\n         * Creates ULongProgression within the specified bounds of a closed range.\n\n         * The progression starts with the [rangeStart] value and goes toward the [rangeEnd] value not excluding
 it, with the specified [step].\n         * In order to go backwards the [step] must be negative.\n         *\n         * [step] must be greater than `Long.MIN_VALUE` and not equal to zero.\n         */\n        public fun fromClosedRange(rangeStart: ULong, rangeEnd: ULong, step: Long): ULongProgression = ULongProgression(rangeStart, rangeEnd, step)\n    }\n}\n\n\n/**\n * An iterator over a progression of values of type `ULong`.\n * @property step the number by which the value is incremented on each step.\n */\n@SinceKotlin(\"1.3\")\nprivate class ULongProgressionIterator(first: ULong, last: ULong, step: Long) : Iterator<ULong> {\n    private val finalElement = last\n    private var hasNext: Boolean = if (step > 0) first <= last else first >= last\n    private val step = step.toULong() // use 2-complement math for negative steps\n    private var next = if (hasNext) first else finalElement\n\n    override fun hasNext(): Boolean = hasNext\n\n    override fun next(): ULong {\n
    val value = next\n        if (value == finalElement) {\n            if (!hasNext) throw kotlin.NoSuchElementException()\n            hasNext = false\n        } else {\n            next += step\n        }\n        return value\n    }\n}\n\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.math\n\n/**\n * Returns the smaller of two values.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun min(a: UInt, b: UInt): UInt {\n    return minOf(a, b)\n}\n\n/**\n * Returns the smaller of two values.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun min(a: ULong, b: ULong): ULong {\n    return minOf(a, b)\n}\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic
 inline fun max(a: UInt, b: UInt): UInt {\n    return maxOf(a, b)\n}\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun max(a: ULong, b: ULong): ULong {\n    return maxOf(a, b)\n}","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:kotlin.jvm.JvmName(\"UNumbersKt\")\npackage kotlin\n\n/**\n * Counts the number of set bits in the binary representation of this [UInt] number.\n
```

*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.countOneBits(): Int = toInt().countOneBits()\n\n/**\n * Counts
the number of consecutive most significant bits that are zero in the binary representation of this [UInt] number.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.countLeadingZeroBits(): Int = toInt().countLeadingZeroBits()\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [UInt] number.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.countTrailingZeroBits(): Int = toInt().countTrailingZeroBits()\n\n/**\n * Returns a number having a single bit set in the position of the most significant set bit of this [UInt] number,\n * or zero, if this number is zero.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun UInt.takeHighestOneBit(): UInt = toInt().takeHighestOneBit().toUInt()\n\n/**\n * Returns a number having a single bit set in the position of the least significant set bit of this [UInt] number,\n * or zero, if this number is zero.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.takeLowestOneBit(): UInt = toInt().takeLowestOneBit().toUInt()\n\n/**\n * Rotates the binary representation of this [UInt] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n *\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n *\n * Rotating by a multiple of [UInt.SIZE_BITS] (32) returns the same number, or more generally\n * `number.rotateLeft(n)
== number.rotateLeft(n % 32)`\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class, ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UInt.rotateLeft(bitCount: Int): UInt = toInt().rotateLeft(bitCount).toUInt()\n\n\n/**\n * Rotates the binary representation of this [UInt] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n *\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n *\n * Rotating by a multiple of [UInt.SIZE_BITS] (32) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 32)`\n
*/\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class, ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline
fun UInt.rotateRight(bitCount: Int): UInt = toInt().rotateRight(bitCount).toUInt()\n\n\n/**\n * Counts the number of set bits in the binary representation of this [ULong] number.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.countOneBits(): Int = toLong().countOneBits()\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [ULong] number.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.countLeadingZeroBits(): Int = toLong().countLeadingZeroBits()\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [ULong] number.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic
inline fun ULong.countTrailingZeroBits(): Int = toLong().countTrailingZeroBits()\n\n/**\n * Returns a number having a single bit set in the position of the most significant set bit of this [ULong] number,\n * or zero, if this number is zero.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class,

ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.takeHighestOneBit(): ULong = toLong().takeHighestOneBit().toULong()\n\n/**\n * Returns a number having a single bit set in the position of the least significant set bit of this [ULong] number,\n * or zero, if this number is zero.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.takeLowestOneBit(): ULong = toLong().takeLowestOneBit().toULong()\n\n/**\n * Rotates the binary representation of this [ULong] number left by the specified [bitCount]
number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n *\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n *\n * Rotating by a multiple of [ULong.SIZE_BITS] (64) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 64)`\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class, ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.rotateLeft(bitCount: Int): ULong = toLong().rotateLeft(bitCount).toULong()\n\n/**\n * Rotates the binary representation of this [ULong] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n *\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n *\n * Rotating by a multiple of [ULong.SIZE_BITS] (64) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 64)`\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class, ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun ULong.rotateRight(bitCount: Int): ULong = toLong().rotateRight(bitCount).toULong()\n\n/**\n * Counts the number of set bits in the binary representation of this [UByte] number.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.countOneBits(): Int = toUInt().countOneBits()\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [UByte] number.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.countLeadingZeroBits(): Int = toByte().countLeadingZeroBits()\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [UByte] number.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.countTrailingZeroBits(): Int = toByte().countTrailingZeroBits()\n\n/**\n * Returns a number having a single bit set in the position of the most significant set bit of this [UByte] number,\n * or zero, if this number is zero.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.takeHighestOneBit(): UByte = toInt().takeHighestOneBit().toUByte()\n\n/**\n * Returns a number having a single bit set in the position of the least significant set bit of this [UByte] number,\n * or zero, if this number is zero.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.takeLowestOneBit(): UByte = toInt().takeLowestOneBit().toUByte()\n\n/**\n * Rotates the binary representation of this [UByte] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant bits on the right side.\n *\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n *\n * Rotating by a multiple of [UByte.SIZE_BITS] (8) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 8)`\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class,

ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic
inline fun UByte.rotateLeft(bitCount: Int): UByte = toByte().rotateLeft(bitCount).toUByte()\n\n/**\n * Rotates the binary representation of this [UByte] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n *\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n *\n * Rotating by a multiple of [UByte.SIZE_BITS] (8) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 8)`\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class, ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UByte.rotateRight(bitCount: Int): UByte = toByte().rotateRight(bitCount).toUByte()\n\n/**\n * Counts the number of set bits in the
binary representation of this [UShort] number.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.countOneBits(): Int = toUInt().countOneBits()\n\n/**\n * Counts the number of consecutive most significant bits that are zero in the binary representation of this [UShort] number.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.countLeadingZeroBits(): Int = toShort().countLeadingZeroBits()\n\n/**\n * Counts the number of consecutive least significant bits that are zero in the binary representation of this [UShort] number.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.countTrailingZeroBits(): Int = toShort().countTrailingZeroBits()\n\n/**\n * Returns
a number having a single bit set in the position of the most significant set bit of this [UShort] number,\n * or zero, if this number is zero.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.takeHighestOneBit(): UShort = toInt().takeHighestOneBit().toUShort()\n\n/**\n * Returns a number having a single bit set in the position of the least significant set bit of this [UShort] number,\n * or zero, if this number is zero.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class, ExperimentalStdlibApi::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.takeLowestOneBit(): UShort = toInt().takeLowestOneBit().toUShort()\n\n/**\n * Rotates the binary representation of this [UShort] number left by the specified [bitCount] number of bits.\n * The most significant bits pushed out from the left side reenter the number as the least significant
bits on the right side.\n *\n * Rotating the number left by a negative bit count is the same as rotating it right by the negated bit count:\n * `number.rotateLeft(-n) == number.rotateRight(n)`\n *\n * Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more generally\n * `number.rotateLeft(n) == number.rotateLeft(n % 16)`\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class, ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.rotateLeft(bitCount: Int): UShort = toShort().rotateLeft(bitCount).toUShort()\n\n/**\n * Rotates the binary representation of this [UShort] number right by the specified [bitCount] number of bits.\n * The least significant bits pushed out from the right side reenter the number as the most significant bits on the left side.\n *\n * Rotating the number right by a negative bit count is the same as rotating it left by the negated bit count:\n * `number.rotateRight(-n) == number.rotateLeft(n)`\n
*\n * Rotating by a multiple of [UShort.SIZE_BITS] (16) returns the same number, or more generally\n * `number.rotateRight(n) == number.rotateRight(n % 16)`\n */\n@SinceKotlin(\"1.6\")\n@WasExperimental(ExperimentalStdlibApi::class, ExperimentalUnsignedTypes::class)\n@kotlin.internal.InlineOnly\npublic inline fun UShort.rotateRight(bitCount: Int): UShort = toShort().rotateRight(bitCount).toUShort()\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be

found in the license/LICENSE.txt file.\n */\n\npackage kotlin.internal\n\n// (a - b) mod c\nprivate fun differenceModulo(a: UInt, b: UInt, c: UInt): UInt {\n   val ac = a % c\n   val bc = b % c\n   return if (ac >= bc) ac - bc else ac - bc + c\n}\n\nprivate fun differenceModulo(a: ULong, b: ULong, c: ULong): ULong {\n   val ac = a % c\n   val bc = b % c\n   return if (ac >= bc) ac - bc else ac - bc + c\n}\n\n/**\n
 * Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in the range\n * from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative\n * [step].\n *\n * No validation on passed parameters is performed. The given parameters should satisfy the condition:\n *\n * - either `step > 0` and `start <= end`,\n * - or `step < 0` and `start >= end`.\n *\n * @param start first element of the progression\n * @param end ending bound for the progression\n * @param step increment, or difference of successive elements in the progression\n * @return the final element of the progression\n * @suppress\n */\n@PublishedApi\n@SinceKotlin(\"1.3\")\ninternal fun getProgressionLastElement(start: UInt, end: UInt, step: Int): UInt = when {\n   step > 0 -> if (start >= end) end else end - differenceModulo(end, start, step.toUInt())\n   step < 0 -> if (start <= end) end else end + differenceModulo(start,
 end, (-step).toUInt())\n   else -> throw kotlin.IllegalArgumentException(\"Step is zero.\")\n}\n\n/**\n * Calculates the final element of a bounded arithmetic progression, i.e. the last element of the progression which is in the range\n * from [start] to [end] in case of a positive [step], or from [end] to [start] in case of a negative\n * [step].\n *\n * No validation on passed parameters is performed. The given parameters should satisfy the condition:\n *\n * - either `step > 0` and `start <= end`,\n * - or `step < 0` and `start >= end`.\n *\n * @param start first element of the progression\n * @param end ending bound for the progression\n * @param step increment, or difference of successive elements in the progression\n * @return the final element of the progression\n * @suppress\n */\n@PublishedApi\n@SinceKotlin(\"1.3\")\ninternal fun getProgressionLastElement(start: ULong, end: ULong, step: Long): ULong = when {\n   step > 0 -> if (start >= end) end else end - differenceModulo(end,
 start, step.toULong())\n   step < 0 -> if (start <= end) end else end + differenceModulo(start, end, (-step).toULong())\n   else -> throw kotlin.IllegalArgumentException(\"Step is zero.\")\n}\n\n","/*\n * Copyright 2010-2021 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmName(\"UStringsKt\") // string representation of unsigned numbers\n\npackage kotlin.text\n\n/**\n * Returns a string representation of this [Byte] value in the specified [radix].\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n//@kotlin.internal.InlineOnly
\npublic /*inline*/ fun UByte.toString(radix: Int): String = this.toInt().toString(radix)\n\n/**\n * Returns a string representation of this [Short] value in the specified
 [radix].\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n//@kotlin.internal.InlineOnly
\npublic /*inline*/ fun UShort.toString(radix: Int): String = this.toInt().toString(radix)\n\n\n/**\n * Returns a string representation of this [Int] value in the specified [radix].\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\n//@kotlin.internal.InlineOnly
\npublic /*inline*/ fun UInt.toString(radix: Int): String = this.toLong().toString(radix)\n\n/**\n * Returns a string representation of this [Long] value in the specified [radix].\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for number to string conversion.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic
 fun ULong.toString(radix: Int): String = ulongToString(this.toLong(), checkRadix(radix))\n\n\n/**\n * Parses the string as a signed [UByte] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n
*/\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUByte(): UByte = toUByteOrNull() ?: numberFormatError(this)\n\n/**\n * Parses the string as a signed [UByte] number and

returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUByte(radix: Int): UByte = toUByteOrNull(radix) ?: numberFormatError(this)\n\n\n/**\n * Parses the string as a [UShort] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUShort(): UShort = toUShortOrNull() ?: numberFormatError(this)\n\n/**\n * Parses the string as a [UShort] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUShort(radix: Int): UShort = toUShortOrNull(radix) ?: numberFormatError(this)\n\n/**\n * Parses the string as an [UInt] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUInt(): UInt = toUIntOrNull() ?: numberFormatError(this)\n\n/**\n * Parses the string as an [UInt] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUInt(radix: Int): UInt = toUIntOrNull(radix) ?: numberFormatError(this)\n\n/**\n * Parses the string as a [ULong] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toULong(): ULong = toULongOrNull() ?: numberFormatError(this)\n\n/**\n * Parses the string as a [ULong] number and returns the result.\n * @throws NumberFormatException if the string is not a valid representation of a number.\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toULong(radix: Int): ULong = toULongOrNull(radix) ?: numberFormatError(this)\n\n\n\n\n\n/**\n * Parses the string as an [UByte] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUByteOrNull(): UByte? = toUByteOrNull(radix = 10)\n\n/**\n * Parses the string as an [UByte] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUByteOrNull(radix: Int): UByte? {\n   val int = this.toUIntOrNull(radix) ?: return null\n   if (int > UByte.MAX_VALUE) return null\n   return int.toUByte()\n}\n\n/**\n * Parses the string as an [UShort] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUShortOrNull(): UShort? = toUShortOrNull(radix = 10)\n\n/**\n * Parses the string as an [UShort] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUShortOrNull(radix: Int): UShort? {\n   val int = this.toUIntOrNull(radix) ?: return null\n   if (int > UShort.MAX_VALUE) return null\n   return int.toUShort()\n}\n\n/**\n * Parses the string as an [UInt] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic

```
fun String.toUIntOrNull(): UInt? = toUIntOrNull(radix = 10)\n\n/**\n * Parses the string as an [UInt] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toUIntOrNull(radix: Int): UInt? {\n    checkRadix(radix)\n\n    val length = this.length\n    if (length == 0) return null\n\n    val limit: UInt = UInt.MAX_VALUE\n    val start: Int\n\n    val firstChar = this[0]\n    if (firstChar < '0') {\n        if (length == 1 || firstChar != '+') return null\n        start = 1\n    } else {\n        start = 0\n    }\n\n    val limitForMaxRadix = 119304647u  //  limit / 36\n\n    var limitBeforeMul = limitForMaxRadix\n    val uradix = radix.toUInt()\n    var result = 0u\n    for (i
 in start until length) {\n        val digit = digitOf(this[i], radix)\n\n        if (digit < 0) return null\n        if (result > limitBeforeMul) {\n            if (limitBeforeMul == limitForMaxRadix) {\n                limitBeforeMul = limit / uradix\n\n                if (result > limitBeforeMul) {\n                    return null\n                }\n            } else {\n                return null\n            }\n        }\n\n        result *= uradix\n\n        val beforeAdding = result\n        result += digit.toUInt()\n        if (result < beforeAdding) return null // overflow has happened\n    }\n\n    return result\n}\n\n/**\n * Parses the string as an [ULong] number and returns the result\n * or `null` if the string is not a valid representation of a number.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toULongOrNull(): ULong? = toULongOrNull(radix = 10)\n\n/**\n * Parses the string as an [ULong] number and returns the result\n
 * or `null` if the string is not a valid representation of a number.\n *\n * @throws IllegalArgumentException when [radix] is not a valid radix for string to number conversion.\n */\n@SinceKotlin(\"1.5\")\n@WasExperimental(ExperimentalUnsignedTypes::class)\npublic fun String.toULongOrNull(radix: Int): ULong? {\n    checkRadix(radix)\n\n    val length = this.length\n    if (length == 0) return null\n\n    val limit: ULong = ULong.MAX_VALUE\n    val start: Int\n\n    val firstChar = this[0]\n    if (firstChar < '0') {\n        if (length == 1 || firstChar != '+') return null\n        start = 1\n    } else {\n        start = 0\n    }\n\n\n    val limitForMaxRadix = 512409557603043100uL  //  limit / 36\n\n    var limitBeforeMul = limitForMaxRadix\n    val uradix = radix.toULong()\n    var result = 0uL\n    for (i in start until length) {\n        val digit = digitOf(this[i], radix)\n\n        if (digit < 0) return null\n        if (result > limitBeforeMul) {\n            if (limitBeforeMul
 == limitForMaxRadix) {\n                limitBeforeMul = limit / uradix\n\n                if (result > limitBeforeMul) {\n                    return null\n                }\n            } else {\n                return null\n            }\n        }\n\n        result *= uradix\n\n        val beforeAdding = result\n        result += digit.toUInt()\n        if (result < beforeAdding) return null // overflow has happened\n    }\n\n    return result\n}\n","/*\n * Copyright 2010-2018 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\n@file:Suppress(\"INVISIBLE_REFERENCE\", \"INVISIBLE_MEMBER\")\npackage kotlin\n\nimport kotlin.annotation.AnnotationTarget.*\nimport kotlin.internal.RequireKotlin\nimport kotlin.internal.RequireKotlinVersionKind\n\n/**\n * Marks the API that is dependent on the experimental unsigned types, including those types themselves.\n
 *\n * Usages of such API will be reported as warnings unless an explicit opt-in with\n * the [OptIn] annotation, e.g. `@OptIn(ExperimentalUnsignedTypes::class)`,\n * or with the `-opt-in=kotlin.ExperimentalUnsignedTypes` compiler option is given.\n *\n * It's recommended to propagate the experimental status to the API that depends on unsigned types by annotating it with this annotation.\n */\n@RequiresOptIn(level = RequiresOptIn.Level.WARNING)\n@MustBeDocumented\n@Target(CLASS, ANNOTATION_CLASS, PROPERTY, FIELD, LOCAL_VARIABLE, VALUE_PARAMETER, CONSTRUCTOR, FUNCTION, PROPERTY_GETTER, PROPERTY_SETTER, TYPEALIAS)\n@Retention(AnnotationRetention.BINARY)\npublic annotation class ExperimentalUnsignedTypes\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the
```

license/LICENSE.txt file.\n
*/\n\n@file:kotlin.jvm.JvmMultifileClass\n@file:kotlin.jvm.JvmName(\"MathKt\")\n\n\npackage kotlin.math\n\n\n\n// constants, can't use them from nativeMath as they are not constants there\n\n/** Ratio of the circumference of a circle to its diameter, approximately 3.14159. */\n@SinceKotlin(\"1.2\")\npublic const val PI: Double = 3.141592653589793\n/** Base of the natural logarithms, approximately 2.71828. */\n@SinceKotlin(\"1.2\")\npublic const val E: Double = 2.718281828459045\n\n// region ================ Double Math ==\n\n/** Computes the sine of the angle [x] given in radians.\n *\n * Special cases:\n * - `sin(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun sin(x: Double): Double\n\n/** Computes the cosine of the angle [x] given in radians.\n *\n * Special cases:\n * - `cos(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun cos(x: Double): Double\n\n/** Computes the tangent of the angle [x] given in radians.\n *\n * Special cases:\n * - `tan(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun tan(x: Double): Double\n\n/**\n * Computes the arc sine of the value [x];\n * the returned value is an angle in the range from `-PI/2` to `PI/2` radians.\n *\n * Special cases:\n * - `asin(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun asin(x: Double): Double\n\n/**\n * Computes the arc cosine of the value [x];\n * the returned value is an angle in the range from `0.0` to `PI` radians.\n *\n * Special cases:\n * - `acos(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun acos(x: Double): Double\n\n/**\n * Computes the arc tangent of the value [x];\n * the returned value is an angle in the range from `-PI/2` to `PI/2` radians.\n *\n * Special cases:\n * - `atan(NaN)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun atan(x: Double): Double\n\n/**\n * Returns the angle `theta` of the polar coordinates `(r, theta)` that correspond\n * to the rectangular coordinates `(x, y)` by computing the arc tangent of the value [y] / [x];\n * the returned value is an angle in the range from `-PI` to `PI` radians.\n *\n * Special cases:\n * - `atan2(0.0, 0.0)` is `0.0`\n * - `atan2(0.0, x)` is `0.0` for `x > 0` and `PI` for `x < 0`\n * - `atan2(-0.0, x)` is `-0.0` for 'x > 0` and `-PI` for `x < 0`\n *  - `atan2(y, +Inf)` is `0.0` for `0 < y < +Inf` and `-0.0` for '-Inf < y < 0`\n * - `atan2(y, -Inf)` is `PI` for `0 < y < +Inf` and `-PI` for `-Inf < y < 0`\n * - `atan2(y, 0.0)` is `PI/2` for `y > 0` and `-PI/2` for `y < 0`\n * - `atan2(+Inf, x)` is `PI/2` for finite `x`y\n * - `atan2(-Inf, x)` is `-PI/2` for finite `x`\n * - `atan2(NaN, x)` and `atan2(y, NaN)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun atan2(y: Double, x: Double): Double\n\n/**\n * Computes the hyperbolic sine of the value [x].\n *\n * Special cases:\n * - `sinh(NaN)` is `NaN`\n * - `sinh(+Inf)` is `+Inf`\n * - `sinh(-Inf)` is `-Inf`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun sinh(x: Double): Double\n\n/**\n * Computes the hyperbolic cosine of the value [x].\n *\n * Special cases:\n * - `cosh(NaN)` is `NaN`\n * - `cosh(+Inf|-Inf)` is `+Inf`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun cosh(x: Double): Double\n\n/**\n * Computes the hyperbolic tangent of the value [x].\n *\n * Special cases:\n * - `tanh(NaN)` is `NaN`\n * - `tanh(+Inf)` is `1.0`\n * - `tanh(-Inf)` is `-1.0`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun tanh(x: Double): Double\n\n/**\n * Computes the inverse hyperbolic sine of the value [x].\n *\n * The returned value is `y` such that `sinh(y) == x`.\n *\n * Special cases:\n * - `asinh(NaN)` is `NaN`\n * - `asinh(+Inf)` is `+Inf`\n * - `asinh(-Inf)` is `-Inf`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun asinh(x: Double): Double\n\n/**\n * Computes the inverse hyperbolic cosine of the value [x].\n *\n * The returned value is positive `y` such that `cosh(y) == x`.\n *\n * Special cases:\n * - `acosh(NaN)` is `NaN`\n * - `acosh(x)` is `NaN` when `x < 1`\n * - `acosh(+Inf)` is `+Inf`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun acosh(x: Double): Double\n\n/**\n * Computes the inverse hyperbolic tangent of the value [x].\n *\n * The returned value is `y` such that `tanh(y) == x`.\n *\n * Special cases:\n * - `tanh(NaN)` is `NaN`\n * - `tanh(x)` is `NaN` when `x > 1` or `x < -1`\n * - `tanh(1.0)` is `+Inf`\n * - `tanh(-1.0)` is `-Inf`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun atanh(x: Double): Double\n\n/**\n * Computes `sqrt(x^2 + y^2)` without intermediate overflow or underflow.\n *\n * Special cases:\n * - returns `+Inf` if any of arguments is infinite\n * - returns `NaN` if any of arguments is `NaN` and the other is not infinite\n */\n@SinceKotlin(\"1.2\")\npublic expect fun hypot(x: Double, y: Double): Double\n\n/**\n * Computes the positive square root of the value [x].\n *\n * Special cases:\n * - `sqrt(x)` is `NaN` when `x

< 0` or `x` is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun sqrt(x: Double): Double\n\n/**\n * Computes Euler's number `e` raised to the power of the value [x].\n *\n * Special cases:\n *  - `exp(NaN)` is `NaN`\n *  - `exp(+Inf)` is `+Inf`\n *  - `exp(-Inf)` is `0.0`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun exp(x: Double): Double\n\n/**\n * Computes `exp(x) - 1`.\n *\n * This function can be implemented to produce more precise result for [x] near zero.\n *\n * Special cases:\n *  - `expm1(NaN)` is `NaN`\n *  - `expm1(+Inf)` is `+Inf`\n *  - `expm1(-Inf)` is `-1.0`\n *\n * @see [exp] function.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun expm1(x: Double): Double\n\n/**\n * Computes the logarithm of the value [x] to the given [base].\n *\n * Special cases:\n *  - `log(x, b)` is `NaN` if either `x` or `b` are `NaN`\n *  - `log(x, b)` is `NaN` when `x < 0` or `b <= 0` or `b == 1.0`\n *  - `log(+Inf, +Inf)` is `NaN`\n *  - `log(+Inf, b)` is `+Inf` for `b > 1` and
 `-Inf` for `b < 1`\n * - `log(0.0, b)` is `-Inf` for `b > 1` and `+Inf` for `b > 1`\n *\n * See also logarithm functions for common fixed bases: [ln], [log10] and [log2].\n */\n@SinceKotlin(\"1.2\")\npublic expect fun log(x: Double, base: Double): Double\n\n/**\n * Computes the natural logarithm (base `E`) of the value [x].\n *\n * Special cases:\n * - `ln(NaN)` is `NaN`\n * - `ln(x)` is `NaN` when `x < 0.0`\n * - `ln(+Inf)` is `+Inf`\n * - `ln(0.0)` is `-Inf`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun ln(x: Double): Double\n\n/**\n * Computes the common logarithm (base 10) of the value [x].\n *\n * @see [ln] function for special cases.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun log10(x: Double): Double\n\n/**\n * Computes the binary logarithm (base 2) of the value [x].\n *\n * @see [ln] function for special cases.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun log2(x: Double): Double\n\n/**\n * Computes `ln(x + 1)`.\n *\n * This function can be implemented to produce more precise result for [x] near zero.\n *\n * Special cases:\n * - `ln1p(NaN)` is `NaN`\n * - `ln1p(x)` is `NaN` where `x < -1.0`\n * - `ln1p(-1.0)` is `-Inf`\n * - `ln1p(+Inf)` is `+Inf`\n *\n * @see [ln] function\n * @see [expm1] function\n */\n@SinceKotlin(\"1.2\")\npublic expect fun ln1p(x: Double): Double\n\n/**\n * Rounds the given value [x] to an integer towards positive infinity.\n\n * @return the smallest double value that is greater than or equal to the given value [x] and is a mathematical integer.\n *\n * Special cases:\n * - `ceil(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun ceil(x: Double): Double\n\n/**\n * Rounds the given value [x] to an integer towards negative infinity.\n\n * @return the largest double value that is smaller than or equal to the given value [x] and is a mathematical integer.\n *\n * Special cases:\n * - `floor(x)` is `x` where `x` is `NaN` or `+Inf` or
 `-Inf` or already a mathematical integer.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun floor(x: Double): Double\n\n/**\n * Rounds the given value [x] to an integer towards zero.\n *\n * @return the value [x] having its fractional part truncated.\n *\n * Special cases:\n * - `truncate(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun truncate(x: Double): Double\n\n/**\n * Rounds the given value [x] towards the closest integer with ties rounded towards even integer.\n *\n * Special cases:\n * - `round(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun round(x: Double): Double\n\n/**\n * Returns the absolute value of the given value [x].\n *\n * Special cases:\n * - `abs(NaN)` is `NaN`\n *\n * @see absoluteValue extension property for [Double]\n */\n@SinceKotlin(\"1.2\")\npublic expect fun abs(x: Double): Double\n\n/**\n * Returns the sign of the given value [x]:\n * - `-1.0` if the value is negative,\n * - zero if the value is zero,\n * - `1.0` if the value is positive\n *\n * Special case:\n * - `sign(NaN)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun sign(x: Double): Double\n\n\n/**\n * Returns the smaller of two values.\n *\n * If either value is `NaN`, then the result is `NaN`.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun min(a: Double, b: Double): Double\n\n/**\n * Returns the greater of two values.\n *\n * If either value is `NaN`, then the result is `NaN`.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun max(a: Double, b: Double): Double\n\n\n/**\n * Returns the cube root of [x]. For any `x`, `cbrt(-x) == -cbrt(x)`;\n * that is, the cube root of a negative value is the negative of the cube root\n * of that value's magnitude. Special cases:\n *\n * Special cases:\n * - If the argument is `NaN`, then the result is `NaN`.\n * - If the argument is infinite, then the result
 is an infinity with the same sign as the argument.\n * - If the argument is zero, then the result is a zero with the same sign as the argument.\n
*/\n@SinceKotlin(\"1.8\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun cbrt(x: Double):

Double\n\n\n// extensions\n\n/**\n * Raises this value to the power [x].\n *\n * Special cases:\n * - `b.pow(0.0)` is `1.0`\n * - `b.pow(1.0) == b`\n * - `b.pow(NaN)` is `NaN`\n * - `NaN.pow(x)` is `NaN` for `x != 0.0`\n * - `b.pow(Inf)` is `NaN` for `abs(b) == 1.0`\n * - `b.pow(x)` is `NaN` for `b < 0` and `x` is finite and not an integer\n */\n@SinceKotlin(\"1.2\")\npublic expect fun Double.pow(x: Double): Double\n\n/**\n * Raises this value to the integer power [n].\n *\n * See the other overload of [pow] for details.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun Double.pow(n: Int): Double\n\n/**\n * Returns the absolute value of this value.\n *\n * Special cases:\n * - `NaN.absoluteValue` is `NaN`\n *\n * @see abs
 function\n */\n@SinceKotlin(\"1.2\")\npublic expect val Double.absoluteValue: Double\n\n/**\n * Returns the sign of this value:\n * - `-1.0` if the value is negative,\n * - zero if the value is zero,\n * - `1.0` if the value is positive\n *\n * Special case:\n * - `NaN.sign` is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect val Double.sign: Double\n\n/**\n * Returns this value with the sign bit same as of the [sign] value.\n *\n * If [sign] is `NaN` the sign of the result is undefined.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun Double.withSign(sign: Double): Double\n\n/**\n * Returns this value with the sign bit same as of the [sign] value.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun Double.withSign(sign: Int): Double\n\n/**\n * Returns the ulp (unit in the last place) of this value.\n *\n * An ulp is a positive distance between this value and the next nearest [Double] value larger in magnitude.\n *\n * Special Cases:\n * - `NaN.ulp` is `NaN`\n * - `x.ulp` is `+Inf`
 when `x` is `+Inf` or `-Inf`\n * - `0.0.ulp` is `Double.MIN_VALUE`\n */\n@SinceKotlin(\"1.2\")\npublic expect val Double.ulp: Double\n\n/**\n * Returns the [Double] value nearest to this value in direction of positive infinity.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun Double.nextUp(): Double\n\n/**\n * Returns the [Double] value nearest to this value in direction of negative infinity.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun Double.nextDown(): Double\n\n/**\n * Returns the [Double] value nearest to this value in direction from this value towards the value [to].\n *\n * Special cases:\n * - `x.nextTowards(y)` is `NaN` if either `x` or `y` are `NaN`\n * - `x.nextTowards(x) == x`\n *\n */\n@SinceKotlin(\"1.2\")\npublic expect fun Double.nextTowards(to: Double): Double\n\n/**\n * Rounds this [Double] value to the nearest integer and converts the result to [Int].\n * Ties are rounded towards positive infinity.\n *\n * Special cases:\n * - `x.roundToInt() == Int.MAX_VALUE`
 when `x > Int.MAX_VALUE`\n * - `x.roundToInt() == Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n *\n * @throws IllegalArgumentException when this value is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun Double.roundToInt(): Int\n\n/**\n * Rounds this [Double] value to the nearest integer and converts the result to [Long].\n * Ties are rounded towards positive infinity.\n *\n * Special cases:\n * - `x.roundToLong() == Long.MAX_VALUE` when `x > Long.MAX_VALUE`\n * - `x.roundToLong() == Long.MIN_VALUE` when `x < Long.MIN_VALUE`\n *\n * @throws IllegalArgumentException when this value is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun Double.roundToLong(): Long\n\n// endregion\n\n\n\n// region ================ Float Math ======================================\n\n/** Computes the sine of the angle [x] given in radians.\n *\n * Special cases:\n * - `sin(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun sin(x: Float): Float\n\n/** Computes the cosine
 of the angle [x] given in radians.\n *\n * Special cases:\n * - `cos(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun cos(x: Float): Float\n\n/** Computes the tangent of the angle [x] given in radians.\n *\n * Special cases:\n * - `tan(NaN|+Inf|-Inf)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun tan(x: Float): Float\n\n/**\n * Computes the arc sine of the value [x];\n * the returned value is an angle in the range from `-PI/2` to `PI/2` radians.\n *\n * Special cases:\n * - `asin(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun asin(x: Float): Float\n\n/**\n * Computes the arc cosine of the value [x];\n * the returned value is an angle in the range from `0.0` to `PI` radians.\n *\n * Special cases:\n * - `acos(x)` is `NaN`, when `abs(x) > 1` or x is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun acos(x: Float): Float\n\n/**\n * Computes the arc tangent of the value [x];\n * the returned value
 is an angle in the range from `-PI/2` to `PI/2` radians.\n *\n * Special cases:\n * - `atan(NaN)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun atan(x: Float): Float\n\n/**\n * Returns the angle `theta` of the polar coordinates `(r, theta)` that correspond\n * to the rectangular coordinates `(x, y)` by computing the arc tangent of the value [y] / [x];\n * the returned value is an angle in the range from `-PI` to `PI` radians.\n *\n * Special cases:\n * -

`atan2(0.0, 0.0)` is `0.0`\n * - `atan2(0.0, x)` is `0.0` for `x > 0` and `PI` for `x < 0`\n * - `atan2(-0.0, x)` is `-0.0` for 'x > 0` and `-PI` for `x < 0`\n *  - `atan2(y, +Inf)` is `0.0` for `0 < y < +Inf` and `-0.0` for '-Inf < y < 0`\n * - `atan2(y, -Inf)` is `PI` for `0 < y < +Inf` and `-PI` for `-Inf < y < 0`\n * - `atan2(y, 0.0)` is `PI/2` for `y > 0` and `-PI/2` for `y < 0`\n * - `atan2(+Inf, x)` is `PI/2` for finite `x`y\n * - `atan2(-Inf, x)` is `-PI/2` for finite `x`\n * - `atan2(NaN, x)` and
`atan2(y, NaN)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun atan2(y: Float, x: Float): Float\n\n/**\n * Computes the hyperbolic sine of the value [x].\n *\n * Special cases:\n * - `sinh(NaN)` is `NaN`\n * - `sinh(+Inf)` is `+Inf`\n * - `sinh(-Inf)` is `-Inf`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun sinh(x: Float): Float\n\n/**\n * Computes the hyperbolic cosine of the value [x].\n *\n * Special cases:\n * - `cosh(NaN)` is `NaN`\n * - `cosh(+Inf|-Inf)` is `+Inf`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun cosh(x: Float): Float\n\n/**\n * Computes the hyperbolic tangent of the value [x].\n *\n * Special cases:\n * - `tanh(NaN)` is `NaN`\n * - `tanh(+Inf)` is `1.0`\n * - `tanh(-Inf)` is `-1.0`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun tanh(x: Float): Float\n\n/**\n * Computes the inverse hyperbolic sine of the value [x].\n *\n * The returned value is `y` such that `sinh(y) == x`.\n *\n * Special cases:\n * - `asinh(NaN)` is `NaN`\n * - `asinh(+Inf)`
is `+Inf`\n * - `asinh(-Inf)` is `-Inf`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun asinh(x: Float): Float\n\n/**\n * Computes the inverse hyperbolic cosine of the value [x].\n *\n * The returned value is positive `y` such that `cosh(y) == x`.\n *\n * Special cases:\n * - `acosh(NaN)` is `NaN`\n * - `acosh(x)` is `NaN` when `x < 1`\n * - `acosh(+Inf)` is `+Inf`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun acosh(x: Float): Float\n\n/**\n * Computes the inverse hyperbolic tangent of the value [x].\n *\n * The returned value is `y` such that `tanh(y) == x`.\n *\n * Special cases:\n * - `tanh(NaN)` is `NaN`\n * - `tanh(x)` is `NaN` when `x > 1` or `x < -1`\n * - `tanh(1.0)` is `+Inf`\n * - `tanh(-1.0)` is `-Inf`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun atanh(x: Float): Float\n\n/**\n * Computes `sqrt(x^2 + y^2)` without intermediate overflow or underflow.\n *\n * Special cases:\n * - returns `+Inf` if any of arguments is infinite\n * - returns `NaN`
if any of arguments is `NaN` and the other is not infinite\n */\n@SinceKotlin(\"1.2\")\npublic expect fun hypot(x: Float, y: Float): Float\n\n/**\n * Computes the positive square root of the value [x].\n *\n * Special cases:\n * - `sqrt(x)` is `NaN` when `x < 0` or `x` is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun sqrt(x: Float): Float\n\n/**\n * Computes Euler's number `e` raised to the power of the value [x].\n *\n * Special cases:\n * - `exp(NaN)` is `NaN`\n * - `exp(+Inf)` is `+Inf`\n * - `exp(-Inf)` is `0.0`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun exp(x: Float): Float\n\n/**\n * Computes `exp(x) - 1`.\n *\n * This function can be implemented to produce more precise result for [x] near zero.\n *\n * Special cases:\n * - `expm1(NaN)` is `NaN`\n * - `expm1(+Inf)` is `+Inf`\n * - `expm1(-Inf)` is `-1.0`\n *\n * @see [exp] function.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun expm1(x: Float): Float\n\n/**\n * Computes the logarithm of the value [x] to
the given [base].\n *\n * Special cases:\n * - `log(x, b)` is `NaN` if either `x` or `b` are `NaN`\n * - `log(x, b)` is `NaN` when `x < 0` or `b <= 0` or `b == 1.0`\n * - `log(+Inf, +Inf)` is `NaN`\n * - `log(+Inf, b)` is `+Inf` for `b > 1` and `-Inf` for `b < 1`\n * - `log(0.0, b)` is `-Inf` for `b > 1` and `+Inf` for `b > 1`\n *\n * See also logarithm functions for common fixed bases: [ln], [log10] and [log2].\n */\n@SinceKotlin(\"1.2\")\npublic expect fun log(x: Float, base: Float): Float\n\n/**\n * Computes the natural logarithm (base `E`) of the value [x].\n *\n * Special cases:\n * - `ln(NaN)` is `NaN`\n * - `ln(x)` is `NaN` when `x < 0.0`\n * - `ln(+Inf)` is `+Inf`\n * - `ln(0.0)` is `-Inf`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun ln(x: Float): Float\n\n/**\n * Computes the common logarithm (base 10) of the value [x].\n *\n * @see [ln] function for special cases.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun log10(x: Float): Float\n\n/**\n * Computes the
binary logarithm (base 2) of the value [x].\n *\n * @see [ln] function for special cases.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun log2(x: Float): Float\n\n/**\n * Computes `ln(x + 1)`.\n *\n * This function can be implemented to produce more precise result for [x] near zero.\n *\n * Special cases:\n * - `ln1p(NaN)` is `NaN`\n * - `ln1p(x)` is `NaN` where `x < -1.0`\n * - `ln1p(-1.0)` is `-Inf`\n * - `ln1p(+Inf)` is `+Inf`\n *\n * @see [ln] function\n * @see [expm1] function\n */\n@SinceKotlin(\"1.2\")\npublic expect fun ln1p(x: Float): Float\n\n/**\n * Rounds the given value [x] to an integer towards positive infinity.\n\n * @return the

smallest Float value that is greater than or equal to the given value [x] and is a mathematical integer.\n *\n * Special cases:\n * - `ceil(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun ceil(x: Float): Float\n\n/**\n * Rounds the given value [x] to an integer towards negative infinity.\n\n * @return the largest Float value that is smaller than or equal to the given value [x] and is a mathematical integer.\n *\n * Special cases:\n * - `floor(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun floor(x: Float): Float\n\n/**\n * Rounds the given value [x] to an integer towards zero.\n *\n * @return the value [x] having its fractional part truncated.\n *\n * Special cases:\n * - `truncate(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun truncate(x: Float): Float\n\n/**\n * Rounds the given value [x] towards the closest integer with ties rounded towards even integer.\n *\n * Special cases:\n * - `round(x)` is `x` where `x` is `NaN` or `+Inf` or `-Inf` or already a mathematical integer.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun round(x: Float): Float\n\n\n/**\n * Returns the absolute value of the given value [x].\n *\n * Special cases:\n * - `abs(NaN)` is `NaN`\n *\n * @see absoluteValue extension property for [Float]\n */\n@SinceKotlin(\"1.2\")\npublic expect fun abs(x: Float): Float\n\n/**\n * Returns the sign of the given value [x]:\n * - `-1.0` if the value is negative,\n * - zero if the value is zero,\n * - `1.0` if the value is positive\n *\n * Special case:\n * - `sign(NaN)` is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun sign(x: Float): Float\n\n\n/**\n * Returns the smaller of two values.\n *\n * If either value is `NaN`, then the result is `NaN`.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun min(a: Float, b: Float): Float\n\n/**\n * Returns the greater of two values.\n *\n * If either value is `NaN`, then the result is `NaN`.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun max(a: Float, b: Float): Float\n\n\n/**\n * Returns the cube root of [x]. For any `x`, `cbrt(-x) == -cbrt(x)`;\n * that is, the cube root of a negative value is the negative of the cube root\n * of that value's magnitude. Special cases:\n *\n * Special cases:\n * - If the argument is `NaN`, then the result is `NaN`.\n * - If the argument is infinite, then the result is an infinity with the same sign as the argument.\n * - If the argument is zero, then the result is a zero with the same sign as the argument.\n */\n@SinceKotlin(\"1.8\")\n@WasExperimental(ExperimentalStdlibApi::class)\npublic expect fun cbrt(x: Float): Float\n\n\n// extensions\n\n\n/**\n * Raises this value to the power [x].\n *\n * Special cases:\n * - `b.pow(0.0)` is `1.0`\n * - `b.pow(1.0) == b`\n * - `b.pow(NaN)` is `NaN`\n * - `NaN.pow(x)` is `NaN` for `x != 0.0`\n * - `b.pow(Inf)` is `NaN` for `abs(b) == 1.0`\n * - `b.pow(x)` is `NaN` for `b < 0` and `x` is finite and not an integer\n */\n@SinceKotlin(\"1.2\")\npublic expect fun Float.pow(x: Float): Float\n\n/**\n * Raises this value to the integer power [n].\n *\n * See the other overload of [pow] for details.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun Float.pow(n: Int): Float\n\n/**\n * Returns the absolute value of this value.\n *\n * Special cases:\n * - `NaN.absoluteValue` is `NaN`\n *\n * @see abs function\n */\n@SinceKotlin(\"1.2\")\npublic expect val Float.absoluteValue: Float\n\n/**\n * Returns the sign of this value:\n * - `-1.0` if the value is negative,\n * - zero if the value is zero,\n * - `1.0` if the value is positive\n *\n * Special case:\n * - `NaN.sign` is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect val Float.sign: Float\n\n/**\n * Returns this value with the sign bit same as of the [sign] value.\n *\n * If [sign] is `NaN` the sign of the result is undefined.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun Float.withSign(sign: Float): Float\n\n/**\n * Returns this value with the sign bit same as of the [sign] value.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun Float.withSign(sign: Int): Float\n\n\n/**\n * Rounds this [Float] value to the nearest integer and converts the result to [Int].\n * Ties are rounded towards positive infinity.\n *\n * Special cases:\n * - `x.roundToInt() == Int.MAX_VALUE` when `x > Int.MAX_VALUE`\n * - `x.roundToInt() == Int.MIN_VALUE` when `x < Int.MIN_VALUE`\n *\n * @throws IllegalArgumentException when this value is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun Float.roundToInt(): Int\n\n/**\n * Rounds this [Float] value to the nearest integer and converts the result to [Long].\n * Ties are rounded towards positive infinity.\n *\n * Special cases:\n * - `x.roundToLong() == Long.MAX_VALUE` when `x > Long.MAX_VALUE`\n * - `x.roundToLong() == Long.MIN_VALUE` when `x < Long.MIN_VALUE`\n *\n * @throws IllegalArgumentException when this value is `NaN`\n */\n@SinceKotlin(\"1.2\")\npublic expect fun Float.roundToLong(): Long\n\n\n// endregion\n\n// region =============== Integer Math

==\n\n\n/**\n * Returns the absolute value of the given value [n].\n *\n * Special cases:\n *   - `abs(Int.MIN_VALUE)` is `Int.MIN_VALUE` due to an overflow\n *\n * @see absoluteValue extension property for [Int]\n */\n@SinceKotlin(\"1.2\")\npublic expect fun abs(n: Int): Int\n\n/**\n * Returns the smaller of two values.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun min(a: Int, b: Int): Int\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun max(a: Int, b: Int): Int\n\n/**\n * Returns the absolute value of this value.\n *\n * Special cases:\n *   - `Int.MIN_VALUE.absoluteValue` is `Int.MIN_VALUE` due to an overflow\n *\n * @see abs function\n */\n@SinceKotlin(\"1.2\")\npublic expect val Int.absoluteValue: Int\n\n/**\n * Returns the sign of this value:\n *   - `-1` if the value is negative,\n *   - `0` if the value is zero,\n *   - `1` if the value is positive\n */\n@SinceKotlin(\"1.2\")\npublic expect val Int.sign: Int\n\n\n\n/**\n * Returns the absolute value of the given value [n].\n *\n * Special cases:\n *   - `abs(Long.MIN_VALUE)` is `Long.MIN_VALUE` due to an overflow\n *\n * @see absoluteValue extension property for [Long]\n */\n@SinceKotlin(\"1.2\")\npublic expect fun abs(n: Long): Long\n\n/**\n * Returns the smaller of two values.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun min(a: Long, b: Long): Long\n\n/**\n * Returns the greater of two values.\n */\n@SinceKotlin(\"1.2\")\npublic expect fun max(a: Long, b: Long): Long\n\n/**\n * Returns the absolute value of this value.\n *\n * Special cases:\n *   - `Long.MIN_VALUE.absoluteValue` is `Long.MIN_VALUE` due to an overflow\n *\n * @see abs function\n */\n@SinceKotlin(\"1.2\")\npublic expect val Long.absoluteValue: Long\n\n/**\n * Returns the sign of this value:\n *   - `-1` if the value is negative,\n *   - `0` if the value is zero,\n *   - `1` if the value is positive\n */\n@SinceKotlin(\"1.2\")\npublic expect val Long.sign: Int\n\n\n// endregion\n","/*\n * Copyright 2010-2022 JetBrains s.r.o. and Kotlin Programming Language contributors.\n * Use of this source code is governed by the Apache 2.0 license that can be found in the license/LICENSE.txt file.\n */\n\npackage kotlin.js\n\n/**\n * Exposes the JavaScript [Math object](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Math) to Kotlin.\n */\n@PublishedApi\n@JsName(\"Math\")\ninternal external object JsMath {\n   val LN2: Double\n   fun abs(value: Double): Double\n   fun acos(value: Double): Double\n   fun asin(value: Double): Double\n   fun atan(value: Double): Double\n   fun atan2(y: Double, x: Double): Double\n   fun cos(value: Double): Double\n   fun sin(value: Double): Double\n   fun exp(value: Double): Double\n   fun max(vararg values: Int): Int\n   fun max(vararg values: Float): Float\n   fun max(vararg values: Double): Double\n   fun min(vararg values: Int): Int\n   fun min(vararg values: Float): Float\n   fun min(vararg values: Double): Double\n   fun sqrt(value: Double): Double\n   fun tan(value: Double): Double\n   fun log(value: Double): Double\n   fun cbrt(value: Double): Double\n   fun pow(base: Double, exp: Double): Double\n   fun round(value: Number): Double\n   fun floor(value: Number): Double\n   fun ceil(value: Number): Double\n}\n\ninternal const val defineTaylorNBound = \"\"\"\n   var epsilon = 2.220446049250313E-16;\n   var taylor_2_bound = Math.sqrt(epsilon);\n   var taylor_n_bound = Math.sqrt(taylor_2_bound);\n\"\"\"\n\ninternal const val defineUpperTaylor2Bound = \"\"\"\n $defineTaylorNBound\n   var upper_taylor_2_bound = 1/taylor_2_bound;\n\"\"\"\n\ninternal const val defineUpperTaylorNBound = \"\"\"\n $defineUpperTaylor2Bound\n   var upper_taylor_n_bound = 1/taylor_n_bound;\n\"\"\"\n"],"names":[],"mappings":"AAWC,CAXA,yB;EACG,IAAI,OAAO,MAAO,KAAI,UAAW,IAAG,MAAM,IAA1C,C;IACI,MAAM,CAAC,QAAD,EAAW,CAAC,SAAD,CAAX,EAAwB,OAAxB,C;SAEL,IAAI,OAAO,OAAQ,KAAI,QAAvB,C;IACD,OAAO,CAAC,MAAM,QAAP,C;;IAGP,IAAI,OAAQ,GAAE,E;IACd,OAAO,CAAC,IAAI,OAAL,C;;CAEd,CAAC,IAAD,EAAO,kB;EACJ,IAAI,IAAI,M;ECPU;;;IAAtB,MAAM,eAAgB,GAAE,a;IACpB,OAAoD,CAA5C,KAAK,QAAQ,CAAC,CAAD,CAAI,IAAG,CAAE,YAAW,SAAW,KAAG,CAAC,OAAQ,KAAI,c;G;EAGxE,MAAM,YAAa,GAAE,a;IACjB,OAO,CAAE,YAAW,SAAU,IAAG,CAAC,OAAQ,KAAI,c;G;EAGlD,MAAM,aAAc,GAAE,a;IAClB,OAO,CAAE,YAAW,U;G;EAGxB,MAAM,YAAa,GAAE,a;IACjB,OAAO,CAAE,YAAW,WAAY,IAAG,CAAC,OAAQ,KAAI,W;G;EAGpD,MAAM,WAAY,GAAE,a;IAChB,OAAO,CAAE,YAAW,U;G;EAGxB,MAAM,aAAc,GAAE,a;IAClB,OAAO,CAAE,YAAW,Y;G;EAGxB,MAAM,cAAe,GAAE,a;IACnB,OAAO,CAAE,YAAW,Y;G;EAGxB,MAAM,YAAa,GAAE,a;IACjB,OAAO,KAAK,QAAQ,CAAC,CAAD,CAAI,IAAG,CAAC,OAAQ,KAAI,W;G;EAG5C,MAAM,QAAS,GAAE,a;IACb,OAAO,KAAK,QA

AQ,CAAC,CAAD,CAAI,IAAG,CAAC,CAAC,O;G;EAGjC,MAAM,WAAY,GAAE,a;IAChB,OAAO,KAAK,QAA
Q,CAAC,CAAD,CAAI,IAAG,WAAW,OAAO,CAAC,CAAD,C;G;EAGjD,MAAM,cAAe,GAAE,a;IACnB,IAAI,C
AAE,KAAI,IAAV,C;MAAgB,OAAO,M;IACvB,IAAI,WAAW,MAAM,YAAY,CAAC,CAAD,CAAI,GAAE,MAA
M,aAAR,GAAwB,MAAM,S;IACnE,OAAO,GAAI,GAAE,KAAK,UAAU,IAAI,KAAK,CAAC,CAAD,EAAI,a;M
AAc,OAAO,QAAQ,CAAC,CAAD,C;KAAjC,CAAwC,KAAK,CAAC,IAAD,CAAO,GAAE,G;G;EAG/F,MAAM,k
BAAmB,GAAE,e;IACvB,OAAO,MAAM,OAAO,YAAY,wBAAwB,CAAC,GAAD,C;G;EAG5D,MAAM,YAAa,G
AAE,gB;IACjB,IAAI,CAAE,KAAI,CAAV,C;MACI,OAAO,I;;IAEX,IAAI,CAAE,KAAI,IAAK,IAAG,CAAE,KA
AI,IAAK,IAAG,CAAC,MAAM,WAAW,CAAC,CAAD,CAAI,IAAG,CAAC,OAAQ,KAAI,CAAC,OAAvE,C;MA
CI,OAAO,K;;IAGX,KAAK,IAAI,IAAI,CAAR,EAAW,IAAI,CAAC,OAArB,EAA8B,CAAE,GAAE,CAAlC,EAAq
C,CAAC,EAAtC,C;MACI,IAAI,CAAC,MAAM,OAAO,CAAC,CAAC,CAAD,CAAF,EAAO,CAAC,CAA
C,CAAD,CAAR,CAAlB,C;QACI,OAAO,K;;;IAGf,OAAO,I;G;EAGX,MAAM,gBAAiB,GAAE,gB;IACrB,OAAO,
MAAM,OAAO,YAAY,sBAAsB,CAAC,CAAD,EAAI,CAAJ,C;G;EAG1D,MAAM,cAAe,GAAE,e;IACnB,IAAI,G
AAI,KAAI,IAAZ,C;MAAkB,OAAO,C;IACzB,IAAI,SAAS,C;IACb,KAAK,IAAI,IAAI,CAAR,EAAW,IAAI,GAA
G,OAAvB,EAAgC,CAAE,GAAE,CAApC,EAAuC,CAAC,EAAxC,C;MACI,MAAO,GAAAqB,CAAjB,EAAG,GAA
E,MAAO,GAAE,CAAG,IAAE,MAAM,SAAS,CAAC,GAAG,CAAC,CAAD,CAAJ,CAAU,GAAE,C;;IAE7D,OA
AO,M;G;EAGX,MAAM,kBAAmB,GAAE,e;IACvB,OAAO,MAAM,OAAO,YAAY,wBAAwB,CAAC,GAAD,C;G
;EAG5D,MAAM,mBAAoB,GAAE,iB;IACxB,KAAK,KAAK,CAAC,MAAM,gBAAP,C;G;ECpFQ;;;IAAtB,MAA
M,eAAgB,GAAE,mB;IACpB,CAAC,aAAc,GAAE,I;IACjB,OAAO,C;G;EAGX,MAAM,uBAAwB,GAAE,4C;IAC
5B,MAAM,IAAK,GAAE,M;IACb,MAAM,IAAK,GAAE,M;IACb,MAAM,aAAc,GAAE,I;IACtB,OAAO,mBAAm
B,CAAC,MAAD,EAAS,MAAT,EAAAiB,6BAA6B,CAAC,UAAD,CAA9C,C;G;EAG9B,iD;IACI,GAAG,WAAY,G
AAE,sBAAsB,CAAC,OAAO,MAAO,KAAI,UAAW,GAAE,KAAK,QAAP,GAAAkB,KAAK,UAArD,C;IACvC,GA
AG,YAAa,GAAE,G;IAClB,OAAO,G;G;EAGX,IAAI,gCAAgC,CAChC,CACI,OADJ,EACa,CAAE,KAAF,EAAS,
IAAT,EAAe,oBAAf,EAAqC,Y;IAC1C,OAAO,MAAM,OAAO,QAAQ,kB;GADvB,CADb,EAII,SAJJ,EAIe,CAAE
,KAAF,EAAS,IAAT,EAAe,oBAAf,EAAqC,Y;IAC5C,OAAO,MAAM,OAAO,QAAQ,W;GADrB,CAJf,CADgC,E
AShC,CACI,OADJ,EACa,CAAE,KAAF,EAAS,IAAT,EAAe,oBAAf,EAAqC,Y;IAC1C,OAAO,MAAM,OAAO,Q
AAQ,kB;GADvB,CADb,EAII,SAJJ,EAIe,CAAE,KAAF,EAAS,IAAT,EAAe,oBAAf,EAAqC,Y;IAC5C,OAAO,M
AAM,OAAO,QAAQ,W;GADrB,CAJf,CATgC,C;EAmBpC,uC;IACI,IAAI,KAAK,MAAO,KAAI,IAApB,C;MACI,
KAAK,MAAO,GAAE,CACV,UADU,EACE,CAAC,KAAK,qBAAqB,EAA3B,CADF,EAEV,SAFU,EAEC,IAFD,
EAGV,SAHU,EAGC,EAHD,EAIV,UAJU,EAIE,EAJF,EAKV,KALU,EAKH,EALG,EAMV,aANU,EAMK,EANL,
C;;IASlB,OAAO,KAAK,M;G;EChDD;;;IAAf,MAAM,QAAS,GAAE,a;IACb,OAAoB,CAAZ,CAAE,GAAE,KAA
Q,KAAG,EAAG,IAAG,E;G;EAGjC,MAAM,OAAQ,GAAE,a;IACZ,OAAkB,CAAV,CAAE,GAAE,GAAM,KAA
G,EAAG,IAAG,E;G;EAG/B,MAAM,OAAQ,GAAE,a;IACZ,OAAO,CAAE,GAAE,K;G;EAGf,MAAM,aAAc,GA
AE,a;IAClB,OAAO,CAAE,YAAW,MAAM,KAAM,GAAE,CAAF,GAAM,MAAM,KAAK,WAAW,CAAC,CAAD
,C;G;EAGhE,MAAM,YAAa,GAAE,a;IACjB,OAAO,CAAE,YAAW,MAAM,KAAM,GAAE,CAAC,MAAM,EAA
T,GAAc,MAAM,YAAY,CAAC,CAAD,C;G;EAGpE,MAAM,cAAe,GAAE,a;IACnB,OAAO,MAAM,QAAQ,CAA
C,MAAM,YAAY,CAAC,CAAD,CAAnB,C;G;EAGzB,MAAM,aAAc,GAAE,a;IAClB,OAAO,MAAM,OAAO,CA
AC,MAAM,YAAY,CAAC,CAAD,CAAnB,C;G;EAGxB,MAAM,eAAgB,GAAE,a;IACpB,OAAO,CAAC,C;G;EA
GZ,MAAM,aAAc,GAAE,a;IAClB,OAAO,MAAM,OAAO,CAAC,MAAM,YAAY,CAAC,CAAD,CAAnB,C;G;EA
GxB,MAAM,YAAa,GAAE,a;IACjB,IAAI,CAAE,GAAE,UAAR,C;MAAoB,OAAO,U;IAC3B,IAAI,CAAE,GAAE
,WAAR,C;MAAqB,OAAO,W;IAC5B,OAAO,CAAE,GAAE,C;G;EAGf,MAAM,YAAa,GAAE,a;IACjB,IAAI,CA
AE,IAAG,IAAT,C;MAAe,OAAO,C;IACtB,IAAI,CAAE,YAAW,MAAM,UAAvB,C;MAAmC,OAAO,C;IAC1C,O
AAO,IAAI,MAAM,UAAV,CAAqB,CAArB,C;G;EAGX,MAAM,UAAW,GAAE,a;IACf,IAAI,CAAE,IAAG,IAAT
,C;MAAe,OAAO,C;IACtB,OAAO,MAAM,OAAO,CAAC,CAAD,C;G;EClDV;;;IAAd,MAAM,OAAQ,GAAE,sB;I
ACZ,IAAI,IAAK,IAAG,IAAZ,C;MACI,OAAO,IAAK,IAAG,I;;IAGnB,IAAI,IAAK,IAAG,IAAZ,C;MACI,OAAO,
K;;IAGX,IAAI,IAAK,KAAI,IAAb,C;MACI,OAAO,IAAK,KAAI,I;;IAGpB,IAAI,OAAO,IAAK,KAAI,QAAS,IAA
G,OAAO,IAAI,OAAQ,KAAI,UAAvD,C;MACI,OAAO,IAAI,OAAO,CAAC,IAAD,C;;IAGtB,IAAI,OAAO,IAAK,
KAAI,QAAS,IAAG,OAAO,IAAK,KAAI,QAAhD,C;MACI,OAAO,IAAK,KAAI,IAAK,KAAI,IAAK,KAAI,CAA
E,IAAG,CAAE,GAAE,IAAK,KAAI,CAAE,GAAE,IAAnC,C;;IAGzB,OAAO,IAAK,KAAI,I;G;EAGpB,MAAM,S

AAU,GAAE,e;IACd,IAAI,GAAI,IAAG,IAAX,C;MACI,OAAO,C;;IAEX,IAAI,UAAU,OAAO,G;IACrB,IAAI,QA
AS,KAAI,OAAjB,C;MACI,OAAO,UAAW,KAAI,OAAO,GAAG,SAAU,GAAE,GAAG,SAAS,EAAd,GAAmB,iB
AAiB,CAAC,GAAD,C;;IAElF,IAAI,UAAW,KAAI,OAAnB,C;MACI,OAAO,iBAAiB,CAAC,GAAD,C;;IAE5B,I
AAI,QAAS,KAAI,OAAjB,C;MACI,OAAO,MAAM,eAAe,CAAC,GAAD,C;;IAEhC,IAAI,SAAU,KAAI,OAAlB,C
;MACI,OAAO,MAAM,CAAC,GAAD,C;;IAGjB,IAAI,MAAM,MAAM,CAAC,GAAD,C;IAChB,OAAO,iBAAiB,
CAAC,GAAD,C;G;EAI5B,MAAM,SAAU,GAAE,a;IACd,IAAI,CAAE,IAAG,IAAT,C;MACI,OAAO,M;WAEN,I
AAI,MAAM,WAAW,CAAC,CAAD,CAArB,C;MACD,OAAO,O;;MAGP,OAAO,CAAC,SAAS,E;;GART,Y;EAah
B,IAAI,WAAW,aAAf;cAGA,IAAI,iCAAiC,sB;EAErC,gC;IACI,IAAI,EAAE,8BAA+B,IAAG,GAApC,CAAJ,C;M
ACI,IAAI,OAAQ,IAAI,OAAO,EAAG,GAAE,QAAU,GAAE,CAAxC;A,MACA,MAAM,eAAe,CAAC,GAAD,EA
AM,8BAAN,EAAsC,CAAE,KAAF,EAAU,IAAV,EAAgB,UAAhB,EAA4B,KAA5B,CAAtC,C;;IAEzB,OAAO,GA
AG,CAAC,8BAAD,C;G;EAGd,gC;IACI,IAAI,OAAO,C;IACX,KAAK,IAAI,IAAI,CAAb,EAAgB,CAAE,GAAE,
GAAG,OAAvB,EAAgC,CAAC,EAAjC,C;MACI,IAAI,OAAQ,GAAG,WAAW,CAAC,CAAD,C;MAC1B,IAAM,
GAAG,IAAK,GAAE,EAAG,GAAE,IAAM,GAAE,CAAvB;A;IAEV,OAAO,I;G;EAGX,MAAM,iBAAkB,GAAE,i
B;EC9Cd;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;IAAZ,MAAM,KAAM,GAAE,qB;IAKF;;;MAAV,IAAI,KAAM,GAAE,GAAI,G
AAE,CAAR;;;;MAMV,IAAI,MAAO,GAAE,IAAK,GAAE,CAAT;A,G;EAGb,MAAM,KAAK,WAAY,GAAE,CA
CrB,IADqB,EACf,OADe,EAErB,UAFqB,EAET,MAFS,EAGrB,UAHqB,EAGV,EAHU,CAAF;;;;;;IAgBvB,MAA
M,KAAK,UAAW,GAAE,EAAF;;;;I;EAQtB,MAAM,KAAK,QAAS,GAAE,iB;IACpB,IAAI,IAAK,IAAG,KAAM,I
AAG,KAAM,GAAE,GAA7B,C;MACE,IAAI,YAAY,MAAM,KAAK,UAAU,CAAC,KAAD,C;MACrC,IAAI,SAA
J,C;QACE,OAAO,S;;;IAIX,IAAI,MAAM,IAAI,MAAM,KAAV,CAAgB,KAAM,GAAE,CAAxB,EAA2B,KAAM,
GAAE,CAAE,GAAE,EAAF,GAAO,CAA5C,C;IACV,IAAI,IAAK,IAAG,KAAM,IAAG,KAAM,GAAE,GAA7B,C
;MACE,MAAM,KAAK,UAAU,CAAC,KAAD,CAAQ,GAAE,G;;IAEjC,OAAO,G;GAZW;;;;;;I;EAwBpB,MAAM
,KAAK,WAAY,GAAE,iB;IACvB,IAAI,KAAK,CAAC,KAAD,CAAT,C;MACE,OAAO,MAAM,KAAK,K;WACb,
IAAI,KAAM,IAAG,CAAC,MAAM,KAAK,gBAAzB,C;MACL,OAAO,MAAM,KAAK,U;WACb,IAAI,KAAM,G
AAE,CAAE,IAAG,MAAM,KAAK,gBAA5B,C;MACL,OAAO,MAAM,KAAK,U;WACb,IAAI,KAAM,GAAE,CA
AZ,C;MACL,OAAO,MAAM,KAAK,WAAW,CAAC,CAAC,KAAF,CAAQ,OAAO,E;;MAE5C,OAAO,IAAI,MA
AM,KAAV,CACF,KAAM,GAAE,MAAM,KAAK,gBAAkB,GAAE,CADrC,EAEF,KAAM,GAAE,MAAM,KAAK
,gBAAkB,GAAE,CAFrC,C,C;;GAVY;;;;;;I;EAwBvB,MAAM,KAAK,SAAU,GAAE,6B;IACrB,OAAO,IAAI,MAAM
,KAAV,CAAgB,OAAhB,EAAyB,QAAzB,C;GADY;;;;;;I;EAYrB,MAAM,KAAK,WAAY,GAAE,0B;IACvB,IAAI
,GAAG,OAAQ,IAAG,CAAlB,C;MACE,MAAM,KAAK,CAAC,mCAAD,C;;IAGb,IAAI,QAAQ,SAAU,IAAG,E;I
ACzB,IAAI,KAAM,GAAE,CAAE,IAAG,EAAG,GAAE,KAAtB,C;MACE,MAAM,KAAK,CAAC,sBAAuB,GAA
E,KAA1B,C;;IAGb,IAAI,GAAG,OAAO,CAAC,CAAD,CAAI,IAAG,GAArB,C;MACE,OAAO,MAAM,KAAK,W
AAW,CAAC,GAAG,UAAU,CAAC,CAAD,CAAd,EAAmB,KAAnB,CAAyB,OAAO,E;WACxD,IAAI,GAAG,QA
AQ,CAAC,GAAD,CAAM,IAAG,CAAxB,C;MACL,MAAM,KAAK,CAAC,+CAAgD,GAAE,GAAnD,C;;;;IAKb,I
AAI,eAAe,MAAM,KAAK,WAAW,CAAC,IAAI,IAAI,CAAC,KAAD,EAAQ,CAAR,CAAT,C;IAEzC,IAAI,SAAS
,MAAM,KAAK,K;IACxB,KAAK,IAAI,IAAI,CAAb,EAAgB,CAAE,GAAE,GAAG,OAAvB,EAAgC,CAAE,IAA
G,CAArC,C;MACE,IAAI,OAAO,IAAI,IAAI,CAAC,CAAD,EAAI,GAAG,OAAQ,GAAE,CAAjB,C;MACnB,IAAI
,QAAQ,QAAQ,CAAC,GAAG,UAAU,CAAC,CAAD,EAAI,CAAE,GAAE,IAAR,CAAd,EAA6B,KAA7B,C;MAC
pB,IAAI,IAAK,GAAE,CAAX,C;QACE,IAAI,QAAQ,MAAM,KAAK,WAAW,CAAC,IAAI,IAAI,CAAC,KAAD,E
AAQ,IAAR,CAAT,C;QAClC,MAAO,GAAE,MAAM,SAAS,CAAC,KAAD,CAAO,IAAI,CAAC,MAAM,KAAK,
WAAW,CAAC,KAAD,CAAvB,C;;QAEnC,MAAO,GAAE,MAAM,SAAS,CAAC,YAAD,C;QACxB,MAAO,GAA
E,MAAM,IAAI,CAAC,MAAM,KAAK,WAAW,CAAC,KAAD,CAAvB,C;;;IAGvB,OAAO,M;GAhCc;;;;;;IA8Cv
B,MAAM,KAAK,gBAAiB,GAAE,CAAE,IAAG,EAAP;;;I;EAO5B,MAAM,KAAK,gBAAiB,GAAE,CAAE,IAAG,
EAAP;;;I;EAO5B,MAAM,KAAK,gBAAiB,GACxB,MAAM,KAAK,gBAAiB,GAAE,MAAM,KAAK,gBADjB;;;I;
EAQ5B,MAAM,KAAK,gBAAiB,GACxB,MAAM,KAAK,gBAAiB,GAAE,CADN;;;I;EAQ5B,MAAM,KAAK,gB
AAiB,GACxB,MAAM,KAAK,gBAAiB,GAAE,MAAM,KAAK,gBADjB;;;I;EAQ5B,MAAM,KAAK,gBAAiB,GA
CxB,MAAM,KAAK,gBAAiB,GAAE,MAAM,KAAK,gBADjB;;;I;EAQ5B,MAAM,KAAK,gBAAiB,GACxB,MA
AM,KAAK,gBAAiB,GAAE,CADN,0B;EAK5B,MAAM,KAAK,KAAM,GAAE,MAAM,KAAK,QAAQ,CAAC,C
AAD,CAArB,0B;EAIjB,MAAM,KAAK,IAAK,GAAE,MAAM,KAAK,QAAQ,CAAC,CAAD,CAArB,0B;EAIhB,

MAAM,KAAK,QAAS,GAAE,MAAM,KAAK,QAAQ,CAAC,EAAD,CAArB,0B;EAIpB,MAAM,KAAK,UAAW,
GAClB,MAAM,KAAK,SAAS,CAAC,aAAW,GAAE,CAAd,EAAiB,UAAW,GAAE,CAA9B,CADF,0B;EAKtB,M
AAM,KAAK,UAAW,GAAE,MAAM,KAAK,SAAS,CAAC,CAAD,EAAI,aAAW,GAAE,CAAjB,CAAtB;;;I;EAOt
B,MAAM,KAAK,YAAa,GAAE,MAAM,KAAK,QAAQ,CAAC,CAAE,IAAG,EAAN,CAArB,kE;EAIxB,MAAM,
KAAK,UAAU,MAAO,GAAE,Y;IAC5B,OAAO,IAAI,K;GADe,+E;EAM5B,MAAM,KAAK,UAAU,SAAU,GAAE
,Y;IAC/B,OAAO,IAAI,MAAO,GAAE,MAAM,KAAK,gBAAiB,GACzC,IAAI,mBAAmB,E;GAFD,yD;EAM/B,M
AAM,KAAK,UAAU,SAAU,GAAE,Y;IAC/B,OAAO,IAAI,MAAO,GAAE,IAAI,K;GADK;;;;I;EAS/B,MAM,KA
AK,UAAU,SAAU,GAAE,qB;IAC/B,IAAI,QAAQ,SAAU,IAAG,E;IACzB,IAAI,KAAM,GAAE,CAAE,IAAG,EA
AG,GAAE,KAAtB,C;MACE,MAAM,KAAK,CAAC,sBAAuB,GAAE,KAA1B,C;;IAGb,IAAI,IAAI,OAAO,EAAf,
C;MACE,OAAO,G;;IAGT,IAAI,IAAI,WAAW,EAAnB,C;MACE,IAAI,IAAI,WAAW,CAAC,MAAM,KAAK,UA
AZ,CAAnB,C;QAGE;;YAAI,YAAY,MAAM,KAAK,WAAW,CAAC,KAAD,C;QACtC,IAAI,MAAM,IAAI,IAAI,
CAAC,SAAD,C;QAClB,IAAI,MAAM,GAAG,SAAS,CAAC,SAAD,CAAW,SAAS,CAAC,IAAD,C;QAC1C,OAA
O,GAAG,SAAS,CAAC,KAAD,CAAQ,GAAE,GAAG,MAAM,EAAE,SAAS,CAAC,KAAD,C;;QAEjD,OAAO,G
AAI,GAAE,IAAI,OAAO,EAAE,SAAS,CAAC,KAAD,C;;;;;IAMvC,IAAI,eAAe,MAAM,KAAK,WAAW,CAAC,I
AAI,IAAI,CAAC,KAAD,EAAQ,CAAR,CAAT,C;IAEzC,IAAI,MAAM,I;IACV,IAAI,SAAS,E;IACb,OAAO,IAAP
,C;MACE,IAAI,SAAS,GAAG,IAAI,CAAC,YAAD,C;MACpB,IAAI,SAAS,GAAG,SAAS,CAAC,MAAM,SAAS,
CAAC,YAAD,CAAhB,CAA+B,MAAM,E;MAC9D,IAAI,SAAS,MAAM,SAAS,CAAC,KAAD,C;MAE5B,GAAI,
GAAE,M;MACN,IAAI,GAAG,OAAO,EAAd,C;QACE,OAAO,MAAO,GAAE,M;;QAEhB,OAAO,MAAM,OAAQ
,GAAE,CAAvB,C;UACE,MAAO,GAAE,GAAI,GAAE,M;;QAEjB,MAAO,GAAE,EAAG,GAAE,MAAO,GAAE,
M;;;GAzCE,0D;EAgD/B,MAAM,KAAK,UAAU,YAAa,GAAE,Y;IAClC,OAAO,IAAI,M;GADqB,yD;EAMlC,MA
AM,KAAK,UAAU,WAAY,GAAE,Y;IACjC,OAAO,IAAI,K;GADoB,4D;EAMjC,MAAM,KAAK,UAAU,mBAAo
B,GAAE,Y;IACzC,OAAQ,IAAI,KAAM,IAAG,CAAG,GACpB,IAAI,KADgB,GACR,MAAM,KAAK,gBAAiB,G
AAE,IAAI,K;GAFX;;;I;EAUzC,MAAM,KAAK,UAAU,cAAe,GAAE,Y;IACpC,IAAI,IAAI,WAAW,EAAnB,C;M
ACE,IAAI,IAAI,WAAW,CAAC,MAAM,KAAK,UAAZ,CAAnB,C;QACE,OAAO,E;;QAEP,OAAO,IAAI,OAAO,
EAAE,cAAc,E;;;MAGpC,IAAI,MAAM,IAAI,MAAO,IAAG,CAAE,GAAE,IAAI,MAAN,GAAe,IAAI,K;MAC7C,
KAAK,IAAI,MAAM,EAAAf,EAAmB,GAAI,GAAE,CAAzB,EAA4B,GAAG,EAA/B,C;QACE,IAAuB,CAAlB,GA
AI,GAAG,CAAE,IAAG,GAAM,KAAG,CAA1B,C;UACE,K;;;MAGJ,OAAO,IAAI,MAAO,IAAG,CAAE,GAAE,
GAAI,GAAE,EAAR,GAAa,GAAI,GAAE,C;;GAdV,mD;EAoBpC,MAAM,KAAK,UAAU,OAAQ,GAAE,Y;IAC7
B,OAAO,IAAI,MAAO,IAAG,CAAE,IAAG,IAAI,KAAM,IAAG,C;GADZ,uD;EAM7B,MAAM,KAAK,UAAU,W
AAY,GAAE,Y;IACjC,OAAO,IAAI,MAAO,GAAE,C;GADW,kD;EAMjC,MAAM,KAAK,UAAU,MAAO,GAAE,
Y;IAC5B,OAAuB,CAAf,IAAI,KAAM,GAAE,CAAG,KAAG,C;GADA;;;I;EAS5B,MAAM,KAAK,UAAU,WAAY
,GAAE,iB;IACjC,OAAQ,IAAI,MAAO,IAAG,KAAK,MAAQ,IAAI,IAAI,KAAM,IAAG,KAAK,K;GAD1B;;;I;EA
SjC,MAAM,KAAK,UAAU,cAAe,GAAE,iB;IACpC,OAAQ,IAAI,MAAO,IAAG,KAAK,MAAQ,IAAI,IAAI,KAA
M,IAAG,KAAK,K;GADvB;;;I;EASpC,MAAM,KAAK,UAAU,SAAU,GAAE,iB;IAC/B,OAAO,IAAI,QAAQ,CAA
C,KAAD,CAAQ,GAAE,C;GADA;;;I;EAS/B,MAAM,KAAK,UAAU,gBAAiB,GAAE,iB;IACtC,OAAO,IAAI,QA
AQ,CAAC,KAAD,CAAQ,IAAG,C;GADM;;;I;EAStC,MAAM,KAAK,UAAU,YAAa,GAAE,iB;IAClC,OAAO,IA
AI,QAAQ,CAAC,KAAD,CAAQ,GAAE,C;GADG;;;I;EASlC,MAAM,KAAK,UAAU,mBAAoB,GAAE,iB;IACzC,
OAAO,IAAI,QAAQ,CAAC,KAAD,CAAQ,IAAG,C;GADS;;;;I;EAWzC,MAAM,KAAK,UAAU,QAAS,GAAE,iB
;IAC9B,IAAI,IAAI,WAAW,CAAC,KAAD,CAAnB,C;MACE,OAAO,C;;IAGT,IAAI,UAAU,IAAI,WAAW,E;IAC
7B,IAAI,WAAW,KAAK,WAAW,E;IAC/B,IAAI,OAAQ,IAAG,CAAC,QAAhB,C;MACE,OAAO,E;;IAET,IAAI,C
AAC,OAAQ,IAAG,QAAhB,C;MACE,OAAO,C;;;IAIT,IAAI,IAAI,SAAS,CAAC,KAAD,CAAO,WAAW,EAAnC,
C;MACE,OAAO,E;;MAEP,OAAO,C;;GAlBmB,wD;EAwB9B9B,MAAM,KAAK,UAAU,OAAQ,GAAE,Y;IAC7B,I
AAI,IAAI,WAAW,CAAC,MAAM,KAAK,UAAZ,CAAnB,C;MACE,OAAO,MAAM,KAAK,U;;MAElB,OAAO,I
AAI,IAAI,EAAE,IAAI,CAAC,MAAM,KAAK,IAAZ,C;;GAJ;;;;I;EAc7B,MAAM,KAAK,UAAU,IAAK,GAAE,iB
;IAG1B;QAAI,MAAM,IAAI,MAAO,KAAI,E;IACzB,IAAI,MAAM,IAAI,MAAO,GAAE,K;IACvB,IAAI,MAAM,
IAAI,KAAM,KAAI,E;IACxB,IAAI,MAAM,IAAI,KAAM,GAAE,K;IAEtB,IAAI,MAAM,KAAK,MAAO,KAAI,E;
IAC1B,IAAI,MAAM,KAAK,MAAO,GAAE,K;IACxB,IAAI,MAAM,KAAK,KAAM,KAAI,E;IACzB,IAAI,MAA
M,KAAK,KAAM,GAAE,K;IAEvB,IAAI,MAAM,CAAV,EAAa,MAAM,CAAnB,EAAsB,MAAM,CAA5B,EAA+

B,MAAM,C;IACrC,GAAI,IAAG,GAAI,GAAE,G;IACb,GAAI,IAAG,GAAI,KAAI,E;IACf,GAAI,IAAG,K;IACP,
GAAI,IAAG,GAAI,GAAE,G;IACb,GAAI,IAAG,GAAI,KAAI,E;IACf,GAAI,IAAG,K;IACP,GAAI,IAAG,GAAI,
GAAE,G;IACb,GAAI,IAAG,GAAI,KAAI,E;IACf,GAAI,IAAG,K;IACP,GAAI,IAAG,GAAI,GAAE,G;IACb,GAA
I,IAAG,K;IACP,OAAO,MAAM,KAAK,SAAS,CAAE,GAAI,IAAG,EAAI,GAAE,GAAf,EAAqB,GAAI,IAAG,EA
AI,GAAE,GAAlC,C;GAzBH;;;;I;EAkC1B,MAAM,KAAK,UAAU,SAAU,GAAE,iB;IAC/B,OAAO,IAAI,IAAI,CA
AC,KAAK,OAAO,EAAb,C;GADc;;;;I;EAU/B,MAAM,KAAK,UAAU,SAAU,GAAE,iB;IAC/B,IAAI,IAAI,OAAO
,EAAf,C;MACE,OAAO,MAAM,KAAK,K;WACb,IAAI,KAAK,OAAO,EAAhB,C;MACL,OAAO,MAAM,KAAK,
K;;IAGpB,IAAI,IAAI,WAAW,CAAC,MAAM,KAAK,UAAZ,CAAnB,C;MACE,OAAO,KAAK,MAAM,EAAG,G
AAE,MAAM,KAAK,UAAb,GAA0B,MAAM,KAAK,K;WACrD,IAAI,KAAK,WAAW,CAAC,MAAM,KAAK,U
AAZ,CAApB,C;MACL,OAAO,IAAI,MAAM,EAAG,GAAE,MAAM,KAAK,UAAb,GAA0B,MAAM,KAAK,K;;I
AG3D,IAAI,IAAI,WAAW,EAAnB,C;MACE,IAAI,KAAK,WAAW,EAApB,C;QACE,OAO,IAAI,OAAO,EAAE,
SAAS,CAAC,KAAK,OAAO,EAAb,C;;QAE7B,OAO,IAAI,OAAO,EAAE,SAAS,CAAC,KAAD,CAAO,OAAO,
E;;WAExC,IAAI,KAAK,WAAW,EAApB,C;MACL,OAAO,IAAI,SAAS,CAAC,KAAK,OAAO,EAAb,CAAgB,O
AAO,E;;;IAI7C,IAAI,IAAI,SAAS,CAAC,MAAM,KAAK,YAAZ,CAA0B,IACvC,KAAK,SAAS,CAAC,MAAM,K
AAK,YAAZ,CADlB,C;MAEE,OAAO,MAAM,KAAK,WAAW,CAAC,IAAI,SAAS,EAAG,GAAE,KAAK,SAAS,
EAAjC,C;;;;IAM/B,IAAI,MAAM,IAAI,MAAO,KAAI,E;IACzB,IAAI,MAAM,IAAI,MAAO,GAAE,K;IACvB,IA
AI,MAAM,IAAI,KAAM,KAAI,E;IACxB,IAAI,MAAM,IAAI,KAAM,GAAE,K;IAEtB,IAAI,MAAM,KAAK,MA
AO,KAAI,E;IAC1B,IAAI,MAAM,KAAK,MAAO,GAAE,K;IACxB,IAAI,MAAM,KAAK,KAAM,KAAI,E;IACzB
,IAAI,MAAM,KAAK,KAAM,GAAE,K;IAEvB,IAAI,MAAM,CAAV,EAAa,MAAM,CAAnB,EAAsB,MAAM,CA
A5B,EAA+B,MAAM,C;IACrC,GAAI,IAAG,GAAI,GAAE,G;IACb,GAAI,IAAG,GAAI,KAAI,E;IACf,GAAI,IAA
G,K;IACP,GAAI,IAG,GAAI,GAAE,G;IACb,GAAI,IAG,GAAI,KAAI,E;IACf,GAAI,IAAG,K;IACP,GAAI,IA
AG,GAAI,GAAE,G;IACb,GAAI,IAAG,GAAI,KAAI,E;IACf,GAAI,IAAG,K;IACP,GAAI,IAAG,GAAI,GAAE,G;I
ACb,GAAI,IAAG,GAAI,KAAI,E;IACf,GAAI,IAAG,K;IACP,GAAI,IAAG,GAAI,GAAE,G;IACb,GAAI,IAAG,G
AAI,KAAI,E;IACf,GAAI,IAAG,K;IACP,GAAI,IAAG,GAAI,GAAE,G;IACb,GAAI,IAAG,GAAI,KAAI,E;IACf,G
AAI,IAAG,K;IACP,GAAI,IAAG,GAAI,GAAE,GAAI,GAAE,GAAI,GAAE,GAAI,GAAE,GAAI,GAAE,GAAI,G
AAE,GAAI,GAAE,G;IACjD,GAAI,IAAG,K;IACP,OAAO,MAAM,KAAK,SAAS,CAAE,GAAI,IAAG,EAAI,GA
AE,GAAf,EAAqB,GAAI,IAAG,EAAI,GAAE,GAAlC,C;GA/DE;;;;I;EAwE/B,MAAM,KAAK,UAAU,IAAK,GAA
E,iB;IAC1B,IAAI,KAAK,OAAO,EAAhB,C;MACE,MAAM,KAAK,CAAC,kBAAD,C;WACN,IAAI,IAAI,OAAO,
EAAf,C;MACL,OAAO,MAAM,KAAK,K;;IAGpB,IAAI,IAAI,WAAW,CAAC,MAAM,KAAK,UAAZ,CAAnB,C;
MACE,IAAI,KAAK,WAAW,CAAC,MAAM,KAAK,IAAZ,CAAkB,IAClC,KAAK,WAAW,CAAC,MAAM,KAA
K,QAAZ,CADpB,C;QAEE,OAAO,MAAM,KAAK,UAAlB;A,aACK,IAAI,KAAK,WAAW,CAAC,MAAM,KAAK
,UAAZ,CAApB,C;QACL,OAAO,MAAM,KAAK,I;;QAGlB;YAAI,WAAW,IAAI,WAAW,CAAC,CAAD,C;QAC9
B,IAAI,SAAS,QAAQ,IAAI,CAAC,KAAD,CAAO,UAAU,CAAC,CAAD,C;QAC1C,IAAI,MAAM,WAAW,CAAC
,MAAM,KAAK,KAAZ,CAArB,C;UACE,OAAO,KAAK,WAAW,EAAG,GAAE,MAAM,KAAK,IAAb,GAAoB,M
AAM,KAAK,Q;;UAEzD,IAAI,MAAM,IAAI,SAAS,CAAC,KAAK,SAAS,CAAC,MAAD,CAAf,C;UACvB,IAAI,
SAAS,MAAM,IAAI,CAAC,GAAG,IAAI,CAAC,KAAD,CAAR,C;UACvB,OAAO,M;;;WAGN,IAAI,KAAK,WA
AW,CAAC,MAAM,KAAK,UAAZ,CAApB,C;MACL,OAAO,MAAM,KAAK,K;;IAGpB,IAAI,IAAI,WAAW,EAA
nB,C;MACE,IAAI,KAAK,WAAW,EAApB,C;QACE,OAAO,IAAI,OAAO,EAAE,IAAI,CAAC,KAAK,OAAO,EA
Ab,C;;QAExB,OAAO,IAAI,OAAO,EAAE,IAAI,CAAC,KAAD,CAAO,OAAO,E;;WAEnC,IAAI,KAAK,WAAW,
EAApB,C;MACL,OAAO,IAAI,IAAI,CAAC,KAAK,OAAO,EAAb,CAAgB,OAAO,E;;;;;;;IAQxC,IAAI,MAAM,M
AAM,KAAK,K;IACrB,IAAI,MAAM,I;IACV,OAAO,GAAG,mBAAmB,CAAC,KAAD,CAA7B,C;MAGE;;UAAI,
SAAS,IAAI,IAAI,CAAC,CAAD,EAAI,IAAI,MAAM,CAAC,GAAG,SAAS,EAAG,GAAE,KAAK,SAAS,EAAhC,
CAAd,CAArB;;A,MAIA,IAAI,OAAO,IAAI,KAAK,CAAC,IAAI,IAAI,CAAC,MAAD,CAAS,GAAE,IAAI,IAAxB
,C;MACpB,IAAI,QAAS,IAAK,IAAG,EAAI,GAAE,CAAF,GAAM,IAAI,IAAI,CAAC,CAAD,EAAI,IAAK,GAAE
,EAAX,CAAvC;;A,MAIA,IAAI,YAAY,MAAM,KAAK,WAAW,CAAC,MAAD,C;MACtC,IAAI,YAAY,SAAS,S
AAS,CAAC,KAAD,C;MAClC,OAAO,SAAS,WAAW,EAAG,IAAG,SAAS,YAAY,CAAC,GAAD,CAAtD,C;QAC
E,MAAO,IAAG,K;QACV,SAAU,GAAE,MAAM,KAAK,WAAW,CAAC,MAAD,C;QAClC,SAAU,GAAE,SAAS,
SAAS,CAAC,KAAD,C;;;;MAKhC,IAAI,SAAS,OAAO,EAApB,C;QACE,SAAU,GAAE,MAAM,KAAK,I;;MAGz

B,GAAI,GAAE,GAAG,IAAI,CAAC,SAAD,C;MACb,GAAI,GAAE,GAAG,SAAS,CAAC,SAAD,C;;IAEpB,OAA
O,G;GA3EiB;;;;I;EAoF1B,MAAM,KAAK,UAAU,OAAQ,GAAE,iB;IAC7B,OAAO,IAAI,SAAS,CAAC,IAAI,IAA
I,CAAC,KAAD,CAAO,SAAS,CAAC,KAAD,CAAzB,C;GADO,2D;EAM7B,MAAM,KAAK,UAAU,IAAK,GAAE
,Y;IAC1B,OAAO,MAAM,KAAK,SAAS,CAAC,CAAC,IAAI,KAAN,EAAa,CAAC,IAAI,MAAlB,C;GADH;;;;I;E
AU1B,MAAM,KAAK,UAAU,IAAK,GAAE,iB;IAC1B,OAAO,MAAM,KAAK,SAAS,CAAC,IAAI,KAAM,GAAE
,KAAK,KAAlB,EACI,IAAI,MAAO,GAAE,KAAK,MADtB,C;GADH;;;;I;EAW1B,MAAM,KAAK,UAAU,GAAI,
GAAE,iB;IACzB,OAAO,MAAM,KAAK,SAAS,CAAC,IAAI,KAAM,GAAE,KAAK,KAAlB,EACI,IAAI,MAAO,
GAAE,KAAK,MADtB,C;GADJ;;;;I;EAWzB,MAAM,KAAK,UAAU,IAAK,GAAE,iB;IAC1B,OAAO,MAAM,KA
AK,SAAS,CAAC,IAAI,KAAM,GAAE,KAAK,KAAlB,EACI,IAAI,MAAO,GAAE,KAAK,MADtB,C;GADH;;;;I;
EAW1B,MAAM,KAAK,UAAU,UAAW,GAAE,mB;IAChC,OAAQ,IAAG,E;IACX,IAAI,OAAQ,IAAG,CAAf,C;
MACE,OAAO,I;;MAEP,IAAI,MAAM,IAAI,K;MACd,IAAI,OAAQ,GAAE,EAAd,C;QACE,IAAI,OAAO,IAAI,M;
QACf,OAAO,MAAM,KAAK,SAAS,CACvB,GAAI,IAAG,OADgB,EAEtB,IAAK,IAAG,OAAS,GAAG,GAAI,KA
AK,EAAG,GAAE,OAFZ,C;;QAI3B,OAAO,MAAM,KAAK,SAAS,CAAC,CAAD,EAAI,GAAI,IAAI,OAAQ,GAA
E,EAAtB,C;;;GAZD;;;;I;EAuBhC,MAAM,KAAK,UAAU,WAAY,GAAE,mB;IACjC,OAAQ,IAAG,E;IACX,IAAI,
OAAQ,IAAG,CAAf,C;MACE,OAAO,I;;MAEP,IAAI,OAAO,IAAI,M;MACf,IAAI,OAAQ,GAAE,EAAd,C;QACE
,IAAI,MAAM,IAAI,K;QACd,OAAO,MAAM,KAAK,SAAS,CACtB,GAAI,KAAI,OAAS,GAAG,IAAK,IAAI,EA
AG,GAAE,OADZ,EAEvB,IAAK,IAAG,OAFe,C;;QAI3B,OAAO,MAAM,KAAK,SAAS,CACvB,IAAK,IAAI,OA
AQ,GAAE,EADI,EAEvB,IAAK,IAAG,CAAE,GAAE,CAAF,GAAM,EAFO,C;;;GAZA;;;;;I;EA2BjC,MAAM,KA
AK,UAAU,mBAAoB,GAAE,mB;IACzC,OAAQ,IAAG,E;IACX,IAAI,OAAQ,IAAG,CAAf,C;MACE,OAAO,I;;M
AEP,IAAI,OAAO,IAAI,M;MACf,IAAI,OAAQ,GAAE,EAAd,C;QACE,IAAI,MAAM,IAAI,K;QACd,OAAO,MAA
M,KAAK,SAAS,CACtB,GAAI,KAAI,OAAS,GAAG,IAAK,IAAI,EAAG,GAAE,OADZ,EAEvB,IAAK,KAAI,OA
Fc,C;aAGtB,IAAI,OAAQ,IAAG,EAAf,C;QACL,OAAO,MAAM,KAAK,SAAS,CAAC,IAAD,EAAO,CAAP,C;;Q
AE3B,OAAO,MAAM,KAAK,SAAS,CAAC,IAAK,KAAK,OAAQ,GAAE,EAArB,EAA0B,CAA1B,C;;;GAdQ;A,E
AoBzC,MAAM,KAAK,UAAU,OAAQ,GAAE,iB;IAC3B,OAAO,KAAM,YAAW,MAAM,KAAM,IAAG,IAAI,W
AAW,CAAC,KAAD,C;G;EAG1D,MAAM,KAAK,UAAU,gBAAiB,GAAE,MAAM,KAAK,UAAU,Q;EAE7D,MA
AM,KAAK,UAAU,IAAK,GAAE,Y;IACxB,OAAO,IAAI,IAAI,CAAC,MAAM,KAAK,IAAZ,C;G;EAGnB,MAA
M,KAAK,UAAU,IAAK,GAAE,Y;IACxB,OAAO,IAAI,IAAI,CAAC,MAAM,KAAK,QAAZ,C;G;EAGnB,MAAM,
KAAK,UAAU,QAAS,GAAE,Y;IAC5B,OAAO,IAAI,SAAS,E;G;EAGxB,MAAM,KAAK,UAAU,UAAW,GAAE,
Y;IAC9B,OAAO,I;G;EAGX,MAAM,KAAK,UAAU,WAAY,GAAE,MAAM,KAAK,UAAU,O;EACxD,MAAM,K
AAK,UAAU,IAAK,GAAE,MAAM,KAAK,UAAU,I;EAEjD,MAAM,KAAK,UAAU,QAAS,GAAE,iB;IAC5B,OA
AO,IAAI,MAAM,OAAO,OAAO,UAAxB,CAAmC,IAAnC,EAAyC,KAAzC,C;G;EC1zBS;;;;;IAApB,MAAM,aA
Ac,GAAE,2B;G;EAGtB,MAAM,qBAAsB,GAAE,oB;IAC1B,OAAO,G;G;EAGX,MAAM,aAAc,GAAE,e;IAClB,I
AAI,IAAI,Y;MACJ,CAAE,GAAE,GAAG,E;MACP,OAAO,CAAC,MAAM,CAAC,IAAD,EAAO,SAAP,C;K;IAEl
B,OAAO,Y;MACH,OAAO,CAAC,MAAM,CAAC,IAAD,EAAO,SAAP,C;K;G;EAItB,MAAM,SAAU,GAAE,gB;I
ACd,OAAO,kB;MACH,OAAO,OAAO,MAAO,KAAI,I;K;G;EAIjC,MAAM,aAAc,GAAE,iB;IAClB,OAAO,kB;M
ACH,OAAO,MAAM,OAAO,CAAC,MAAD,EAAS,KAAT,C;K;G;EAI5B,MAAM,OAAQ,GAAE,c;IACZ,OAAO,
kB;MACH,OAAO,MAAO,IAAG,IAAK,IAAG,EAAE,CAAC,MAAD,C;K;G;EAInC,MAAM,aAAc,GAAE,gB;IA
ClB,OAAO,kB;MACH,OAAO,CAAC,CAAC,MAAD,CAAS,IAAG,CAAC,CAAC,MAAD,C;K;G;EAI7B,MAAM,
qBAAsB,GAAE,wC;G;EAG9B,MAAM,YAAa,GAAE,iB;IACjB,OAAO,K;G;EAGX,MAAM,gBAAiB,GAAE,qB;
IACrB,gBAAgB,E;G;EAGpB,MAAM,oBAAqB,GAAE,qB;IACzB,gBAAgB,E;G;EAGpB,MAAM,kBAAmB,GAA
E,qB;IACvB,gBAAgB,E;G;EAGpB,MAAM,mBAAoB,GAAE,4B;IACxB,gBAAgB,E;G;EAGpB,MAAM,6BAA8
B,GAAE,yB;IAClC,gBAAgB,E;G;EAGpB,4B;IACI,MAAM,IAAI,KAAJ,CACF,iDAAkD,GAClD,qDAAsD,GACt
D,uDAHE,C;G;EAMV,MAAM,gBAAiB,GAAE,4B;IACrB,OAAO,Y;MACH,OAAO,Y;K;G;ECjFE;;;IAAjB,MAA
M,UAAW,GAAE,gB;IACf,IAAI,QAAQ,OAAO,C;IACnB,IAAI,KAAM,KAAI,QAAd,C;MACI,IAAI,OAAO,CAA
E,KAAI,QAAjB,C;QACI,OAAO,MAAM,gBAAgB,CAAC,CAAD,EAAI,CAAJ,C;;MAEjC,OAAO,MAAM,mBA
AmB,CAAC,CAAD,EAAI,CAAJ,C;;IAEpC,IAAI,KAAM,KAAI,QAAS,IAAG,KAAM,KAAI,SAApC,C;MACI,O
AAO,MAAM,mBAAmB,CAAC,CAAD,EAAI,CAAJ,C;;IAEpC,OAAO,CAAC,gBAAgB,CAAC,CAAD,C;G;EAG
5B,MAAM,mBAAoB,GAAE,gB;IACxB,OAAO,CAAE,GAAE,CAAE,GAAE,EAAF,GAAO,CAAE,GAAE,CAAE

,GAAE,CAAF,GAAM,C;G;EAGpC,MAAM,gBAAiB,GAAE,gB;IACrB,IAAI,CAAE,GAAE,CAAR,C;MAAW,O
AAO,E;IAClB,IAAI,CAAE,GAAE,CAAR,C;MAAW,OAAO,C;IAElB,IAAI,CAAE,KAAI,CAAV,C;MACI,IAAI,
CAAE,KAAI,CAAV,C;QAAa,OAAO,C;MAEpB,IAAI,KAAK,CAAE,GAAE,C;MACb,OAAO,EAAG,KAAI,CA
AE,GAAE,CAAE,GAAE,CAAF,GAAO,EAAG,GAAE,CAAE,GAAE,EAAF,GAAO,C;;IAG7C,OAAO,CAAE,KA
AI,CAAE,GAAG,CAAE,KAAI,CAAE,GAAE,CAAF,GAAM,CAAjB,GAAsB,E;G;EAGzC,MAAM,QAAS,GAAE
,iB;IACb,OAAO,MAAM,OAAO,CAAC,KAAK,GAAC,CAAP,C;G;EAGxB,MAAM,QAAS,GAAE,iB;IACb,OAA
O,MAAM,OAAO,CAAC,KAAK,GAAC,CAAP,C;G;EAGxB,MAAM,KAAM,GAAE,IAAI,KAAM,IAAG,I;EAE3
B,MAAM,aAAc,GAAE,I;EAEtB,oB;IACI,OAAyB,CAAhB,CAAE,GAAE,YAAY,KAAG,CAAE,GAAE,KAAP,C
AAe,GAAe,CAAZ,CAAE,GAAE,KAAQ,KAAG,CAAE,GAAE,CAAP,CAAW,GAAE,C;G;EA6DtE,CA1DD,Y;IA
CG,IAAI,MAAM,IAAI,WAAJ,CAAgB,CAAhB,C;IACV,IAAI,aAAa,IAAI,YAAJ,CAAiB,GAAjB,C;IACjB,IAAI,
aAAa,IAAI,YAAJ,CAAiB,GAAjB,C;IACjB,IAAI,WAAW,IAAI,UAAJ,CAAe,GAAf,C;IACf,IAAI,WAAW,C;IAC
f,IAAI,YAAY,C;IAEhB,UAAU,CAAC,CAAD,CAAI,GAAE,EAAF;A,IACd,IAAI,QAAQ,CAAC,QAAD,CAAW,
KAAI,CAA3B,C;MACI,QAAS,GAAE,C;MACX,SAAU,GAAE,C;;IAGhB,MAAM,aAAc,GAAE,iB;MAClB,OAA
O,MAAM,gBAAgB,CAAC,KAAK,CAAC,KAAD,CAAQ,GAAE,GAAF,GAAQ,KAAtB,C;K;IAGjC,MAAM,gBA
AiB,GAAE,iB;MACrB,UAAU,CAAC,CAAD,CAAI,GAAE,K;MAChB,OAAO,MAAM,KAAK,SAAS,CAAC,QA
AQ,CAAC,QAAD,CAAT,EAAqB,QAAQ,CAAC,SAAD,CAA7B,C;K;IAG/B,MAAM,eAAgB,GAAE,iB;MACpB,
QAAQ,CAAC,QAAD,CAAW,GAAE,KAAK,K;MAC1B,QAAQ,CAAC,SAAD,CAAY,GAAE,KAAK,M;MAC3B,
OAAO,UAAU,CAAC,CAAD,C;K;IAGrB,MAAM,YAAa,GAAE,iB;MACjB,OAAO,MAAM,eAAe,CAAC,KAAK,
CAAC,KAAD,CAAQ,GAAE,GAAF,GAAQ,KAAtB,C;K;IAGhC,MAAM,eAAgB,GAAE,iB;MACpB,UAAU,CA
AC,CAAD,CAAI,GAAE,K;MAChB,OAAO,QAAQ,CAAC,CAAD,C;K;IAGnB,MAAM,cAAe,GAAE,iB;MACnB,
QAAQ,CAAC,CAAD,CAAI,GAAE,K;MACd,OAAO,UAAU,CAAC,CAAD,C;KAFA;A,IAMrB,MAAM,cAAe,G
AAE,iB;MACnB,UAAU,CAAC,CAAD,CAAI,GAAE,K;MAChB,OAAO,QAAQ,CAAC,SAAD,CAAY,GAAE,a;K
;IAGjC,MAAM,eAAgB,GAAE,e;MACpB,IAAc,CAAT,GAAI,GAAE,CAAG,MAAI,GAAlB,C;QACI,OAAO,GA
AI,GAAE,C;;QAGb,UAAU,CAAC,CAAD,CAAI,GAAE,G;QAChB,OAAsC,CAA9B,QAAQ,CAAC,SAAD,CAA
Y,GAAE,EAAG,GAAE,CAAG,IAAE,QAAQ,CAAC,QAAD,CAAW,GAAE,C;;K;GAGvE,G;EAEF,MAAM,cAAe
,GAAE,a;IACnB,OAAO,CAAE,IAAG,IAAK,GAAE,CAAF,GAAM,MAAM,SAAS,E;G;EC7G1C;;;QAAI,OAAO,
MAAM,UAAU,WAAY,KAAI,WAA3C,C;IACI,MAAM,eAAe,CAAC,MAAM,UAAP,EAAmB,YAAnB,EAAiC,C
AClD,KADkD,EAC3C,kC;MACH,QAAS,GAAE,QAAS,IAAG,C;MACvB,OAAO,IAAI,YAAY,CAAC,YAAD,E
AAe,QAAf,CAAyB,KAAI,Q;KAHN,CAAjC,C;;EAOzB,IAAI,OAAO,MAAM,UAAU,SAAU,KAAI,WAAzC,C;I
ACI,MAAM,eAAe,CAAC,MAAM,UAAP,EAAmB,UAAnB,EAA+B,CAChD,KADgD,EACzC,kC;MACH,IAAI,g
BAAgB,IAAI,SAAS,E;MACjC,IAAI,QAAS,KAAI,SAAU,IAAG,QAAS,GAAE,aAAa,OAAtD,C;QACI,QAAS,G
AAE,aAAa,O;;MAE5B,QAAS,IAAG,YAAY,O;MACxB,IAAI,YAAY,aAAa,QAAQ,CAAC,YAAD,EAAe,QAAf,
C;MACrC,OAAO,SAAU,KAAI,EAAG,IAAG,SAAU,KAAI,Q;KARG,CAA/B,C;;;EAazB,IAAI,OAAO,IAAI,KA
AM,KAAI,WAAzB,C;IACI,IAAI,KAAM,GAAE,a;MACR,CAAE,GAAE,CAAC,CAAH;A,MACF,IAAI,CAAE,K
AAI,CAAE,IAAG,KAAK,CAAC,CAAD,CAApB,C;QACI,OAAO,MAAM,CAAC,CAAD,C;;MAEjB,OAAO,CAA
E,GAAE,CAAE,GAAE,CAAF,GAAM,E;K;;EAG3B,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;IACI,IAAI,MA
AO,GAAE,a;MACT,IAAI,KAAK,CAAC,CAAD,CAAT,C;QACI,OAAO,G;;MAEX,IAAI,CAAE,GAAE,CAAR,C;
QACI,OAAO,IAAI,MAAM,CAAC,CAAD,C;;MAErB,OAAO,IAAI,KAAK,CAAC,CAAD,C;K;;EAuKtB,CAnKD,
Y;IACG,IAAI,UAAU,qB;IACd,IAAI,iBAAiB,IAAI,KAAK,CAAC,OAAD,C;IAC9B,IAAI,iBAAiB,IAAI,KAAK,
CAAC,cAAD,C;IAC9B,IAAI,uBAAuB,CAAC,GAAC,c;IAC7B,IAAI,uBAAuB,CAAC,GAAC,c;IAE7B,IAAI,OA
AO,IAAI,KAAM,KAAI,WAAzB,C;MACI,IAAI,KAAM,GAAE,a;QACR,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,G
AAE,cAAlB,C;UACI,IAAI,SAAS,C;UACb,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAlB,C;YACI,MAAO
,IAAI,CAAE,GAAE,CAAE,GAAE,CAAG,GAAE,C;;UAE5B,OAAO,M;;UAEP,IAAI,IAAI,IAAI,IAAI,CAAC,CA
AD,C;UAChB,IAAI,KAAK,CAAE,GAAE,C;UACb,IAAI,CAAC,QAAQ,CAAC,CAAD,CAAb,C;YAAkB,OAAO,
IAAI,IAAI,CAAC,CAAE,GAAE,IAAI,IAAT,C;UACjC,IAAI,CAAC,QAAQ,CAAC,EAAD,CAAb,C;YAAmB,OA
AO,CAAC,IAAI,IAAI,CAAC,CAAC,CAAE,GAAE,IAAI,IAAV,C;UACnC,OAAgB,CAAR,CAAE,GAAE,EAAI,I
AAE,C;;O;;IAI9B,IAAI,OAAO,IAAI,KAAM,KAAI,WAAzB,C;MACI,IAAI,KAAM,GAAE,a;QACR,IAAI,IAAI,I
AAI,IAAI,CAAC,CAAD,C;QAChB,IAAI,KAAK,CAAE,GAAE,C;QACb,IAAI,CAAC,QAAQ,CAAC,CAAD,CAAD,CA

AI,IAAG,CAAC,QAAQ,CAAC,EAAD,CAA7B,C;UAAmC,OAAO,IAAI,IAAI,CAAC,IAAI,IAAI,CAAC,CAAD,
CAAI,GAAE,IAAI,IAAnB,C;QAClD,OAAgB,CAAR,CAAE,GAAE,EAAI,IAAE,C;O;;IAI1B,IAAI,OAAO,IAAI,
KAAM,KAAI,WAAzB,C;MACI,IAAI,KAAM,GAAE,a;QACR,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cA
AlB,C;UACI,IAAI,SAAS,C;UACb,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAlB,C;YACI,MAAO,IAAI,C
AAE,GAAE,CAAE,GAAE,CAAG,GAAE,C;;UAE5B,OAAO,M;;UAGP,IAAI,IAAI,IAAI,IAAI,CAAC,CAAC,CA
AF,CAAhB,EAAsB,IAAI,IAAI,IAAI,CAAC,CAAC,CAAF,C;UAClC,OAAO,CAAE,KAAI,QAAS,GAAE,CAAF,
GAAM,CAAE,KAAI,QAAS,GAAE,EAAF,GAAe,CAAP,CAAE,GAAE,CAAG,KAAG,CAAE,GAAE,CAAP,C;;O
;;;;IAQtE,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,QAAQ,a;QACR,IAAI,CAAE,IAAG,CAAC,c
AAV,C;UAEI,IAAI,CAAE,GAAE,oBAAR,C;YAEI,IAAI,CAAE,GAAE,oBAAR,C;cAGI;qBAAO,IAAI,IAAI,CA
AC,CAAD,CAAI,GAAE,IAAI,I;;cAKzB;qBAAO,IAAI,IAAI,CAAC,CAAE,GAAE,CAAE,GAAG,CAAE,IAAG,C
AAE,GAAE,CAAP,CAAZ,C;;;YAKnB,OAAO,IAAI,IAAI,CAAC,CAAE,GAAE,IAAI,KAAK,CAAC,CAAE,GA
AE,CAAE,GAAE,CAAT,CAAd,C;;eAGlB,IAAI,CAAE,IAAG,CAAC,cAAV,C;UAED,OAAO,CAAC,KAAK,CA
AC,CAAC,CAAF,C;;UAKb;cAAI,SAAS,C;UACb,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,IAAG,cAAnB,C;YAEI,I
AAI,KAAK,CAAE,GAAE,CAAE,GAAE,CAAjB;A,YAEA,MAAO,IAAG,EAAG,GAAE,C;;UAEnB,OAAO,M;;O
;MAGf,IAAI,MAAO,GAAE,K;;IAEjB,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,MAAO,GAAE,a
;QACT,IAAI,CAAE,GAAE,CAAR,C;UAEI,OAAO,G;eAEN,IAAI,CAAE,GAAE,CAAE,IAAG,cAAb,C;UAED,I
AAI,CAAE,GAAE,oBAAR,C;YAGI;mBAAO,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,IAAI,I;;YAIzB,OAAO,IA
AI,IAAI,CAAC,CAAE,GAAE,IAAI,KAAK,CAAC,CAAE,GAAE,CAAE,GAAE,CAAT,CAAd,C;;;UAKnB,IAAI,
IAAI,IAAI,KAAK,CAAC,CAAE,GAAE,CAAL,CAAjB;A,UAEA,IAAI,SAAS,C;UACb,IAAI,CAAE,IAAG,cAAT
,C;YAEI,IAAI,KAAK,CAAE,GAAE,CAAE,GAAE,CAAjB;A,YAEA,MAAO,IAAG,EAAG,GAAE,E;;UAGnB,O
AAO,IAAI,KAAK,CAAC,CAAD,CAAI,GAAE,M;;O;;IAIlC,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,
IAAI,MAAO,GAAE,a;QACT,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAlB,C;UACI,IAAI,SAAS,C;UACb
,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAlB,C;YACI,MAAO,IAAI,CAAE,GAAE,CAAE,GAAE,CAAG,
GAAE,C;;UAE5B,OAAO,M;;QAEX,OAAO,IAAI,IAAI,CAAS,CAAP,CAAE,GAAE,CAAG,KAAG,CAAE,GAA
E,CAAP,CAAT,CAAoB,GAAE,C;O;;IAG7C,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,MAAO,G
AAE,a;QACT,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAlB,C;UACI,IAAI,KAAK,CAAE,GAAE,C;UACb
,IAAI,KAAK,EAAG,GAAE,C;UACd,IAAI,KAAK,EAAG,GAAE,CAAd;A,UAEA,OAAQ,CAAC,EAAG,GAAE,
CAAE,GAAE,EAAG,GAAE,CAAE,GAAE,EAAG,GAAE,CAAE,GAAE,C;;QAExC,OAAO,IAAI,IAAI,CAAC,C
AAE,GAAE,CAAL,C;O;;IAGvB,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;MACI,IAAI,MAAO,GAAE,a;QAC
T,IAAI,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,cAAlB,C;UACI,IAAI,KAAK,CAAE,GAAE,C;UACb,IAAI,KAA
K,EAAG,GAAE,C;UACd,IAAI,KAAK,EAAG,GAAE,CAAd;A,UAEA,OAAQ,EAAG,GAAE,EAAG,GAAE,EAA
G,GAAE,CAAE,GAAE,EAAG,GAAE,CAAE,GAAE,C;;QAExC,OAAO,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,
C;O;;GAG/B,G;EACF,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;IACI,IAAI,MAAO,GAAE,Y;MACT,IAAI,IA
AI,C;MACR,IAAI,SAAS,SAAS,O;MAEtB,KAAK,IAAI,IAAI,CAAb,EAAgB,CAAE,GAAE,MAApB,EAA4B,CA
AC,EAA7B,C;QACI,IAAI,SAAS,CAAC,CAAD,CAAI,KAAI,QAAS,IAAG,SAAS,CAAC,CAAD,CAAI,KAAI,C
AAC,QAAnD,C;UACI,OAAO,Q;;QAEX,CAAE,IAAG,SAAS,CAAC,CAAD,CAAI,GAAE,SAAS,CAAC,CAAD,
C;;MAEjC,OAAO,IAAI,KAAK,CAAC,CAAD,C;K;;EAGxB,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;IACI,I
AAI,MAAO,GAAE,a;MACT,OAAO,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,IAAI,O;K;;EAGjC,IAAI,OAAO,IA
AI,KAAM,KAAI,WAAzB,C;IACI,IAAI,KAAM,GAAE,a;MACR,OAAO,IAAI,IAAI,CAAC,CAAD,CAAI,GAAE,
IAAI,M;K;;EAGjC,IAAI,OAAO,IAAI,MAAO,KAAI,WAA1B,C;IACI,IAAI,MAAO,GAAG,oB;MACV,OAAO,a;
QACH,IAAI,SAAS,CAAE,KAAI,C;QACnB,IAAI,MAAO,KAAI,CAAf,C;UACI,OAAO,E;;QAEX,OAAO,EAAG,
IAAG,GAAG,CAAC,MAAD,CAAS,GAAE,GAAI,GAAE,CAAvB,CAA0B,GAAE,CAAtC;A,O;KAEN,CAAC,IA
AI,IAAL,EAAW,IAAI,IAAf,C;;;EAIN,IAAI,OAAO,WAAW,OAAQ,KAAI,WAAlC,C;IACI,WAAW,OAAQ,GAA
E,a;MACjB,OAAO,CAAE,IAAG,IAAK,IAAG,CAAC,UAAW,IAAG,IAAK,IAAG,CAAC,UAAU,UAAW,KAAI,
SAAS,UAAU,U;K;;EAIhG,IAAI,OAAO,KAAK,UAAU,KAAM,KAAI,WAApC,C;IAEyB;IAArB,MAAM,eAAe,C
AAC,KAAK,UAAN,EAAkB,MAAlB,EAA0B,CAC3C,KAD2C,EACpC,iB;MAGH;UAAI,IAAK,IAAG,IAAZ,C;Q
ACI,MAAM,IAAI,SAAJ,CAAc,6BAAd,C;;MAGV,IAAI,IAAI,MAAM,CAAC,IAAD,CAAd;A,MAGA,IAAI,MA
AM,CAAC,OAAQ,KAAI,CAAvB;A,MAGA,IAAI,QAAQ,SAAS,CAAC,CAAD,C;MACrB,IAAI,gBAAgB,KAA

M,IAAG,CAA7B;A,MAGA,IAAI,IAAI,aAAc,GAAE,CAAE,GAClB,IAAI,IAAI,CAAC,GAAI,GAAE,aAAP,EAA
sB,CAAtB,CADU,GAElB,IAAI,IAAI,CAAC,aAAD,EAAgB,GAAhB,CAFhB;A,MAKA,IAAI,MAAM,SAAS,CA
AC,CAAD,C;MACnB,IAAI,cAAc,GAAI,KAAI,SAAU,GAClB,GADkB,GACZ,GAAI,IAAG,CAD/B;A,MAIA,IA
AI,aAAa,WAAY,GAAE,CAAE,GAChB,IAAI,IAAI,CAAC,GAAI,GAAE,WAAP,EAAoB,CAApB,CADQ,GAEhB
,IAAI,IAAI,CAAC,WAAD,EAAc,GAAd,CAFzB;A,MAKA,OAAO,CAAE,GAAE,UAAX,C;QACI,CAAC,CAAC,
CAAD,CAAI,GAAE,K;QACP,CAAC,E;;;MAIL,OAAO,C;KAvCgC,CAA1B,C;;EA4HvB,CAhFD,Y;IACG,yC;M
ACI,IAAI,MAAO,GAAE,CAAb,C;QAAgB,OAAO,IAAI,IAAI,CAAC,CAAD,EAAI,MAAO,GAAE,MAAb,C;MA
C/B,OAAO,IAAI,IAAI,CAAC,MAAD,EAAS,MAAT,C;K;IAEnB,qC;MACI,IAAI,OAAO,GAAI,KAAI,WAAnB,
C;QACI,GAAI,GAAE,IAAI,O;;MAEd,KAAM,GAAE,eAAe,CAAC,KAAM,IAAG,CAAV,EAAa,IAAI,OAAjB,C;
MACvB,GAAI,GAAE,IAAI,IAAI,CAAC,KAAD,EAAQ,eAAe,CAAC,GAAD,EAAM,IAAI,OAAV,CAAvB,C;M
ACd,OAAO,IAAI,IAAI,YAAR,CAAqB,IAAI,SAAS,CAAC,KAAD,EAAQ,GAAR,CAAlC,C;K;IAGX,IAAI,SAA
S,CAAC,SAAD,EAAY,UAAZ,EAAwB,WAAxB,EAAqC,UAArC,EAAiD,YAAjD,EAA+D,YAA/D,C;IACb,KAA
K,IAAI,IAAI,CAAb,EAAgB,CAAE,GAAE,MAAM,OAA1B,EAAmC,EAAE,CAArC,C;MACI,IAAI,aAAa,MAA
M,CAAC,CAAD,C;MACvB,IAAI,OAAO,UAAU,UAAU,KAAM,KAAI,WAAzC,C;QACI,MAAM,eAAe,CAAC,
UAAU,UAAX,EAAuB,MAAvB,EAA+B,CAAChD,KADgD,EAClzC,KAAK,UAAU,KAD0B,CAA/B,C;;MAIzB,IA
AI,OAAO,UAAU,UAAU,MAAO,KAAI,WAA1C,C;QACI,MAAM,eAAe,CAAC,UAAU,UAAX,EAAuB,OAAvB,
EAAgC,CACjD,KADiD,EAC1C,eAD0C,CAAhC,C;;;;;MAQJ,CAApB,Y;OAAc,MAAM,CAAC,IAAD,EAAO,IA
AI,UAAJ,CAAe,CAAf,CAAP,E;;MAErB,IAAI,QAAQ,QAAQ,UAAU,M;MAC9B,MAAM,eAAe,CAAC,QAAQ,U
AAT,EAAqB,OAArB,EAA8B,CAC/C,KAD+C,EACxC,uB;QACH,OAAO,KAAK,KAAK,CAAC,IAAD,EAAO,IA
AP,EAAa,EAAE,MAAM,KAAK,CAAC,KAAD,CAA1B,C;OAF0B,CAA9B,C;;;IASzB,KAAK,IAAI,IAAI,CAAb,
EAAgB,CAAE,GAAE,MAAM,OAA1B,EAAmC,EAAE,CAArC,C;MACI,IAAI,aAAa,MAAM,CAAC,CAAD,C;M
ACvB,IAAI,OAAO,UAAU,UAAU,IAAK,KAAI,WAAxC,C;QACI,MAAM,eAAe,CAAC,UAAU,UAAX,EAAuB,
KAAvB,EAA8B,CAC/C,KAD+C,EACxC,0B;UACH,OAAO,EAAE,MAAM,KAAK,CAAC,IAAD,CAAM,IAAI,C
AAC,QAAD,EAAW,IAAX,C;SAFa,CAA9B,C;;;;;IAU7B,IAAI,uBAAuB,gB;MACvB,IAAI,CAAE,GAAE,CAAR,
C;QAAW,OAAO,E;MAClB,IAAI,CAAE,GAAE,CAAR,C;QAAW,OAAO,C;MAElB,IAAI,CAAE,KAAI,CAAV,
C;QACI,IAAI,CAAE,KAAI,CAAV,C;UAAa,OAAO,C;QAEpB,IAAI,KAAK,CAAE,GAAE,C;QACb,OAAO,EAA
G,KAAI,CAAE,GAAE,CAAE,GAAE,CAAE,GAAE,CAAE,GAAE,CAAE,GAAE,EAAG,GAAE,CAAE,GAAE,EAAG,GAAE,EAAF,GAAO,C;;MAG7C,OA
AO,CAAE,KAAI,CAAE,GAAG,CAAE,KAAI,CAAE,GAAE,CAAE,GAAE,CAAF,GAAM,CAAjB,GAAsB,E;K;IAGzC,KAAK,I
AAI,IAAI,CAAb,EAAgB,CAAE,GAAE,MAAM,OAA1B,EAAmC,EAAE,CAArC,C;MACI,IAAI,aAAa,MAAM,C
AAC,CAAD,C;MACvB,IAAI,OAAO,UAAU,UAAU,KAAM,KAAI,WAAzC,C;QACI,MAAM,eAAe,CAAC,UAA
U,UAAX,EAAuB,MAAvB,EAA+B,CAChD,KADgD,EACzC,2B;UACH,OAAO,KAAK,UAAU,KAAK,KAAK,C
AAC,IAAD,EAAO,eAAgB,IAAG,oBAA1B,C;SAFY,CAA/B,C;;;GAO/B,G;ECxXU;;;IAAZ,MAAM,KAAM,GAA
E,CACV,KADU,EACH,OADG,EAEV,SAFU,EAEC,WAFD,EAGV,MAHU,EAGF,QAHE,C;EAMd,MAAM,WA
AY,GAAE,2C;IAChB,IAAI,qBAAqB,MAAM,yBAAyB,CAAC,KAAD,EAAQ,YAAR,C;IACxD,IAAI,kBAAmB,I
AAG,IAAK,IAAG,kBAAkB,IAAK,IAAG,IAA5D,C;MACI,OAAO,kBAAkB,IAAI,KAAK,CAAC,UAAD,C;;IAGt
C,kBAAmB,GAAE,MAAM,yBAAyB,CAAC,UAAD,EAAa,YAAb,C;IACpD,IAAI,kBAAmB,IAAG,IAAK,IAAG,
OAAQ,IAAG,kBAA7C,C;MACI,OAAO,UAAU,CAAC,YAAD,C;;IAGrB,OAAO,MAAM,WAAW,CAAC,UAAD,
EAAa,MAAM,eAAe,CAAC,KAAD,CAAlC,EAA2C,YAA3C,C;G;EAG5B,MAAM,WAAY,GAAE,kD;IAChB,IA
AI,qBAAqB,MAAM,yBAAyB,CAAC,KAAD,EAAQ,YAAR,C;IACxD,IAAI,kBAAmB,IAAG,IAAK,IAAG,kBAA
kB,IAAK,IAAG,IAA5D,C;MACI,kBAAkB,IAAI,KAAK,CAAC,UAAD,EAAa,KAAb,C;MAC3B,M;;IAGJ,kBAA
mB,GAAE,MAAM,yBAAyB,CAAC,UAAD,EAAa,YAAb,C;IACpD,IAAI,kBAAmB,IAAG,IAAK,IAAG,OAAQ,I
AAG,kBAA7C,C;MACI,UAAU,CAAC,YAAD,CAAe,GAAE,K;MAC3B,M;;IAGJ,MAAM,WAAW,CAAC,UAA
D,EAAa,MAAM,eAAe,CAAC,KAAD,CAAlC,EAA2C,YAA3C,EAAyD,KAAzD,C;G;EAGrB,iD;IACI,IAAI,IAA
K,KAAI,KAAb,C;MAAoB,OAAO,I;IAE3B,IAAI,WAAW,IAAI,W;IACnB,IAAI,QAAS,IAAG,IAAhB,C;MACI,I
AAI,aAAa,QAAQ,W;MACzB,KAAK,IAAI,IAAI,CAAb,EAAgB,CAAE,GAAE,UAAU,OAA9B,EAAuC,CAAC,E
AAxC,C;QACI,IAAI,0BAA0B,CAAC,UAAU,CAAC,CAAD,CAAX,EAAgB,KAAhB,CAA9B,C;UACI,OAAO,I;;;
;IAKnB,IAAI,iBAAiB,IAAI,UAAW,IAAG,IAAK,GAAE,MAAM,eAAe,CAAC,IAAI,UAAL,CAAvB,GAA0C,I;I
ACtF,IAAI,mBAAmB,cAAe,IAAG,IAAK,GAAE,cAAc,YAAhB,GAA+B,I;IAC7E,OAAO,gBAAiB,IAAG,IAAK,

IAAG,0BAA0B,CAAC,gBAAD,EAAmB,KAAnB,C;G;EASnD;;;;;IAAd,MAAM,OAAQ,GAAE,yB;IACZ,IAAI,K
AAM,KAAI,MAAd,C;MACI,QAAQ,OAAO,MAAf,C;aACS,Q;aACA,Q;aACA,S;aACA,U;UACD,OAAO,I;;UAE
P,OAAO,MAAO,YAAW,M;;;IAIrC,IAAI,MAAO,IAAG,IAAK,IAAG,KAAM,IAAG,IAAK,KAAI,OAAO,MAAO
,KAAI,QAAS,IAAG,OAAO,MAAO,KAAI,UAApD,CAApC,C;MACI,OAAO,K;;IAGX,IAAI,OAAO,KAAM,KA
AI,UAAW,IAAG,MAAO,YAAW,KAArD,C;MACI,OAAO,I;;IAGX,IAAI,QAAQ,MAAM,eAAe,CAAC,KAAD,C;
IACjC,IAAI,cAAc,KAAM,IAAG,IAAK,GAAE,KAAK,YAAP,GAAsB,I;IACtD,IAAI,WAAY,IAAG,IAAK,IAAG
,YAAa,IAAG,WAA3C,C;MACI,IAAI,WAAW,WAAW,W;MAC1B,IAAI,QAAQ,KAAM,KAAI,MAAM,KAAK,O
AAjC,C;QACI,OAAO,MAAO,KAAI,K;;;IAI1B,IAAI,gBAAgB,KAAK,WAAzB;A,IAGA,IAAI,aAAc,IAAG,IAAr
B,C;MACI,OAAO,MAAO,YAAW,K;;IAG7B,IAAI,aAAa,KAAM,KAAI,MAAM,KAAK,UAAW,IAAG,MAAM,
YAAa,IAAG,IAA1E,C;MACI,OAAO,0BAA0B,CAAC,MAAM,YAAP,EAAqB,KAArB,C;;IAGrC,OAAO,K;G;EA
GX,MAAM,SAAU,GAAE,a;IACd,OAAO,OAAO,CAAE,IAAG,QAAS,IAAG,CAAE,YAAW,MAAM,K;G;EAGt
D,MAAM,OAAQ,GAAE,iB;IACZ,OAAO,KAAM,YAAW,MAAM,U;G;EAGlC,MAAM,aAAc,GAAE,iB;IAClB,I
AAI,OAAO,OAAO,K;IAElB,OAAO,IAAK,KAAI,QAAS,IAClB,IAAK,KAAI,SAAU,IACnB,MAAM,SAAS,CAA
C,KAAD,CAAQ,IACvB,MAAM,OAAO,CAAC,KAAD,EAAQ,MAAM,OAAO,WAArB,C;G;EAGxB,MAAM,eA
AgB,GAAE,iB;IACpB,OAAO,OAAO,KAAM,KAAI,QAAS,IAAG,MAAM,OAAO,CAAC,KAAD,EAAQ,MAAM
,OAAO,aAArB,C;G;;;;;;;;;aCnDV,gB;;;;ICrE3C,gB;MAkBI,4B;MAjBA,aAA6C,E;MAC7C,gBAAgD,C;K;4EAG5
C,Y;MAAQ,iB;K;+EAGR,Y;MAAQ,oB;K;qCAEZ,iB;MAAyC,OAAQ,0BAAR,YAAQ,EAAU,KAAM,QAAhB,C;
K;4BAEjD,iB;MAAmC,gBAAS,K;K;8BAE5C,Y;MAA+B,OAAG,MAAH,kBAA8B,IAA9B,C;K;8BAE/B,Y;MAA
0B,gB;K;IAE1B,0B;MAAA,8B;K;;;IAAA,sC;MAAA,qC;QAAA,oB;;MAAA,8B;K;;IDfJ,mC;MAC4C,oBAAa,MA
AS,IAAT,CAAb,EAA6B,SAA7B,C;K;gEAE5C,yB;MAAA,mB;MAAA,6B;QAC2D,YAAa,QAAS,IAAT,C;QAIv
D,Q;QAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,MAAM,CAAN,IALgF,IAKrE,
CAAK,CAAL,C;;QALwC,OAOhD,K;O;KARX,C;gEAGA,uB;MAEiB,Q;MAAA,OAAA,KAAM,OAAN,GAAa,C
AAb,I;MAAb,aAAU,CAAV,iB;QACI,MAAM,CAAN,IAAW,KAAK,CAAL,C;;MAEf,OAAO,K;K;IAGX,kC;MAI
iB,IAAN,I;MAFP,aAAsB,MAAe,IAAf,C;MACtB,gBAAkB,c;MAEd,IADS,IACT,mBADS,IACT,EAAM,IAAN,E;
QAAc,oBAAa,MAAb,EAAqB,KAArB,C;WACd,WAFS,IAET,S;QAAS,a;;QAZA,U;QAAA,SAaqB,MAbf,OAAN,
GAAa,CAAb,I;QAAb,aAAU,CAAV,mB;UAakC,MAZ9B,CAAM,CAAN,IAYsC,IAZ3B,CAAK,CAAL,C;;QAYH,
OAAsB,M;;MAHlC,W;K;2EAOJ,yB;MAAA,iC;MAAA,6B;QACoF,YAAa,aAAa,IAAb,EAAmB,KAAnB,C;QAlB
hF,Q;QAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,MAAM,CAAN,IAiBoH,IAjBz
G,CAAK,CAAL,C;;QAiBiE,OAfzE,K;O;KAcX,C;IAGA,+B;MAKiB,IAAN,I;MAFP,aAAa,IAAO,WAAP,CAAm
B,IAAnB,C;MACb,gBAAkB,W;MAEd,IADS,IACT,mBADS,IACT,EAAM,IAAN,YADS,IACT,EAAY,KAAZ,E;
QAAqB,a;;QA1BZ,U;QAAA,SA2BkB,MA3BZ,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,mB;UA2B+B,MA1B
3B,CAAM,CAAN,IA0BmC,IA1BxB,CAAK,CAAL,C;;QA0BH,OAAmB,M;;MAF/B,W;K;qEAMJ,yB;MAAA,2B;
MAAA,gC;MAAA,6B;QAGiB,Q;QADb,YAAY,UAAU,IAAV,EAAgB,IAAhB,C;QACC,OAAA,KAAM,OAAN,G
AAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,YACY,eAAK,CAAL,E;UACT,KAAK,CAAC,CAAD,CAAR,GAAc,
K;;QAElB,OAAO,K;O;KARX,C;mFAWA,yB;MAAA,mB;MAAA,gC;MAAA,6B;QAGiB,Q;QADb,YAAY,QAA
Y,IAAZ,C;QACC,OAAA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,YACY,eAAK,CAAL,E;U
ACT,KAAK,CAAC,CAAD,CAAR,GAAc,K;;QAElB,OAAO,K;O;KARX,C;IAWA,+B;MAIiB,IAAN,I;MAFP,aA
AsB,MAAY,IAAZ,C;MACtB,gBAAkB,W;MAEd,IADS,IACT,mBADS,IACT,EAAM,IAAN,E;QAAc,oBAAa,MA
Ab,K;WACd,WAFS,IAET,S;QAAS,a;;QA3DA,U;QAAA,SA4DkB,MA5DZ,OAAN,GAAa,CAAb,I;QAAb,aAAU,
CAAV,mB;UA4D+B,MA3D3B,CAAM,CAAN,IA2DmC,IA3DxB,CAAK,CAAL,C;;QA2DH,OAAmB,M;;MAH/B
,W;K;qEAOJ,yB;MAAA,2B;MAAA,6B;QAC2E,YAAa,UAAU,IAAV,EAAgB,KAAhB,C;QAjEvE,Q;QAAA,OA
AA,KAAM,OAAN,GAAa,CAAb,I;QAAb,aAAU,CAAV,iB;UACI,MAAM,CAAN,IAgEwG,IAhE7F,CAAK,CAA
L,C;;QAgEwD,OA9DhE,K;O;KA6DX,C;IAGA,wC;MACiB,Q;MAAA,OAAA,KAAM,OAAN,GAAa,CAAb,I;MA
Ab,aAAU,CAAV,iB;QACI,MAAM,CAAN,IAAW,S;;MAEf,OAAO,K;K;IElFX,iC;MAAA,qC;MAEI,iBAC8B,Q;
MAE9B,iBAC8B,sB;MAE9B,yBAEsC,MAAM,G;MAE5C,yBAEsC,CAAC,GAAD,GAAO,G;MAE7C,WAEwB,E
AAE,MAAM,GAAR,C;MAExB,kBACuB,C;MAEvB,iBACsB,E;K;;;IAxB1B,6C;MAAA,4C;QAAA,2B;;MAAA,q
C;K;IA2BA,gC;MAAA,oC;MAEI,iBAC6B,O;MAE7B,iBAC6B,Y;MAE7B,yBAEqC,MAAO,G;MAE5C,yBAEqC,
CAAC,GAAD,GAAQ,G;MAE7C,WAEuB,EAAE,MAAO,GAAT,C;MAEvB,kBACuB,C;MAEvB,iBACsB,E;K;;;I

AxB1B,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;IA2BA,8B;MAAA,kC;MAEI,iBACqB,W;MAErB,iBACqB,U;M
AErB,kBACuB,C;MAEvB,iBACsB,E;K;;;IAZ1B,0C;MAAA,yC;QAAA,wB;;MAAA,kC;K;IAeA,+B;MAAA,mC;
MAEI,iBACyB,MAAM,KAAT,U;MAEtB,iBACyB,MAAM,KAAT,U;MAEtB,kBACuB,C;MAEvB,iBACsB,E;K;;;
IAZ1B,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;IAeA,gC;MAAA,oC;MAEI,iBACuB,U;MAEvB,iBACuB,K;MA
EvB,kBACuB,C;MAEvB,iBACsB,E;K;;;IAZ1B,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;IAeA,+B;MAAA,mC;M
AEI,iBACsB,Q;MAEtB,iBACsB,G;MAEtB,kBACuB,C;MAEvB,iBACsB,C;K;;;IAZ1B,2C;MAAA,0C;QAAA,yB;
;MAAA,mC;K;IAeA,+B;MAAA,mC;MAEI,iBACmC,C;MAEnC,iBACmC,K;MAEnC,0BAC4C,K;MAE5C,0BAC
4C,K;MAE5C,yBAC2C,K;MAE3C,yBAC2C,K;MAE3C,qBACuC,uB;MAEvC,qBACuC,sB;MAEvC,kBACuB,C;
MAEvB,iBACsB,E;K;;;IA9B1B,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;IAiCA,iC;MAAA,qC;K;;;IAAA,6C;MA
AA,4C;QAAA,2B;;MAAA,qC;K;IAEA,kC;MAAA,sC;K;;;IAAA,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;aC27vBoB,gB;;;cClsuB0C,mB;;gBAyEvC,yB;eAAyB,wB;;;uBAgBzB,gC;sBAAwB,+B;mCA4Jj
C,qB;mCA5ImC,qB;;kBAQ1B,2B;iBAA0B,0B;;;;;;;eCvrBgB,wB;sBCoBA,sB;iBCnBA,0B;;;;;;;;;;;;;;;;;;;gCCxTlB
,yC;+BCVA,uC;+BCAA,sC;;aC+EgD,e;gCC0E/E,+B;+BAIW,sC;gCCyxCc,+B;0BAHvB,kC;uBA96BO,gC;yBAq
XD,iC;0BACA,mC;yBA8JA,iC;gCAmZP,oC;+BAbc,oC;+BAEC,+B;yBAEQ,kC;;gBCz1C6C,yB;;;;;;;;;;;;;;;;;;;;;;;;
;;
;;;;;;;;ICtErF,yB;K;;IAQA,6B;K;;IAUA,oC;K;;IC3BA,kD;MAMuF,wC;K;IANvF,4CAOI,Y;MAAuC,8B;K;IAP3C
,8E;ICGA,kD;MAQuF,wC;K;IARvF,4CASI,Y;MAAuC,8B;K;IAT3C,8E;0FdOA,qB;MAQI,OAAO,UAAI,CAAJ,C
;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,O
AAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;
4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OA
AO,UAAI,CAAJ,C;K;0FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4
FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAA
O,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4F
AGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;0FAGX,qB;MAQI,OAAO,
UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAG
X,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UA
AI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,
qB;MAQI,OAAO,UAAI,CAAJ,C;K;0FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAA
I,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB
;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,C
AAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;0FAGX,qB;M
AQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CA
AJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MA
QI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,
C;K;4FAGX,qB;MAQI,OAAO,UAAI,CAAJ,C;K;IAGX,sC;MAII,OAAO,mBAAQ,OAAR,KAAoB,C;K;IAG/B,w
C;MAII,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/B,wC;
MAII,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAG/B,wC;MA
MW,c;;QA88XS,Q;QAAhB,iD;UAAgB,gBAAhB,e;UAAsB,IAAc,SA98XvB,YA88XS,C;YAAwB,aAAO,I;YAAP,
e;;;QAC9C,aAAO,K;;;MA/8XP,iB;K;IAGJ,wC;MAMW,c;;QA+8XS,Q;QAAhB,iD;UAAgB,gBAAhB,e;UAAsB,I
AAc,SA/8XvB,YA+8XS,C;YAAwB,aAAO,I;YAAP,e;;;QAC9C,aAAO,K;;;MAh9XP,iB;K;IAGJ,wC;MAII,OAAO
,qBAAQ,OAAR,KAAoB,C;K;IAG/B,wC;MAII,OAAO,qBAAQ,OAAR,KAAoB,C;K;oGAkE/B,yB;MAAA,8D;M
AAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,
C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UA
AI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,C
AAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAA
A,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa
,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GA
AsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,

SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,y
B;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAs
D,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBA
A3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;sGAUA,yB;MAAA,8D;MAAA,gC;MAA
A,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,uBAAa,KAAb,E;
O;KAPjE,C;oGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAAY,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;qGA
UA,yB;MAAA,qD;MAAA,mC;QAOI,OAAY,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA,sD
;MAAA,mC;QAOI,OAAY,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAO
I,OAAY,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAAY,UAAL,S
AAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAAY,UAAL,SAAK,EAAU,KAA
V,C;O;KAPhB,C;sGAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAAY,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;s
GAUA,yB;MAAA,sD;MAAA,mC;QAOI,OAAY,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;sGAUA,yB;MAAA
,sD;MAAA,mC;QAOI,OAAY,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;8EAUA,gC;MAOW,sB;;QAkcS,Q;QA
AhB,iD;UAAgB,cAAhB,e;UAAsB,IAlcH,SAkcO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9
C,qBAAO,I;;;MAncP,yB;K;gFAGJ,gC;MAOW,sB;;QAgcS,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IAhcH,SAgc
O,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MAjcP,yB;K;gFAGJ,gC;MAOW,s
B;;QA8bS,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IA9bH,SA8bO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;Y
AAP,uB;;;QAC9C,qBAAO,I;;;MA/bP,yB;K;gFAGJ,gC;MAOW,sB;;QA4bS,Q;QAAhB,iD;UAAgB,cAAhB,e;UAA
sB,IA5bH,SA4bO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MA7bP,yB;K;gFA
GJ,gC;MAOW,sB;;QA0bS,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IA1bH,SA0bO,CAAU,OAAV,CAAJ,C;YAA
wB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MA3bP,yB;K;gFAGJ,gC;MAOW,sB;;QAwbS,Q;QAAhB,iD;UAA
gB,cAAhB,e;UAAsB,IAxbH,SAwbO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;
;MAzbP,yB;K;gFAGJ,gC;MAOW,sB;;QAsbS,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IAtbH,SAsbO,CAAU,OA
AV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MAvbP,yB;K;gFAGJ,gC;MAOW,sB;;QAobS,Q
;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IApbH,SAobO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;Q
AC9C,qBAAO,I;;;MArbP,yB;K;gFAGJ,yB;MAqbA,oC;MAAA,gC;MArbA,uC;QAOW,sB;;UAkbS,Q;UAAhB,iD;
YAAgB,cAAhB,0B;YAAsB,IAlbH,SAkbO,CAAU,oBAAV,CAAJ,C;cAAwB,qBAAO,O;cAAP,uB;;;UAC9C,qBA
AO,I;;;QAnbP,yB;O;KAPJ,C;sFAUA,yB;MAi2CA,0D;MAAA,+C;MAj2CA,uC;QAOW,qB;;UAg2CO,Q;UAAA,O
AAa,SAAR,sBAAQ,CAAb,W;UAAd,OAAc,cAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAl2Cc,SA
k2CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAp2CP,wB;O;KAPJ,C;wFAU
A,yB;MAo2CA,0D;MAAA,+C;MAp2CA,uC;QAOW,qB;;UAm2CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;U
AAd,OAAc,cAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAr2Cc,SAq2CV,CAAU,OAAV,CAAJ,C;c
AAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAv2CP,wB;O;KAPJ,C;wFAUA,yB;MAu2CA,0D;MAAA,+C;
MAv2CA,uC;QAOW,qB;;UAs2CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAd,OAAc,cAAd,C;YAAc,uB;Y
ACV,cAAc,UAAK,KAAL,C;YACd,IAx2Cc,SAw2CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UA
E5B,oBAAO,I;;;QA12CP,wB;O;KAPJ,C;wFAUA,yB;MA02CA,0D;MAAA,+C;MA12CA,uC;QAOW,qB;;UAy2C
O,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAd,OAAc,cAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YA
Cd,IA32Cc,SA22CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA72CP,wB;O;
KAPJ,C;wFAUA,yB;MA62CA,0D;MAAA,+C;MA72CA,uC;QAOW,qB;;UA42CO,Q;UAAA,OAAa,SAAR,sBAA
Q,CAAb,W;UAAd,OAAc,cAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IA92Cc,SA82CV,CAAU,OA
AV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAh3CP,wB;O;KAPJ,C;wFAUA,yB;MAg3CA,0
D;MAAA,+C;MAh3CA,uC;QAOW,qB;;UA+2CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAd,OAAc,cAAd,
C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAj3Cc,SAi3CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;c
AAP,sB;;;UAE5B,oBAAO,I;;;QAn3CP,wB;O;KAPJ,C;wFAUA,yB;MAm3CA,0D;MAAA,+C;MAn3CA,uC;QAO
W,qB;;UAk3CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,W;UAAd,OAAc,cAAd,C;YAAc,uB;YACV,cAAc,UAAK
,KAAL,C;YACd,IAp3Cc,SAo3CV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA
t3CP,wB;O;KAPJ,C;wFAUA,yB;MAs3CA,0D;MAAA,+C;MAt3CA,uC;QAOW,qB;;UAq3CO,Q;UAAA,OAAa,S
AAR,sBAAQ,CAAb,W;UAAd,OAAc,cAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IAv3Cc,SAu3CV,

CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAz3CP,wB;O;KAPJ,C;wFAUA,yB;
MAy3CA,0D;MAAA,+C;MAAA,oC;MAz3CA,uC;QAOW,qB;;UAw3CO,Q;UAAA,OAAa,SAAR,sBAAQ,CAAb,
W;UAAd,OAAc,cAAd,C;YAAc,uB;YACV,cAAc,UAAK,KAAL,C;YACd,IA13Cc,SA03CV,CAAU,oBAAV,CAA
J,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA53CP,wB;O;KAPJ,C;IAUA,0B;MAMI,IAovNO,qBAA
Q,CApvNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAMI,IAivNO,q
BAAQ,CAjvNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAMI,IA8u
NO,qBAAQ,CA9uNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;MAMI,
IA2uNO,qBAAQ,CA3uNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,4B;M
AMI,IAwuNO,qBAAQ,CAxuNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IAGX,
4B;MAMI,IAquNO,qBAAQ,CAruNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;K;IA
GX,4B;MAMI,IAkuNO,qBAAQ,CAluNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAAL,C;
K;IAGX,4B;MAMI,IA+tNO,qBAAQ,CA/tNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,CAA
L,C;K;IAGX,4B;MAMI,IA4tNO,qBAAQ,CA5tNf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,C
AAL,C;K;kFAGX,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;
UAAsB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;kF
ASA,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UAAsB,IAA
I,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;mFASA,yB;MA
AA,iE;MAAA,uC;QAKoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UAAsB,IAAI,UAAU,OA
AV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;mFASA,yB;MAAA,iE;MAA
A,uC;QAKoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UAAsB,IAAI,UAAU,OAAV,CAAJ,C;
YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;mFASA,yB;MAAA,iE;MAAA,uC;QAKo
B,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UAAsB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OA
AO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;mFASA,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAAhB,
wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UAAsB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACr
D,MAAM,gCAAuB,mDAAvB,C;O;KANV,C;mFASA,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAAhB,wBAAgB,S
AAhB,gB;UAAgB,cAAA,SAAhB,M;UAAsB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,g
CAAuB,mDAAvB,C;O;KANV,C;mFASA,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAAhB,wBAAgB,SAAhB,gB;U
AAgB,cAAA,SAAhB,M;UAAsB,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mD
AAvB,C;O;KANV,C;mFASA,yB;MAAA,oC;MAAA,gC;MAAA,iE;MAAA,uC;QAKoB,Q;QAAhB,wBAAgB,SA
AhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UAAsB,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,
MAAM,gCAAuB,mDAAvB,C;O;KANV,C;kGASA,yB;MAAA,iE;MAAA,uC;QAS8C,IAAnC,I;QAAA,+B;;UAY
S,U;UAAhB,uD;YAAgB,cAAhB,iB;YACI,aAbwB,SAaX,CAAU,OAAV,C;YACb,IAAI,cAAJ,C;cACI,8BAAO,M;
cAAP,gC;;;UAGR,8BAAO,I;;;QAlBA,kC;QAAA,iB;UAAmC,MAAM,gCAAuB,8DAAvB,C;;QAAhD,OAAO,I;O;
KATX,C;8GAYA,gC;MASoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,aAAa,UAAU,O
AAV,C;QACb,IAAI,cAAJ,C;UACI,OAAO,M;;;MAGf,OAAO,I;K;IAGX,gC;MAII,OAoiNO,qBAAQ,CApiNR,GA
Ae,IAAf,GAAyB,UAAK,CAAL,C;K;IAGpC,kC;MAII,OAqiNO,qBAAQ,CAriNR,GAAe,IAAf,GAAyB,UAAK,C
AAL,C;K;IAGpC,kC;MAII,OAsiNO,qBAAQ,CAtiNR,GAAe,IAAf,GAAyB,UAAK,CAAL,C;K;IAGpC,kC;MAII,
OAuiNO,qBAAQ,CAviNR,GAAe,IAAf,GAAyB,UAAK,CAAL,C;K;IAGpC,kC;MAII,OAwiNO,qBAAQ,CAxiNR
,GAAe,IAAf,GAAyB,UAAK,CAAL,C;K;IAGpC,kC;MAII,OAyiNO,qBAAQ,CAziNR,GAAe,IAAf,GAAyB,UAA
K,CAAL,C;K;IAGpC,kC;MAII,OA0iNO,qBAAQ,CA1iNR,GAAe,IAAf,GAAyB,UAAK,CAAL,C;K;IAGpC,kC;M
AII,OA2iNO,qBAAQ,CA3iNR,GAAe,IAAf,GAAyB,UAAK,CAAL,C;K;IAGpC,kC;MAII,OA4iNO,qBAAQ,CA5i
NR,GAAe,IAAf,GAAyB,UAAK,CAAL,C;K;8FAGpC,gC;MAIoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAA
A,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;8FAGX,gC;MAIoB,
Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAA
O,O;;MACrD,OAAO,I;K;+FAGX,gC;MAIoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAs
B,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAIoB,Q;MAAhB,wBAAg
B,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAA
O,I;K;+FAGX,gC;MAIoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OA

AV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAIoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAA
gB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,gC;
MAIoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAA
wB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAIoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,
M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,yB;MAAA,oC;MAAA,
gC;MAAA,uC;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UAAsB,IAAI,UAAU
,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,OAAO,I;O;KALX,C;wFAQA,yB;MAAA,8D;MAAA,iD;QAKI,OA
AW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KALjE,C;0FA
QA,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,
GAAsD,aAAa,KAAb,C;O;KALjE,C;0FAQA,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,
wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KALjE,C;0FAQA,yB;MAAA,8D;MAAA,iD;Q
AKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KALj
E,C;0FAQA,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,
CAAtC,GAAsD,aAAa,KAAb,C;O;KALjE,C;0FAQA,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAA
c,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KALjE,C;0FAQA,yB;MAAA,8D;MA
AA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;
O;KALjE,C;0FAQA,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAA
I,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KALjE,C;0FAQA,yB;MAAA,8D;MAAA,gC;MAAA,iD;QAKI,OAAW
,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,uBAAa,KAAb,E;O;KALjE,C;IAQA,
qC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,uC;MA
MI,OAAW,SAAS,CAAT,IAAc,SAAS,0BAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,uC;MAMI,OA
AW,SAAS,CAAT,IAAc,SAAS,0BAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,uC;MAMI,OAAW,SA
AS,CAAT,IAAc,SAAS,0BAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,uC;MAMI,OAAW,SAAS,CA
AT,IAAc,SAAS,0BAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,uC;MAMI,OAAW,SAAS,CAAT,IAA
c,SAAS,0BAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS
,0BAA3B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,0BAA3
B,GAAsC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,0BAA3B,GAA
sC,UAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,qC;MAII,IAAI,eAAJ,C;QACI,wD;UACI,IAAI,UAAK,KAAL,SAAJ,
C;YACI,OAAO,K;;;;QAIf,8D;UACI,IAAI,gBAAW,UAAK,OAAL,CAAX,CAAJ,C;YACI,OAAO,O;;;;MAInB,OA
AO,E;K;IAGX,uC;MAII,wD;QACI,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IA
GX,uC;MAII,wD;QACI,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,uC;MA
II,wD;QACI,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,uC;MAII,wD;QAC
I,IAAI,gBAAW,UAAK,KAAL,CAAX,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,uC;MAMI,wD;QACI,
IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,uC;MAMI,wD;QACI,IAAI,YA
AW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,uC;MAII,wD;QACI,IAAI,YAAW,UAAK
,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,uC;MAII,wD;QACI,IAAI,YAAW,UAAK,KAAL,CA
Af,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;8FAGX,gC;MAII,wD;QACI,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ
,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAII,wD;QACI,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C
;UACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAII,wD;QACI,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;U
ACI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAII,wD;QACI,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;UA
CI,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAII,wD;QACI,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;UACI
,OAAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAII,wD;QACI,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;UACI,O
AAO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAII,wD;QACI,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;UACI,OA
AO,K;;;MAGf,OAAO,E;K;gGAGX,gC;MAII,wD;QACI,IAAI,UAAU,UAAK,KAAL,CAAV,CAAJ,C;UACI,OAA
O,K;;;MAGf,OAAO,E;K;gGAGX,yB;MAAA,oC;MAAA,uC;QAII,wD;UACI,IAAI,UAAU,sBAAK,KAAL,EAAV
,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;4FAYA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAIkB,Q;
QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,IAAI,UAAU,UAAK,KAAL,CA
AV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAIkB

,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,IAAI,UAAU,UAAK,KAAL,
CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QA
IkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,IAAI,UAAU,UAAK,KA
AL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAAA,+C;MAAA,uC;
QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,IAAI,UAAU,UAAK,
KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAAA,+C;MAAA,
uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,IAAI,UAAU,UA
AK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAAA,+C;MA
AA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,IAAI,UAAU,
UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAAA,+C;
MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,IAAI,UA
AU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MAAA
,+C;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,IAAI,
UAAU,UAAK,KAAL,CAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;8FAYA,yB;MAAA,0D;MA
AA,+C;MAAA,oC;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd,C;UAAc,u
B;UACV,IAAI,UAAU,sBAAK,KAAL,EAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;KATX,C;IAYA,yB;
MAQI,IAg7LO,qBAAQ,CAh7Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,wBAAL,C;K;IAG
X,2B;MAQI,IA26LO,qBAAQ,CA36Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,0BAAL,C;K
;IAGX,2B;MAQI,IAs6LO,qBAAQ,CAt6Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,0BAAL,
C;K;IAGX,2B;MAQI,IAi6LO,qBAAQ,CAj6Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK,0BA
AL,C;K;IAGX,2B;MAQI,IA45LO,qBAAQ,CA55Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAK
,0BAAL,C;K;IAGX,2B;MAQI,IAu5LO,qBAAQ,CAv5Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,U
AAK,0BAAL,C;K;IAGX,2B;MAQI,IAk5LO,qBAAQ,CAl5Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OA
AO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IA64LO,qBAAQ,CA74Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MAC
V,OAAO,UAAK,0BAAL,C;K;IAGX,2B;MAQI,IAw4LO,qBAAQ,CAx4Lf,C;QACI,MAAM,2BAAuB,iBAAvB,C;
MACV,OAAO,UAAK,0BAAL,C;K;gFAGX,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,
OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;U
ACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;gFAe
A,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,
W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,
OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAA
A,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,c
AAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAA
vB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAA
L,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,O
AAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MA
AA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAA
d,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,M
AAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QA
AA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C
;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFA
eA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAA
b,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAw
B,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,iE;MA
AA,uC;QAQkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,
cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDA
AvB,C;O;KAZV,C;iFAeA,yB;MAAA,0D;MAAA,+C;MAAA,oC;MAAA,iE;MAAA,uC;QAQkB,Q;QAAA,OAAa,
SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IA

AI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KAZV,C;IAeA,yC;MA
KsB,UAMA,M;MAPlB,IAAI,eAAJ,C;QACkB,OAAQ,WAAR,sBAAQ,CAAR,W;QAAd,OAAc,cAAd,C;UAAc,uB
;UACV,IAAI,UAAK,KAAL,SAAJ,C;YACI,OAAO,K;;;;QAID,SAAQ,WAAR,sBAAQ,CAAR,W;QAAd,OAAc,g
BAAd,C;UAAc,2B;UACV,IAAI,gBAAW,UAAK,OAAL,CAAX,CAAJ,C;YACI,OAAO,O;;;;MAInB,OAAO,E;K;I
AGX,2C;MAIkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,YA
AW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAIkB,Q;MAAA,OAAQ,WAAR,wB
AAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;
MAGf,OAAO,E;K;IAGX,2C;MAIkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAc,
uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAIkB,Q;MAA
A,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,gBAAW,UAAK,KAAL,CA
AX,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAMkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,
W;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAA
O,E;K;IAGX,2C;MAMkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,
IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAIkB,Q;MAAA,OAAQ,W
AAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,O
AAO,K;;;MAGf,OAAO,E;K;IAGX,2C;MAIkB,Q;MAAA,OAAQ,WAAR,wBAAQ,CAAR,W;MAAd,OAAc,cAAd
,C;QAAc,uB;QACV,IAAI,YAAW,UAAK,KAAL,CAAf,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;IAGX,+B;MAMI,
OA8jLO,qBAAQ,CA9jLR,GAAe,IAAf,GAAyB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OA6jLO,q
BAAQ,CA7jLR,GAAe,IAAf,GAAyB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OA4jLO,qBAAQ,CA
5jLR,GAAe,IAAf,GAAyB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OA2jLO,qBAAQ,CA3jLR,GAA
e,IAAf,GAAyB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OA0jLO,qBAAQ,CA1jLR,GAAe,IAAf,GA
AyB,UAAK,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OAyjLO,qBAAQ,CAzjLR,GAAe,IAAf,GAAyB,UAA
K,mBAAO,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OAwjLO,qBAAQ,CAxjLR,GAAe,IAAf,GAAyB,UAAK,mBAAO
,CAAP,IAAL,C;K;IAGpC,iC;MAMI,OAujLO,qBAAQ,CAvjLR,GAAe,IAAf,GAAyB,UAAK,mBAAO,CAAP,IA
AL,C;K;IAGpC,iC;MAMI,OAsjLO,qBAAQ,CAtjLR,GAAe,IAAf,GAAyB,UAAK,mBAAO,CAAP,IAAL,C;K;4F
AGpC,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;Q
AAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OA
AO,O;;QAEnC,OAAO,I;O;KAVX,C;4FAaA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SA
AR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,
UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MA
AA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,
cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6F
AaA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QA
Ad,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAA
O,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAA
R,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,U
AAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MAA
A,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,c
AAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FA
aA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd
,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O
;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,Y
AAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAA
U,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;6FAaA,yB;MAAA,0D;MAAA,+C;MAAA,o
C;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;
UACV,cAAc,UAAK,KAAL,C;UACd,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KA
VX,C;kFAaA,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;oFAWA,yB;MA
AA,mC;MAAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;oFAWA,yB;MAAA,mC;MAAA,gD;

MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;oFAWA,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAQI,O
AAO,kBAAO,cAAP,C;O;KARX,C;oFAWA,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,
C;O;KARX,C;oFAWA,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;oFAW
A,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;oFAWA,yB;MAAA,mC;M
AAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;oFAWA,yB;MAAA,mC;MAAA,gD;MAAA,4B;
QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;IAWA,qC;MAOI,IAoxKO,qBAAQ,CApxKf,C;QACI,MAAM,2BAAu
B,iBAAvB,C;MACV,OAAO,UAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;IAGX,qC;MAOI,IAgxKO,qBAAQ,CAh
xKf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;IAGX,sC;
MAOI,IA4wKO,qBAAQ,CA5wKf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAI,MAAO,iBAAQ,g
BAAR,CAAX,C;K;IAGX,sC;MAOI,IAwwKO,qBAAQ,CAxwKf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,O
AAO,UAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;IAGX,sC;MAOI,IAowKO,qBAAQ,CApwKf,C;QACI,MAAM,2
BAAuB,iBAAvB,C;MACV,OAAO,UAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;IAGX,sC;MAOI,IAgwKO,qBAA
Q,CAhwKf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;IA
GX,sC;MAOI,IA4vKO,qBAAQ,CA5vKf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,UAAI,MAAO,iB
AAQ,gBAAR,CAAX,C;K;IAGX,sC;MAOI,IAwvKO,qBAAQ,CAxvKf,C;QACI,MAAM,2BAAuB,iBAAvB,C;MA
CV,OAAO,UAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;IAGX,sC;MAOI,IAovKO,qBAAQ,CApvKf,C;QACI,MAA
M,2BAAuB,iBAAvB,C;MACV,OAAO,UAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;8FAGX,yB;MAAA,mC;MAA
A,4D;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;gGAUA,yB;MAAA,mC;MAAA,4D;MAAA,4B;QA
OI,OAAO,wBAAa,cAAb,C;O;KAPX,C;gGAUA,yB;MAAA,mC;MAAA,4D;MAAA,4B;QAOI,OAAO,wBAAa,cA
Ab,C;O;KAPX,C;gGAUA,yB;MAAA,mC;MAAA,4D;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;gG
AUA,yB;MAAA,mC;MAAA,4D;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;gGAUA,yB;MAAA,mC;
MAAA,4D;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;gGAUA,yB;MAAA,mC;MAAA,4D;MAAA,4
B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;gGAUA,yB;MAAA,mC;MAAA,4D;MAAA,4B;QAOI,OAAO,wBA
Aa,cAAb,C;O;KAPX,C;gGAUA,yB;MAAA,mC;MAAA,4D;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,
C;IAUA,2C;MAMI,IA+kKO,qBAAQ,CA/kKf,C;QACI,OAAO,I;MACX,OAAO,UAAI,MAAO,iBAAQ,gBAAR,C
AAX,C;K;IAGX,2C;MAMI,IA4kKO,qBAAQ,CA5kKf,C;QACI,OAAO,I;MACX,OAAO,UAAI,MAAO,iBAAQ,g
BAAR,CAAX,C;K;IAGX,4C;MAMI,IAykKO,qBAAQ,CAzkKf,C;QACI,OAAO,I;MACX,OAAO,UAAI,MAAO,i
BAAQ,gBAAR,CAAX,C;K;IAGX,4C;MAMI,IAskKO,qBAAQ,CAtkKf,C;QACI,OAAO,I;MACX,OAAO,UAAI,
MAAO,iBAAQ,gBAAR,CAAX,C;K;IAGX,4C;MAMI,IAmkKO,qBAAQ,CAnkKf,C;QACI,OAAO,I;MACX,OAA
O,UAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;IAGX,4C;MAMI,IAgkKO,qBAAQ,CAhkKf,C;QACI,OAAO,I;MA
CX,OAAO,UAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;IAGX,4C;MAMI,IA6jKO,qBAAQ,CA7jKf,C;QACI,OAA
O,I;MACX,OAAO,UAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;IAGX,4C;MAMI,IA0jKO,qBAAQ,CA1jKf,C;QA
CI,OAAO,I;MACX,OAAO,UAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;IAGX,4C;MAMI,IAujKO,qBAAQ,CAvjK
f,C;QACI,OAAO,I;MACX,OAAO,UAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;IAGX,2B;MAIiB,IAAN,I;MAAA,
QAAM,gBAAN,C;aACH,C;UAAK,MAAM,2BAAuB,iBAAvB,C;aACX,C;UAAK,iBAAK,CAAL,C;UAAL,K;;U
ACQ,MAAM,gCAAyB,kCAAzB,C;;MAHlB,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;U
AAK,MAAM,2BAAuB,iBAAvB,C;aACX,C;UAAK,iBAAK,CAAL,C;UAAL,K;;UACQ,MAAM,gCAAyB,kCAAz
B,C;;MAHlB,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,MAAM,2BAAuB,iBAAv
B,C;aACX,C;UAAK,iBAAK,CAAL,C;UAAL,K;;UACQ,MAAM,gCAAyB,kCAAzB,C;;MAHlB,W;K;IAOJ,6B;M
AIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,MAAM,2BAAuB,iBAAvB,C;aACX,C;UAAK,iBAAK,
CAAL,C;UAAL,K;;UACQ,MAAM,gCAAyB,kCAAzB,C;;MAHlB,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,
gBAAN,C;aACH,C;UAAK,MAAM,2BAAuB,iBAAvB,C;aACX,C;UAAK,iBAAK,CAAL,C;UAAL,K;;UACQ,M
AM,gCAAyB,kCAAzB,C;;MAHlB,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,
MAM,2BAAuB,iBAAvB,C;aACX,C;UAAK,iBAAK,CAAL,C;UAAL,K;;UACQ,MAAM,gCAAyB,kCAAzB,C;;
MAHlB,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,MAAM,2BAAuB,iBAAvB,C;
aACX,C;UAAK,iBAAK,CAAL,C;UAAL,K;;UACQ,MAAM,gCAAyB,kCAAzB,C;;MAHlB,W;K;IAOJ,6B;MAIiB
,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,MAAM,2BAAuB,iBAAvB,C;aACX,C;UAAK,iBAAK,CAA
L,C;UAAL,K;;UACQ,MAAM,gCAAyB,kCAAzB,C;;MAHlB,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBA

AN,C;aACH,C;UAAK,MAAM,2BAAuB,iBAAvB,C;aACX,C;UAAK,iBAAK,CAAL,C;UAAL,K;;UACQ,MAAM
,gCAAyB,kCAAzB,C;;MAHlB,W;K;oFAOJ,yB;MAAA,kF;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAMoB
,UAST,M;QAXP,aAAiB,I;QACjB,YAAY,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,
UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAA
Q,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAY,MAAM,gCAAuB,mDAAvB,C;QAElB,OAAO,6E;O;KAfX,C;oFAkB
A,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QACpB,YAAY,K;QACZ,
wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAA
W,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAY,MAA
M,gCAAuB,mDAAvB,C;QAElB,OAAO,2D;O;KAfX,C;qFAkBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC
;QAMoB,UAST,M;QAXP,aAAqB,I;QACrB,YAAY,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;U
ACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;Y
ACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAY,MAAM,gCAAuB,mDAAvB,C;QAElB,OAAO,2D;O;KAf
X,C;qFAkBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAmB,I;QACnB,YAA
Y,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KA
AJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UA
AY,MAAM,gCAAuB,mDAAvB,C;QAElB,OAAO,2D;O;KAfX,C;qFAkBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;
MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QACpB,YAAY,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,S
AAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,
SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAY,MAAM,gCAAuB,mDAAvB,C;QAElB,OAAO,i
E;O;KAfX,C;qFAkBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAqB,I;QACr
B,YAAY,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,I
AAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KA
AL,C;UAAY,MAAM,gCAAuB,mDAAvB,C;QAElB,OAAO,2D;O;KAfX,C;qFAkBA,yB;MAAA,kF;MAAA,iE;M
AAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAsB,I;QACtB,YAAY,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,c
AAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;
YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAY,MAAM,gCAAuB,mDAAvB,C;QAElB,
OAAO,2D;O;KAfX,C;qFAkBA,yB;MAAA,kF;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAu
B,I;QACvB,YAAY,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,
C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,C
AAC,KAAL,C;UAAY,MAAM,gCAAuB,mDAAvB,C;QAElB,OAAO,4D;O;KAfX,C;qFAkBA,yB;MAAA,oC;MA
AA,kF;MAAA,gC;MAAA,iE;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QAXP,aAAoB,I;QACpB,YAAY,K;QAC
Z,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,IAAI,UAAU,oBAAV,CAAJ,C;YACI,IAAI,K
AAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;U
AAY,MAAM,gCAAuB,mDAAvB,C;QAElB,OAAO,4E;O;KAfX,C;IAkBA,iC;MAII,OAAW,qBAAQ,CAAZ,GAA
e,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4
B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAA
W,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UA
AK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;
IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qB
AAQ,CAAZ,GAAe,UAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,mC;MAII,OAAW,qBAAQ,CAAZ,GAAe,UAAK,C
AAL,CAAf,GAA4B,I;K;gGAGvC,gC;MAMoB,Q;MAFhB,aAAiB,I;MACjB,YAAY,K;MACZ,wBAAgB,SAAhB,
gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UAClB,
SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;gGAGX,gC;MA
MoB,Q;MAFhB,aAAoB,I;MACpB,YAAY,K;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI
,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UAClB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,
CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAqB,I;MACrB,YAAY
,K;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAA
J,C;YAAW,OAAO,I;UAClB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,

OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAmB,I;MACnB,YAAY,K;MACZ,wBAAgB,SAAhB,gB;QAAgB,
cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UAClB,SAAS,O;U
ACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;M
AFhB,aAAoB,I;MACpB,YAAY,K;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,UAAU,O
AAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UAClB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,K
AAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAqB,I;MACrB,YAAY,K;MAC
Z,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YA
AW,OAAO,I;UAClB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO
,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aAAsB,I;MACtB,YAAY,K;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,S
AAhB,M;QACI,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UAClB,SAAS,O;UACT,QA
AQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MAMoB,Q;MAFhB,aA
AuB,I;MACvB,YAAY,K;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,UAAU,OAAV,CA
AJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UAClB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;Q
AAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,yB;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAFhB,aAAoB,I;
QACpB,YAAY,K;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,IAAI,UAAU,oBAAV,
CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,OAAO,I;YAClB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;
UAAY,OAAO,I;QACnB,OAAO,M;O;KAdX,C;IAiBA,4B;Me9qGI,IAAI,EfsrGI,KAAK,CetrGT,CAAJ,C;QACI,cf
qrGc,sD;QeprGd,MAAM,gCAAyB,OAAQ,WAAjC,C,C;;MfqrGV,OAAO,oBAAoB,gBAAV,mBAAO,CAAP,IAAU,
EAAc,CAAd,CAApB,C;K;IAGX,8B;Me1rGI,IAAI,EfksGI,KAAK,CelsGT,CAAJ,C;QACI,cfisGc,sD;QehsGd,MA
AM,gCAAyB,OAAQ,WAAjC,C,C;;MfisGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAc,CAAd,CAApB,
C;K;IAGX,8B;MetsGI,IAAI,Ef8sGI,KAAK,Ce9sGT,CAAJ,C;QACI,cf6sGc,sD;Qe5sGd,MAAM,gCAAyB,OAAQ,
WAAjC,C,C;;Mf6sGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAc,CAAd,CAApB,C;K;IAGX,8B;MeltG
I,IAAI,Ef0tGI,KAAK,Ce1tGT,CAAJ,C;QACI,cfytGc,sD;QextGd,MAAM,gCAAyB,OAAQ,WAAjC,C,C;;MfytGV,O
AAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAc,CAAd,CAApB,C;K;IAGX,8B;Me9tGI,IAAI,EfsuGI,KAAK,
CetuGT,CAAJ,C;QACI,cfquGc,sD;QepuGd,MAAM,gCAAyB,OAAQ,WAAjC,C,C;;MfquGV,OAAO,sBAAoB,gBA
AV,mBAAO,CAAP,IAAU,EAAc,CAAd,CAApB,C;K;IAGX,8B;Me1uGI,IAAI,EfkvGI,KAAK,CelvGT,CAAJ,C;Q
ACI,cfivGc,sD;QehvGd,MAAM,gCAAyB,OAAQ,WAAjC,C,C;;MfivGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,
IAAU,EAAc,CAAd,CAApB,C;K;IAGX,8B;MetvGI,IAAI,Ef8vGI,KAAK,Ce9vGT,CAAJ,C;QACI,cf6vGc,sD;Qe5
vGd,MAAM,gCAAyB,OAAQ,WAAjC,C,C;;Mf6vGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAc,CAA
d,CAApB,C;K;IAGX,8B;MelwGI,IAAI,Ef0wGI,KAAK,Ce1wGT,CAAJ,C;QACI,cfywGc,sD;QexwGd,MAAM,gC
AAyB,OAAQ,WAAjC,C,C;;MfywGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAc,CAAd,CAApB,C;K;I
AGX,8B;Me9wGI,IAAI,EfsxGI,KAAK,CetxGT,CAAJ,C;QACI,cfqxGc,sD;QepxGd,MAAM,gCAAyB,OAAQ,WA
AjC,C,C;;MfqxGV,OAAO,sBAAoB,gBAAV,mBAAO,CAAP,IAAU,EAAc,CAAd,CAApB,C;K;IAGX,gC;Me1xGI,I
AAI,EfkyGI,KAAK,CelyGT,CAAJ,C;QACI,cfiyGc,sD;QehyGd,MAAM,gCAAyB,OAAQ,WAAjC,C,C;;MfiyGV,O
AAO,gBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAc,CAAd,CAAhB,C;K;IAGX,kC;MetyGI,IAAI,Ef8yGI,KAAK
,Ce9yGT,CAAJ,C;QACI,cf6yGc,sD;Qe5yGd,MAAM,gCAAyB,OAAQ,WAAjC,C,C;;Mf6yGV,OAAO,kBAAgB,gB
AAV,mBAAO,CAAP,IAAU,EAAc,CAAd,CAAhB,C;K;IAGX,kC;MelzGI,IAAI,Ef0zGI,KAAK,Ce1zGT,CAAJ,C;
QACI,cfyzGc,sD;QexzGd,MAAM,gCAAyB,OAAQ,WAAjC,C,C;;MfyzGV,OAAO,kBAAgB,gBAAV,mBAAO,CA
AP,IAAU,EAAc,CAAd,CAAhB,C;K;IAGX,kC;Me9zGI,IAAI,Efs0GI,KAAK,Cet0GT,CAAJ,C;QACI,cfq0Gc,sD;
Qep0Gd,MAAM,gCAAyB,OAAQ,WAAjC,C,C;;Mfq0GV,OAAO,kBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAc,C
AAd,CAAhB,C;K;IAGX,kC;Me10GI,IAAI,Efk1GI,KAAK,Cel1GT,CAAJ,C;QACI,cfi1Gc,sD;Qeh1Gd,MAAM,gC
AAyB,OAAQ,WAAjC,C,C;;Mfi1GV,OAAO,kBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAc,CAAd,CAAhB,C;K;I
AGX,kC;Met1GI,IAAI,Ef81GI,KAAK,Ce91GT,CAAJ,C;QACI,cf61Gc,sD;Qe51Gd,MAAM,gCAAyB,OAAQ,WA
AjC,C,C;;Mf61GV,OAAO,kBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAc,CAAd,CAAhB,C;K;IAGX,kC;Mel2GI,I
AAI,Ef02GI,KAAK,Ce12GT,CAAJ,C;QACI,cfy2Gc,sD;Qex2Gd,MAAM,gCAAyB,OAAQ,WAAjC,C,C;;Mfy2GV,O
AAO,kBAAgB,gBAAV,mBAAO,CAAP,IAAU,EAAc,CAAd,CAAhB,C;K;IAGX,kC;Me92GI,IAAI,Efs3GI,KAA
K,Cet3GT,CAAJ,C;QACI,cfq3Gc,sD;Qep3Gd,MAAM,gCAAyB,OAAQ,WAAjC,C,C;;Mfq3GV,OAAO,kBAAgB,gB
AAV,mBAAO,CAAP,IAAU,EAAc,CAAd,CAAhB,C;K;IAGX,kC;Me13GI,IAAI,Efk4GI,KAAK,Cel4GT,CAAJ,C;

QACI,cfi4Gc,sD;Qeh4Gd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Mfi4GV,OAAO,kBAAgB,gBAAV,mBAAO,CAA
P,IAAU,EAAc,CAAd,CAAhB,C;K;gGAGX,yB;MAAA,8D;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAc,wB
AAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAA
Q,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,2C;MAAA,qD;MAAA,uC;QAM
I,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,g
BAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,qD;MA
AA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YA
CI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MA
AA,qD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,
CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;M
AAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,K
AAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;kGAcA,yB;
MAAA,8D;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UA
AU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;
kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,
CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;
O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;
UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf
,OAAO,W;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,oC;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,iBAAc,wBA
Ad,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,EAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ
,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAXX,C;wFAcA,yB;MAAA,+D;MAAA,uC;QAAQiB,Q;QAFb,eAAe,K;QA
Cf,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ
,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAEnB,OAAO,I;O
;KAfX,C;0FAkBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACX,wBAAa,SAAb,
gB;UAAa,WAAA,SAAb,M;UACI,IAAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,C
AAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAEnB,OAAO,I;O;KAfX,C;0FAkBA,yB;MAAA,+D;M
AAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAAb,M;UACI,I
AAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,
C;YACL,WAAW,I;;;QAEnB,OAAO,I;O;KAfX,C;0FAkBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;
QACf,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,QAAJ,C;YACI,IAAK,WAAI,I
AAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAEnB,OAA
O,I;O;KAfX,C;0FAkBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACX,wBAAa,
SAAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,I
AAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAEnB,OAAO,I;O;KAfX,C;0FAkBA,yB;MAAA
,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAAb,M;U
ACI,IAAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,I
AAJ,C;YACL,WAAW,I;;;QAEnB,OAAO,I;O;KAfX,C;0FAkBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAA
e,K;QACf,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,QAAJ,C;YACI,IAAK,WA
AI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAEnB,O
AAO,I;O;KAfX,C;0FAkBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACX,wBA
Aa,SAAb,gB;UAAa,WAAA,SAAb,M;UACI,IAAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAA
U,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAEnB,OAAO,I;O;KAfX,C;0FAkBA,yB;MA
AA,+D;MAAA,oC;MAAA,gC;MAAA,uC;QAQiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACX,wBAAa,SAAb,gB;
UAAa,WAAb,UAAa,SAAb,O;UACI,IAAI,QAAJ,C;YACI,IAAK,WAAI,iBAAJ,C;eACJ,IAAI,CAAC,UAAU,iBA
AV,CAAL,C;YACD,IAAK,WAAI,iBAAJ,C;YACL,WAAW,I;;;QAEnB,OAAO,I;O;KAfX,C;kFAkBA,yB;MAAA,
+D;MAAA,uC;QAMW,kBAAS,gB;QAmgBA,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IAngBU,SAmgBN,CAAU
,OAAV,CAAJ,C;YAAwB,WAAY,WAAI,OAAJ,C;;QAngB1D,OAogBO,W;O;KA1gBX,C;oFASA,yB;MAAA,+D;
MAAA,uC;QAMW,kBAAS,gB;QAogBA,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IApgBa,SAogBT,CAAU,OAA

V,CAAJ,C;YAAwB,WAAY,WAAI,OAAJ,C;;QApgB1D,OAqgBO,W;O;KA3gBX,C;oFASA,yB;MAAA,+D;MAA
A,uC;QAMW,kBAAS,gB;QAqgBA,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IArgBc,SAqgBV,CAAU,OAAV,CA
AJ,C;YAAwB,WAAY,WAAI,OAAJ,C;;QArgB1D,OAsgBO,W;O;KA5gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;
QAMW,kBAAS,gB;QAsgBA,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IAtgBY,SAsgBR,CAAU,OAAV,CAAJ,C;
YAAwB,WAAY,WAAI,OAAJ,C;;QAtgB1D,OAugBO,W;O;KA7gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAM
W,kBAAS,gB;QAugBA,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IAvgBa,SAugBT,CAAU,OAAV,CAAJ,C;YAA
wB,WAAY,WAAI,OAAJ,C;;QAvgB1D,OAwgBO,W;O;KA9gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,k
BAAS,gB;QAwgBA,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IAxgBc,SAwgBV,CAAU,OAAV,CAAJ,C;YAAwB
,WAAY,WAAI,OAAJ,C;;QAxgB1D,OAygBO,W;O;KA/gBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBA
AS,gB;QAygBA,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IAzgBe,SAygBX,CAAU,OAAV,CAAJ,C;YAAwB,WA
AY,WAAI,OAAJ,C;;QAzgB1D,OA0gBO,W;O;KAhhBX,C;oFASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,g
B;QA0gBA,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IA1gBgB,SA0gBZ,CAAU,OAAV,CAAJ,C;YAAwB,WAAY
,WAAI,OAAJ,C;;QA1gB1D,OA2gBO,W;O;KAjhBX,C;oFASA,yB;MAAA,+D;MA2gBA,oC;MAAA,gC;MA3gB
A,uC;QAMW,kBAAS,gB;QA2gBA,Q;QAAhB,iD;UAAgB,cAAhB,0B;UAAsB,IA3gBa,SA2gBT,CAAU,oBAAV,
CAAJ,C;YAAwB,WAAY,WAAI,oBAAJ,C;;QA3gB1D,OA4gBO,W;O;KAlhBX,C;gGASA,yB;MAAA,+D;MAA
A,uC;QAQW,kBAAgB,gB;QA0iTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UAt8SI,IApGmC,SAoG/B,E
As8SkB,cAt8SlB,EAs8SkB,sBAt8SlB,WAs8S2B,IAt8S3B,CAAJ,C;YAA2C,sBAs8SZ,IAt8SY,C;;QApG/C,OAsG
O,W;O;KA9GX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAgB,gB;QAyiTV,gB;QADb,YAAY,C;QACZ,
iD;UAAa,WAAb,e;UAl8SI,IAvGsC,SAuGlC,EAk8SkB,cAl8SlB,EAk8SkB,sBAl8SlB,WAk8S2B,IAl8S3B,CAAJ,
C;YAA2C,sBAk8SZ,IAl8SY,C;;QAvG/C,OAyGO,W;O;KAjHX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,k
BAAgB,gB;QAwiTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA97SI,IA1GuC,SA0GnC,EA87SkB,cA97S
lB,EA87SkB,sBA97SlB,WA87S2B,IA97S3B,CAAJ,C;YAA2C,sBA87SZ,IA97SY,C;;QA1G/C,OA4GO,W;O;KAp
HX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAgB,gB;QAuiTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,W
AAb,e;UA17SI,IA7GqC,SA6GjC,EA07SkB,cA17SlB,EA07SkB,sBA17SlB,WA07S2B,IA17S3B,CAAJ,C;YAA2C
,sBA07SZ,IA17SY,C;;QA7G/C,OA+GO,W;O;KAvHX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAgB,g
B;QAsiTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UAt7SI,IAhHsC,SAgHlC,EAs7SkB,cAt7SlB,EAs7SkB,
sBAt7SlB,WAs7S2B,IAt7S3B,CAAJ,C;YAA2C,sBAs7SZ,IAt7SY,C;;QAhH/C,OAkHO,W;O;KA1HX,C;kGAWA,
yB;MAAA,+D;MAAA,uC;QAQW,kBAAgB,gB;QAqiTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UAl7SI,
IAnHuC,SAmHnC,EAk7SkB,cAl7SlB,EAk7SkB,sBAl7SlB,WAk7S2B,IAl7S3B,CAAJ,C;YAA2C,sBAk7SZ,IAl7
SY,C;;QAnH/C,OAqHO,W;O;KA7HX,C;kGAWA,yB;MAAA,+D;MAAA,uC;QAQW,kBAAgB,gB;QAoiTV,gB;Q
ADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA96SI,IAtHwC,SAsHpC,EA86SkB,cA96SlB,EA86SkB,sBA96SlB,W
A86S2B,IA96S3B,CAAJ,C;YAA2C,sBA86SZ,IA96SY,C;;QAtH/C,OAwHO,W;O;KAhIX,C;kGAWA,yB;MAAA,
+D;MAAA,uC;QAQW,kBAAgB,gB;QAmiTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UA16SI,IAzHyC,S
AyHrC,EA06SkB,cA16SlB,EA06SkB,sBA16SlB,WA06S2B,IA16S3B,CAAJ,C;YAA2C,sBA06SZ,IA16SY,C;;QA
zH/C,OA2HO,W;O;KAnIX,C;kGAWA,yB;MAAA,+D;MA2HA,gC;MAw6SA,oC;MAniTA,uC;QAQW,kBAAgB,
gB;QAkiTV,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,0B;UAAmB,eAAO,cAAP,EAAO,sBAAP,S;UAAAA,cA
AgB,iB;UAt6S/B,IA5HsC,SA4HlC,CAAU,OAAV,EAAiB,OAAjB,CAAJ,C;YAA2C,sBAAI,OAAJ,C;;QA5H/C,O
A8HO,W;O;KAtIX,C;oGAWA,6C;MA+8SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAt8SI,IAAI,WAs
8SkB,cAt8SlB,EAs8SkB,sBAt8SlB,WAs8S2B,IAt8S3B,CAAJ,C;UAA2C,sBAs8SZ,IAt8SY,C;;MAE/C,OAAO,W;
K;qGAGX,6C;MA28SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAl8SI,IAAI,WAk8SkB,cAl8SlB,EAk8
SkB,sBAl8SlB,WAk8S2B,IAl8S3B,CAAJ,C;UAA2C,sBAk8SZ,IAl8SY,C;;MAE/C,OAAO,W;K;sGAGX,6C;MAu
8SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA97SI,IAAI,WA87SkB,cA97SlB,EA87SkB,sBA97SlB,W
A87S2B,IA97S3B,CAAJ,C;UAA2C,sBA87SZ,IA97SY,C;;MAE/C,OAAO,W;K;qGAGX,6C;MAm8SiB,gB;MADb
,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA17SI,IAAI,WA07SkB,cA17SlB,EA07SkB,sBA17SlB,WA07S2B,IA17S3
B,CAAJ,C;UAA2C,sBA07SZ,IA17SY,C;;MAE/C,OAAO,W;K;sGAGX,6C;MA+7SiB,gB;MADb,YAAY,C;MAC
Z,iD;QAAa,WAAb,e;QAt7SI,IAAI,WAs7SkB,cAt7SlB,EAs7SkB,sBAt7SlB,WAs7S2B,IAt7S3B,CAAJ,C;UAA2C
,sBAs7SZ,IAt7SY,C;;MAE/C,OAAO,W;K;sGAGX,6C;MA27SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;
QAl7SI,IAAI,WAk7SkB,cAl7SlB,EAk7SkB,sBAl7SlB,WAk7S2B,IAl7S3B,CAAJ,C;UAA2C,sBAk7SZ,IAl7SY,C

;;MAE/C,OAAO,W;K;sGAGX,6C;MAu7SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA96SI,IAAI,WA8
6SkB,cA96SlB,EA86SkB,sBA96SlB,WA86S2B,IA96S3B,CAAJ,C;UAA2C,sBA86SZ,IA96SY,C;;MAE/C,OAAO,
W;K;sGAGX,6C;MAm7SiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QA16SI,IAAI,WA06SkB,cA16SlB,E
A06SkB,sBA16SlB,WA06S2B,IA16S3B,CAAJ,C;UAA2C,sBA06SZ,IA16SY,C;;MAE/C,OAAO,W;K;sGAGX,yB
;MAAA,gC;MAw6SA,oC;MAx6SA,oD;QA+6SiB,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,0B;UAAmB,eAA
O,cAAP,EAAO,sBAAP,S;UAAA,cAAgB,iB;UAt6S/B,IAAI,UAAU,OAAV,EAAiB,OAAjB,CAAJ,C;YAA2C,sBA
AI,OAAJ,C;;QAE/C,OAAO,W;O;KAXX,C;sGAcA,yB;MAAA,+D;MAAA,sC;QAMW,kBAAmB,gB;QASV,Q;Q
AAhB,iD;UAAgB,cAAhB,e;UAAsB,IAAI,YAAJ,C;YAAkB,WAAY,WAAI,OAAJ,C;;QATpD,OAUO,W;O;KAhB
X,C;0GASA,4C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,YAAJ,C;UA
AkB,WAAY,WAAI,OAAJ,C;;MACpD,OAAO,W;K;wFAGX,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QA
oGH,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IAAI,CApGS,SAoGR,CAAU,OAAV,CAAL,C;YAAyB,WAAY,W
AAI,OAAJ,C;;QApG3D,OAqGO,W;O;KA3GX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QAqG
H,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IAAI,CArGY,SAqGX,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAA
I,OAAJ,C;;QArG3D,OAsGO,W;O;KA5GX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QAsGH,Q;
QAAhB,iD;UAAgB,cAAhB,e;UAAsB,IAAI,CAtGa,SAsGZ,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAI,OA
AJ,C;;QAtG3D,OAuGO,W;O;KA7GX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QAuGH,Q;QAA
hB,iD;UAAgB,cAAhB,e;UAAsB,IAAI,CAvGW,SAuGV,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAI,OAAJ,
C;;QAvG3D,OAwGO,W;O;KA9GX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QAwGH,Q;QAAh
B,iD;UAAgB,cAAhB,e;UAAsB,IAAI,CAxGY,SAwGX,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAI,OAAJ,C;
;QAxG3D,OAyGO,W;O;KA/GX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QAyGH,Q;QAAhB,iD
;UAAgB,cAAhB,e;UAAsB,IAAI,CAzGa,SAyGZ,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAI,OAAJ,C;;QAz
G3D,OA0GO,W;O;KAhHX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QA0GH,Q;QAAhB,iD;UA
AgB,cAAhB,e;UAAsB,IAAI,CA1Gc,SA0Gb,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAI,OAAJ,C;;QA1G3D,
OA2GO,W;O;KAjHX,C;0FASA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAY,gB;QA2GH,Q;QAAhB,iD;UAAgB,c
AAhB,e;UAAsB,IAAI,CA3Ge,SA2Gd,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAI,OAAJ,C;;QA3G3D,OA4
GO,W;O;KAlHX,C;0FASA,yB;MAAA,+D;MA4GA,oC;MAAA,gC;MA5GA,uC;QAMW,kBAAY,gB;QA4GH,Q;
QAAhB,iD;UAAgB,cAAhB,0B;UAAsB,IAAI,CA5GY,SA4GX,CAAU,oBAAV,CAAL,C;YAAyB,WAAY,WAAI,
oBAAJ,C;;QA5G3D,OA6GO,W;O;KAnHX,C;IASA,kC;MAMI,OAAO,2BAAgB,gBAAhB,C;K;IAGX,iD;MAMo
B,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,eAAJ,C;UAAqB,WAAY,WAAI,OA
AJ,C;;MACvD,OAAO,W;K;4FAGX,6C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;Q
AAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,6C;
MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL
,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB
;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAC
3D,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,
CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,6C;MAMoB,Q;
MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,
WAAY,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,c
AAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAC3D,OAAO
,W;K;8FAGX,6C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UA
AU,OAAV,CAAL,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAhB,wB
AAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WA
AI,OAAJ,C;;MAC3D,OAAO,W;K;8FAGX,yB;MAAA,oC;MAAA,gC;MAAA,oD;QAMoB,Q;QAAhB,wBAAgB,
SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UAAsB,IAAI,CAAC,UAAU,oBAAV,CAAL,C;YAAyB,WAAY,
WAAI,oBAAJ,C;;QAC3D,OAAO,W;O;KAPX,C;sFAUA,6C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,
cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;
wFAGX,6C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,C
AAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAhB,wBAAgB,SAA

hB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1
D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,
UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAh
B,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAI
,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,
M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C;M
AMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAw
B,WAAY,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB
,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;
wFAGX,yB;MAAA,oC;MAAA,gC;MAAA,oD;QAMoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAg
B,SAAhB,O;UAAsB,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,WAAY,WAAI,oBAAJ,C;;QAC1D,OAAO,W;O;KA
PX,C;IAUA,mC;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OKvtIe,W;;MLwtItC,OAA4D,OAArD,yBAAY,OAAQ,MA
ApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,CAAqD,C;K;IAGhE,qC;MAII,IAAI,OAAQ,UAAZ,C;QAAu
B,OK/tIe,W;;MLguItC,OgBpsIsC,OhBosI/B,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IA
A3B,CgBpsI+B,C;K;IhBusI1C,qC;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OKvuIe,W;;MLwuItC,OgBpsIuC,OhBosI
hC,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,CgBpsIgC,C;K;IhBusI3C,qC;MAII,I
AAI,OAAQ,UAAZ,C;QAAuB,OK/uIe,W;;MLgvItC,OgBpsIqC,OhBosI9B,yBAAY,OAAQ,MAApB,EAA2B,OAA
Q,aAAR,GAAuB,CAAvB,IAA3B,CgBpsI8B,C;K;IhBusIzC,qC;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OKvvIe,W;;
MLwvItC,OgBpsIsC,OhBosI/B,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,CgBpsI+
B,C;K;IhBusI1C,qC;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OK/vIe,W;;MLgwItC,OgBpsIuC,OhBosIhC,yBAAY,O
AAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,CgBpsIgC,C;K;IhBusI3C,qC;MAII,IAAI,OAAQ,U
AAZ,C;QAAuB,OKvwIe,W;;MLwwItC,OgBpsIwC,OhBosIjC,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,G
AAuB,CAAvB,IAA3B,CgBpsIiC,C;K;IhBusI5C,qC;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OK/wIe,W;;MLgxItC,O
gBpsIyC,OhBosIlC,0BAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,CgBpsIkC,C;K;IhBu
sI7C,qC;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OKvxIe,W;;MLwxItC,OAA4D,SAArD,0BAAY,OAAQ,MAApB,E
AA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,CAAqD,C;K;IAGhE,qC;MAOkB,Q;MAHd,WAAmB,wBAAR,OA
AQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACtB,WAAW,iBAAa,IAAb,C;MACG
,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAK,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,qC;M
AOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;
MACtB,WAAW,iBAAgB,IAAhB,C;MACG,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAK,WAAI,UAAI,KA
AJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MAC
nB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACtB,WAAW,iBAAiB,IAAjB,C;MACG,yB;MAAd,OAAc,cAAd,C;
QAAc,uB;QACV,IAAK,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,
wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACtB,WAAW,iBAAe,IA
Af,C;MACG,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAK,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K
;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAA
e,OAAO,W;MACtB,WAAW,iBAAgB,IAAhB,C;MACG,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAK,WA
AI,UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EA
AxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACtB,WAAW,iBAAiB,IAAjB,C;MACG,yB;MAAd,O
AAc,cAAd,C;QAAc,uB;QACV,IAAK,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MA
Hd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACtB,WAA
W,iBAAkB,IAAlB,C;MACG,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAK,WAAI,UAAI,KAAJ,CAAJ,C;;M
AET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ
,CAAZ,C;QAAe,OAAO,W;MACtB,WAAW,iBAAmB,IAAnB,C;MACG,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QA
CV,IAAK,WAAI,UAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAA
Q,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACtB,WAAW,iBAAgB,IAAhB,C;MAC
G,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAK,WAAI,sBAAI,KAAJ,EAAJ,C;;MAET,OAAO,I;K;IAGX,w
C;MAMwB,UACT,M;MAHX,aAAa,aAAa,SAAb,EAAmB,OAAQ,KAA3B,C;MACb,kBAAkB,C;MACE,yB;MAA

pB,OAAoB,cAApB,C;QAAoB,6B;QAChB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,WAAL,C;;MAE5B,O
AAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,cAAU,OAAQ,KAAlB,C;MACb,kBAAkB,C;MACE,yB;M
AApB,OAAoB,cAApB,C;QAAoB,6B;QAChB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,WAAL,C;;MAE5
B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,eAAW,OAAQ,KAAnB,C;MACb,kBAAkB,C;MACE,
yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QAChB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,WAAL,C;;
MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,eAAS,OAAQ,KAAjB,C;MACb,kBAAkB,C;M
ACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QAChB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,WAAL
,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAU,OAAQ,KAAlB,C;MACb,kBAAkB,
C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QAChB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UAAK,W
AAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAW,OAAQ,KAAnB,C;MACb,kB
AAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QAChB,OAAO,oBAAP,EAAO,4BAAP,YAAwB,UA
AK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAY,OAAQ,KAApB,C;MAC
b,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QAChB,OAAO,oBAAP,EAAO,4BAAP,YAAw
B,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,oBAAa,OAAQ,KArB,C;
MACb,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QAChB,OAAO,oBAAP,EAAO,4BAAP,Y
AAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAMwB,UACT,M;MAHX,aAAa,iBAAU,OAAQ,KAA
lB,C;MACb,kBAAkB,C;MACE,yB;MAApB,OAAoB,cAApB,C;QAAoB,6B;QAChB,OAAO,oBAAP,EAAO,4BA
AP,YAAwB,UAAK,WAAL,C;;MAE5B,OAAO,M;K;IAGX,0C;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,yBA
AY,CAAZ,EAAe,CAAf,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA
3B,C;K;IAGX,0C;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,cAAU,CAAV,C;MAC9B,OAAO,yBAAY,OAAQ,
MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OA
AO,eAAW,CAAX,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;
K;IAGX,2C;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,eAAS,CAAT,C;MAC9B,OAAO,yBAAY,OAAQ,MAA
pB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,i
BAAU,CAAV,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;I
AGX,2C;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,iBAAW,CAAX,C;MAC9B,OAAO,yBAAY,OAAQ,MAAp
B,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,iB
AAY,CAAZ,C;MAC9B,OAAO,yBAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IA
GX,2C;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,oBAAa,CAAb,C;MAC9B,OAAO,0BAAY,OAAQ,MAApB,
EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAGX,2C;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,iBA
AU,CAAV,C;MAC9B,OAAO,0BAAY,OAAQ,MAApB,EAA2B,OAAQ,aAAR,GAAuB,CAAvB,IAA3B,C;K;IAG
X,4B;MAciB,Q;MeloJb,IAAI,Ef4nJI,KAAK,Ce5nJT,CAAJ,C;QACI,cf2nJc,sD;Qe1nJd,MAAM,gCAAyB,OAAQ,
WAAjC,C;;Mf2nJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,iB;
MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW
,iBAAa,CAAb,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mC
AAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;MexpJb,IAAI,EfkpJI,KAAK,CelpJT,CAAJ,C;QA
CI,cfipJc,sD;QehpJd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MfipJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MAC
nB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAA
L,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAgB,CAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAA
b,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q
;Me9qJb,IAAI,EfwqJI,KAAK,CexqJT,CAAJ,C;QACI,cfuqJc,sD;QetqJd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Mfu
qJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACtB,IAAI,
MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAiB,CA
AjB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CA
Af,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;MepsJb,IAAI,Ef8rJI,KAAK,Ce9rJT,CAAJ,C;QACI,cf6rJc,
sD;Qe5rJd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Mf6rJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,
KAAK,gBAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,
C;MACnB,YAAY,C;MACZ,WAAW,iBAAe,CAAf,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,I

AAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;Me1tJb,IA
AI,EfotJI,KAAK,CeptJT,CAAJ,C;QACI,cfmtJc,sD;QeltJd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MfmtJV,IAAI,M
AAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAA
T,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAgB,CAAhB,C;MAC
X,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,
K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;MehvJb,IAAI,Ef0uJI,KAAK,Ce1uJT,CAAJ,C;QACI,cfyuJc,sD;QexuJd
,MAAM,gCAAyB,OAAQ,WAAjC,C;;MfyuJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gB
AAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACn
B,YAAY,C;MACZ,WAAW,iBAAiB,CAAjB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,
WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;MetwJb,IAAI,Ef
gwJI,KAAK,CehwJT,CAAJ,C;QACI,cf+vJc,sD;Qe9vJd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Mf+vJV,IAAI,MAA
K,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;
QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAkB,CAAlB,C;MACX,wB
AAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;M
AER,OAAO,I;K;IAGX,8B;MAciB,Q;Me5xJb,IAAI,EfsxJI,KAAK,CetxJT,CAAJ,C;QACI,cfqxJc,sD;QepxJd,MAA
M,gCAAyB,OAAQ,WAAjC,C;;MfqxJV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,
C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,CAAL,CAAP,C;MACnB,YA
AY,C;MACZ,WAAW,iBAAmB,CAAnB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,IAAK,WA
AI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,8B;MAciB,Q;MelzJb,IAAI,Ef4yJI,
KAAK,Ce5yJT,CAAJ,C;QACI,cf2yJc,sD;Qe1yJd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Mf2yJV,IAAI,MAAK,CA
AT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,gBAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAA
Y,OAAO,OAAO,sBAAK,CAAL,EAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAgB,CAAhB,C;MACX,wBAA
a,SAAb,gB;QAAa,WAAb,UAAa,SAAb,O;QACI,IAAK,WAAI,iBAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;
;MAER,OAAO,I;K;IAGX,gC;Me1zJI,IAAI,Efk0JI,KAAK,Cel0JT,CAAJ,C;QACI,cfi0Jc,sD;Qeh0Jd,MAAM,gCA
AyB,OAAQ,WAAjC,C;;Mfi0JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KA
AK,IAAT,C;QAAe,OAAO,iB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,
CAAP,C;MACnB,WAAW,iBAAa,CAAb,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,
WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Me70JI,IAAI,Efq1JI,KAAK,Cer1JT,CAAJ,C;QACI,c
fo1Jc,sD;Qen1Jd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Mfo1JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB
,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OA
AO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAgB,CAAhB,C;MACX,iBAAc,OAAO,CAAP,IA
Ad,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Meh2JI,IAAI,Ef
w2JI,KAAK,Cex2JT,CAAJ,C;QACI,cfu2Jc,sD;Qet2Jd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Mfu2JV,IAAI,MAA
K,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACtB,IAAI
,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAiB,CAAjB,
C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OA
AO,I;K;IAGX,kC;Men3JI,IAAI,Ef23JI,KAAK,Ce33JT,CAAJ,C;QACI,cf03Jc,sD;Qez3Jd,MAAM,gCAAyB,OAA
Q,WAAjC,C;;Mf03JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,
C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C
;MACnB,WAAW,iBAAe,CAAf,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,U
AAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Met4JI,IAAI,Ef84JI,KAAK,Ce94JT,CAAJ,C;QACI,cf64Jc,sD;
Qe54Jd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Mf64JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,
gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAA
K,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAgB,CAAhB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6
B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mez5JI,IAAI,Efi6JI,KAAK
,Cej6JT,CAAJ,C;QACI,cfg6Jc,sD;Qe/5Jd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Mfg6JV,IAAI,MAAK,CAAT,C;Q
AAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAA
T,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAiB,CAAjB,C;MACX,iB

AAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAG
X,kC;Me56JI,IAAI,Efo7JI,KAAK,Cep7JT,CAAJ,C;QACI,cfm7Jc,sD;Qel7Jd,MAAM,gCAAyB,OAAQ,WAAjC,C;
;Mfm7JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OA
AO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,W
AAW,iBAAkB,CAAlB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,UAAK,KA
AL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Me/7JI,IAAI,Efu8JI,KAAK,Cev8JT,CAAJ,C;QACI,cfs8Jc,sD;Qer8Jd,M
AAM,gCAAyB,OAAQ,WAAjC,C;;Mfs8JV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,gB;MAC
X,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,UAAK,OAAO,
CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAmB,CAAnB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B
,U;QACI,IAAK,WAAI,UAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,kC;Mel9JI,IAAI,Ef09JI,KAAK,Ce19JT,
CAAJ,C;QACI,cfy9Jc,sD;Qex9Jd,MAAM,gCAAyB,OAAQ,WAAjC,C;;Mfy9JV,IAAI,MAAK,CAAT,C;QAAY,O
AAO,W;MACnB,WAAW,gB;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;Q
AAY,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,EAAP,C;MACnB,WAAW,iBAAgB,CAAhB,C;MACX,iBAAc,
OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,EAAJ,C;MACT,OAAO,I;K;gGAGX,
yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,
UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAX
X,C;kGAcA,yB;MAAA,8D;MAAA,2C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,I
AAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO
,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/
B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;Q
AGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WA
A+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,
IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,
wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,Q
AAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,gD;MAAA,uC;Q
AMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C;YACI,OAA
O,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;MAAA,gD;
MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CAAV,CAAL,C
;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8D;MAAA,4C;
MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,UAAK,KAAL,CA
AV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAXX,C;kGAcA,yB;MAAA,8
D;MAAA,oC;MAAA,4C;MAAA,gD;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,U
AAU,sBAAK,KAAL,EAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAXX
,C;wFAcA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SA
Ab,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KAZX
,C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAA
b,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KAZX,
C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAA
b,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KAZX,
C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAA
b,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KAZX,
C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAA
b,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KAZX,
C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAA
b,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KAZX,
C;0FAeA,yB;MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,SAAb,gB;UAAa,WAAA,SAA

b,M;UACI,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KAZX,
C;0FAeA,yB;MAAA,+D;MAAA,oC;MAAA,gC;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACX,wBAAa,SAAb,
gB;UAAa,WAAb,UAAa,SAAb,O;UACI,IAAI,CAAC,UAAU,iBAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,iBA
AJ,C;;QAET,OAAO,I;O;KAZX,C;IAeA,4B;MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI
,WAAW,CAAf,C;QAAkB,M;MAClB,mBAAmB,wB;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,U
AAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAI
R,8B;MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAAkB,M;MAClB,
mBAAmB,0B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,I
AAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,CAAC,mBAAO,CAA
P,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAAkB,M;MAClB,mBAAmB,0B;MACnB,iBAAc,CAAd,W
AAiB,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YA
AL,IAAqB,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WA
AW,CAAf,C;QAAkB,M;MAClB,mBAAmB,0B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,
KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,8B;
MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAAkB,M;MAClB,mBA
AmB,0B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,
UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,CAAC,mBAAO,CAAP,IA
AD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAAkB,M;MAClB,mBAAmB,0B;MACnB,iBAAc,CAAd,WAAi
B,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,I
AAqB,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,
CAAf,C;QAAkB,M;MAClB,mBAAmB,0B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,KAA
L,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,8B;MAII,
eAAe,CAAC,mBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAAkB,M;MAClB,mBAAmB,
0B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK
,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,8B;MAII,eAAe,CAAC,mBAAO,CAAP,IAAD,IA
Aa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAAkB,M;MAClB,mBAAmB,0B;MACnB,iBAAc,CAAd,WAAiB,QA
AjB,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB
,G;QACrB,mC;;K;IAIR,kD;MAWI,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAe,C
AAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAA
U,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc
,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,kD;MAWI,oCAAa,2BAAkB,SAAlB,EAA
6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,Q
AAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU
,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;I
AIR,mD;MAWI,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAA
Z,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB
,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;Q
ACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,mD;MAWI,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC
,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M
;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;Q
ACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,mD;MAWI,oCA
Aa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CA
AxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA
8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,
IAAqB,G;QACrB,mC;;K;IAIR,mD;MAWI,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,
eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAA
L,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,mD;MAWI,oCAAa,2BAAkB,SAA

lB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,
cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QAC
I,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,
mC;;K;IAIR,mD;MAWI,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAe,CAAC,YAA
Y,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;
MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,Y
AAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,mD;MAWI,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7
B,EAAsC,gBAAtC,C;MACb,eAAe,CAAC,YAAY,OAAZ,IAAD,IAAwB,CAAxB,I;MACf,IAAI,cAAa,QAAjB,C;
QAA2B,M;MAC3B,mBAAmB,UAAU,CAAV,I;MACnB,iBAAc,SAAd,UAA8B,QAA9B,U;QACI,UAAU,UAAK,
KAAL,C;QACV,UAAK,KAAL,IAAc,UAAK,YAAL,C;QACd,UAAK,YAAL,IAAqB,G;QACrB,mC;;K;IAIR,6B;
MAII,IA+nEO,qBAAQ,CA/nEf,C;QAAe,OAAO,W;MACtB,WAAW,wB;MACN,WAAL,IAAK,C;MACL,OAAO,
I;K;IAGX,+B;MAII,IA6nEO,qBAAQ,CA7nEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;
MACL,OAAO,I;K;IAGX,+B;MAII,IA2nEO,qBAAQ,CA3nEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,W
AAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAynEO,qBAAQ,CAznEf,C;QAAe,OAAO,W;MACtB,WAAW,
0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAunEO,qBAAQ,CAvnEf,C;QAAe,OAAO,W;M
ACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAqnEO,qBAAQ,CArnEf,C;QAAe
,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAmnEO,qBAAQ,C
AnnEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+B;MAII,IAin
EO,qBAAQ,CAjnEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,+
B;MAII,IA+mEO,qBAAQ,CA/mEf,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OA
AO,I;K;IAGX,kC;MAII,IAqiEO,qBAAQ,CAriEf,C;QAAe,OAAO,S;MACtB,aAAa,aAAa,SAAb,EAAmB,gBAAn
B,C;MACb,gBAAgB,wB;MAChB,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UA
AK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAiiEO,qBAAQ,CAjiEf,C;QAAe,OAAO,S;MACtB,aAAa,cA
AU,gBAAV,C;MACb,gBAAgB,0B;MAChB,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IA
AwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IA6hEO,qBAAQ,CA7hEf,C;QAAe,OAAO,S;MACt
B,aAAa,eAAW,gBAAX,C;MACb,gBAAgB,0B;MAChB,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CA
AZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAyhEO,qBAAQ,CAzhEf,C;QAAe,OA
AO,S;MACtB,aAAa,eAAS,gBAAT,C;MACb,gBAAgB,0B;MAChB,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,
YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAqhEO,qBAAQ,CArhEf,C;
QAAe,OAAO,S;MACtB,aAAa,iBAAU,gBAAV,C;MACb,gBAAgB,0B;MAChB,aAAU,CAAV,OAAa,SAAb,M;Q
ACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IAihEO,qBAAQ
,CAjhEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAAW,gBAAX,C;MACb,gBAAgB,0B;MAChB,aAAU,CAAV,OAAa,
SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,oC;MAII,IA6g
EO,qBAAQ,CA7gEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAAY,gBAAZ,C;MACb,gBAAgB,0B;MAChB,aAAU,C
AAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,M;K;IAGX,o
C;MAII,IAygEO,qBAAQ,CAzgEf,C;QAAe,OAAO,S;MACtB,aAAa,oBAAa,gBAAb,C;MACb,gBAAgB,0B;MAC
hB,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MAC5B,OAAO,
M;K;IAGX,oC;MAII,IAqgEO,qBAAQ,CArgEf,C;QAAe,OAAO,S;MACtB,aAAa,iBAAU,gBAAV,C;MACb,gBA
AgB,0B;MAChB,aAAU,CAAV,OAAa,SAAb,M;QACI,OAAO,YAAY,CAAZ,IAAP,IAAwB,UAAK,CAAL,C;MA
C5B,OAAO,M;K;IAGX,4B;MAKI,qBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,qBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,s
BAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,
sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAKI,sBAAQ,4BAAR,C;K;IAGJ,8B;MAK
I,sBAAQ,4BAAR,C;K;IAGJ,sC;MAOI,aAAU,wBAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,C
AAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAA
L,IAAU,I;;K;IAIlB,sC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAA
R,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;
;K;IAIlB,uC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf
,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIlB,u

C;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,
UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIlB,uC;MAOI
,aAAU,0BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,C
AAL,C;QACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIlB,uC;MAOI,aAAU,0
BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;Q
ACX,UAAK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIlB,uC;MAOI,aAAU,0BAAV,O
AA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UA
AK,CAAL,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIlB,uC;MAOI,aAAU,0BAAV,OAA2B,CA
A3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAA
L,IAAU,UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;IAIlB,uC;MAOI,aAAU,0BAAV,OAA2B,CAA3B,M;
QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,UAAK,CAAL,C;QACX,UAAK,CAAL,IAAU,
UAAK,CAAL,C;QACV,UAAK,CAAL,IAAU,I;;K;kFAIlB,yB;MAAA,oD;MiB15LA,sC;MAAA,oC;MAAA,uBA
Oe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA
2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjBm5Lf,sC;QAMI,IAAI,mBAAO,CAAX,C;UAAc,oBiBz5Ld,eA
AW,iBjBy5LsB,QiBz5LtB,CAAX,CjBy5Lc,C;;O;KANlB,C;sGASA,yB;MAAA,oD;MiBh5LA,sC;MAAA,oC;MA
AA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAA
d,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjBy4Lf,sC;QAMI,IAAI,mBAAO,CAAX,C;UAAc,oBiB/4L
d,eAAW,2BjB+4LgC,QiB/4LhC,CAAX,CjB+4Lc,C;;O;KANlB,C;IASA,mC;MAMI,oBAAS,cAAT,C;K;IAGJ,qC;
MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;
K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,i
B;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CA
AX,C;QACI,e;QACA,oB;;K;IAIR,qC;MAII,IAAI,mBAAO,CAAX,C;QACI,e;QACA,oB;;K;IAIR,2B;MAMI,OAA
qB,OAAd,sBAAc,C;K;IAGzB,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OkBnhMhC,WlBmhMgC,C;
K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OkB1hMhC,WlB0hMgC,C;K;IAG3C,6B;MAI0B,
kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OkBjiMhC,WlBiiMgC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB
;MAA9B,OAAuC,OkBxiMhC,WlBwiMgC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,Ok
B/iMhC,WlB+iMgC,C;K;IAG3C,6B;MAI0B,kBAAf,yB;MAAuB,mB;MAA9B,OAAuC,OkBtjMhC,WlBsjMgC,C;
K;IAG3C,6B;MAI0B,kBAAf,0B;MAAuB,mB;MAA9B,OAAuC,OkB7jMhC,WlB6jMgC,C;K;IAG3C,gC;MAMI,I
A6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SgB3jKiB,Q;MhB2jKK,mB;MAA7B,OkBvkMO,W;
K;IlB0kMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBzjKiB,Q;MhByjKK,iB;M
AA7B,OkB/kMO,W;K;IlBklMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBvjKi
B,Q;MhBujKK,iB;MAA7B,OkBvlMO,W;K;IlB0lMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MA
CD,kBAAd,SgBrjKiB,Q;MhBqjKK,iB;MAA7B,OkB/lMO,W;K;IlBkmMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;
QAAe,OAAO,S;MACD,kBAAT,UAAL,SAAK,C;MAAiB,mB;MAA7B,OkBvmMO,W;K;IlB0mMX,kC;MAII,IA6
kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBljKiB,Q;MhBkjKK,iB;MAA7B,OkB/mMO,W;K;Il
BknMX,kC;MAII,IA6kDO,qBAAQ,CA7kDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBhjKiB,Q;MhBgjKK,iB;MAA
7B,OkBvnMO,W;K;IlB0nMX,kC;MAII,IAqlDO,qBAAQ,CArlDf,C;QAAe,OAAO,S;MACD,kBAAT,UAAL,SAA
K,C;MAAiB,iB;MAA7B,OkB/nMO,W;K;IlBkoMX,0C;MAMI,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAAO,S;MA
CD,kBAAd,SgB7nKiB,Q;MhB6nKK,sBAAS,cAAT,C;MAA7B,OkBzoMO,W;K;IlB4oMX,4C;MAII,IA2gDO,qB
AAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kBAAd,SgB3nKiB,Q;MhB2nKK,6B;MAA7B,OkBjpMO,W;K;IlBopMX
,4C;MAII,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBznKiB,Q;MhBynKK,6B;MAA7B,Ok
BzpMO,W;K;IlB4pMX,4C;MAII,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBvnKiB,Q;Mh
BunKK,6B;MAA7B,OkBjqMO,W;K;IlBoqMX,4C;MAII,IA2gDO,qBAAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kB
AAT,UAAL,SAAK,C;MAAiB,6B;MAA7B,OkBzqMO,W;K;IlB4qMX,4C;MAII,IA2gDO,qBAAQ,CA3gDf,C;QA
Ae,OAAO,S;MACD,kBAAd,SgBpnKiB,Q;MhBonKK,6B;MAA7B,OkBjrMO,W;K;IlBorMX,4C;MAII,IA2gDO,q
BAAQ,CA3gDf,C;QAAe,OAAO,S;MACD,kBAAd,SgBlnKiB,Q;MhBknKK,6B;MAA7B,OkBzrMO,W;K;IlB4rM
X,4C;MAII,IAmhDO,qBAAQ,CAnhDf,C;QAAe,OAAO,S;MACD,kBAAT,UAAL,SAAK,C;MAAiB,6B;MAA7B,
OkBjsMO,W;K;IlBosMX,gD;MAMI,IAy8CO,qBAAQ,CAz8Cf,C;QAAe,OAAO,S;MACD,kBAAd,SgB/rKiB,Q;M

hB+rKK,iC;MAA7B,OkB3sMO,W;K;sFlB8sMX,yB;MAAA,wD;MiBnsMA,sC;MAAA,oC;MAAA,uBAOe,yB;Q
ArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4D
M,CA5DN,CAA3B,C;W;S;OA4DI,C;MjB4rMf,sC;QAQI,OAAO,sBiBpsMP,eAAW,iBjBosMiB,QiBpsMjB,CAA
X,CjBosMO,C;O;KARX,C;wFAWA,yB;MAAA,wD;MiB9sMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAq
Ee,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,C
AA3B,C;W;S;OA4DI,C;MjBusMf,sC;QAMI,OAAO,sBiB7sMP,eAAW,iBjB6sMiB,QiB7sMjB,CAAX,CjB6sMO,
C;O;KANX,C;wFASA,yB;MAAA,wD;MiBvtMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,
uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;
OA4DI,C;MjBgtMf,sC;QAMI,OAAO,sBiBttMP,eAAW,iBjBstMiB,QiBttMjB,CAAX,CjBstMO,C;O;KANX,C;wF
ASA,yB;MAAA,wD;MiBhuMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAs
B,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjByt
Mf,sC;QAMI,OAAO,sBiB/tMP,eAAW,iBjB+tMiB,QiB/tMjB,CAAX,CjB+tMO,C;O;KANX,C;wFASA,yB;MAAA
,wD;MiBzuMA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,O
A5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjBkuMf,sC;QAMI,O
AAO,sBiBxuMP,eAAW,iBjBwuMiB,QiBxuMjB,CAAX,CjBwuMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiBlv
MA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,
SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjB2uMf,sC;QAMI,OAAO,sBiBjv
MP,eAAW,iBjBivMiB,QiBjvMjB,CAAX,CjBivMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiB3vMA,sC;MAAA,
oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5D
hB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjBovMf,sC;QAMI,OAAO,sBiB1vMP,eAAW,iBj
B0vMiB,QiB1vMjB,CAAX,CjB0vMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiBpwMA,sC;MAAA,oC;MAAA,
uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,E
AA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjB6vMf,sC;QAMI,OAAO,sBiBnwMP,eAAW,iBjBmwMiB,Q
iBnwMjB,CAAX,CjBmwMO,C;O;KANX,C;wFASA,yB;MAAA,wD;MiB7wMA,sC;MAAA,oC;MAAA,uBAOe,y
B;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,S
A4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MjBswMf,sC;QAMI,OAAO,sBiB5wMP,eAAW,iBjB4wMiB,QiB5wMj
B,CAAX,CjB4wMO,C;O;KANX,C;0GASA,yB;MAAA,wD;MiBnwMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,
8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/
EN,CAA3B,C;W;S;OA+EI,C;MjB4vMf,sC;QAMI,OAAO,sBiBlwMP,eAAW,2BjBkwM2B,QiBlwM3B,CAAX,Cj
BkwMO,C;O;KANX,C;4GASA,yB;MAAA,wD;MiB5wMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4
B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,
C;W;S;OA+EI,C;MjBqwMf,sC;QAII,OAAO,sBiBzwMP,eAAW,2BjBywM2B,QiBzwM3B,CAAX,CjBywMO,C;O
;KAJX,C;4GAOA,yB;MAAA,wD;MiBnxMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;
YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+E
I,C;MjB4wMf,sC;QAII,OAAO,sBiBhxMP,eAAW,2BjBgxM2B,QiBhxM3B,CAAX,CjBgxMO,C;O;KAJX,C;4GA
OA,yB;MAAA,wD;MiB1xMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB
,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjBmxMf,
sC;QAII,OAAO,sBiBvxMP,eAAW,2BjBuxM2B,QiBvxM3B,CAAX,CjBuxMO,C;O;KAJX,C;4GAOA,yB;MAAA,
wD;MiBjyMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/
Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjB0xMf,sC;QAII,OAAO,
sBiB9xMP,eAAW,2BjB8xM2B,QiB9xM3B,CAAX,CjB8xMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiBxyMA,s
C;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+E
gB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjBiyMf,sC;QAII,OAAO,sBiBryMP,eAA
W,2BjBqyM2B,QiBryM3B,CAAX,CjBqyMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiB/yMA,sC;MAAA,oC;M
AAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CA
Ad,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjBwyMf,sC;QAII,OAAO,sBiB5yMP,eAAW,2BjB4yM2B
,QiB5yM3B,CAAX,CjB4yMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiBtzMA,sC;MAAA,oC;MAAA,iCAOe,yB
;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+

EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MjB+yMf,sC;QAII,OAAO,sBiBnzMP,eAAW,2BjBmzM2B,QiBnzM3B,CA
AX,CjBmzMO,C;O;KAJX,C;4GAOA,yB;MAAA,wD;MiB7zMA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eA
wFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,C
AA3B,C;W;S;OA+EI,C;MjBszMf,sC;QAII,OAAO,sBiB1zMP,eAAW,2BjB0zM2B,QiB1zM3B,CAAX,CjB0zMO,
C;O;KAJX,C;IAOA,qC;MAMI,OAAO,sBAAW,cAAX,C;K;IAGX,uC;MAIoB,kBgBz1KQ,iB;MhBy1KA,iB;MAA
xB,OAAiC,WkB/2M1B,WlB+2M0B,C;K;IAGrC,uC;MAIoB,kBgBt1KQ,iB;MhBs1KA,iB;MAAxB,OAAiC,WkBt
3M1B,WlBs3M0B,C;K;IAGrC,uC;MAIoB,kBgBn1KQ,iB;MhBm1KA,iB;MAAxB,OAAiC,WkB73M1B,WlB63M
0B,C;K;IAGrC,uC;MAIoB,kBAAT,oB;MAAiB,mB;MAAxB,OAAiC,WkBp4M1B,WlBo4M0B,C;K;IAGrC,uC;M
AIoB,kBgB90KQ,iB;MhB80KA,iB;MAAxB,OAAiC,WkB34M1B,WlB24M0B,C;K;IAGrC,uC;MAIoB,kBgB30K
Q,iB;MhB20KA,iB;MAAxB,OAAiC,WkBl5M1B,WlBk5M0B,C;K;IAGrC,uC;MAIoB,kBAAT,oB;MAAiB,iB;MA
AxB,OAAiC,WkBz5M1B,WlBy5M0B,C;K;IAGrC,2C;MAMI,OAAmC,OAA5B,2BAAgB,UAAhB,CAA4B,C;K;I
AGvC,6C;MAI0B,kBAAf,yB;MAAuB,iC;MAA9B,OAAqD,OkBz6M9C,WlBy6M8C,C;K;IAGzD,6C;MAI0B,kB
AAf,yB;MAAuB,iC;MAA9B,OAAqD,OkBh7M9C,WlBg7M8C,C;K;IAGzD,6C;MAI0B,kBAAf,yB;MAAuB,iC;M
AA9B,OAAqD,OkBv7M9C,WlBu7M8C,C;K;IAGzD,6C;MAI0B,kBAAf,yB;MAAuB,iC;MAA9B,OAAqD,OkB97
M9C,WlB87M8C,C;K;IAGzD,6C;MAI0B,kBAAf,yB;MAAuB,iC;MAA9B,OAAqD,OkBr8M9C,WlBq8M8C,C;K;
IAGzD,6C;MAI0B,kBAAf,yB;MAAuB,iC;MAA9B,OAAqD,OkB58M9C,WlB48M8C,C;K;IAGzD,6C;MAI0B,kB
AAf,yB;MAAuB,iC;MAA9B,OAAqD,OkBn9M9C,WlBm9M8C,C;K;IAGzD,6C;MAI0B,kBAAf,0B;MAAuB,iC;
MAA9B,OAAqD,OkB19M9C,WlB09M8C,C;K;IAkoCrD,gC;MAAQ,oBAAS,CAAT,EAAY,wBAAZ,C;K;IAMR,
kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;
MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;MA
AQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;MAAQ,
oBAAS,CAAT,EAAY,0BAAZ,C;K;IAMR,kC;MAAQ,oBAAS,CAAT,EAAY,0BAAZ,C;K;oFAEZ,qB;MAKI,OA
AO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;
MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAAQ,C;K;sF
AGnB,qB;MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAAQ,C;K;sFAGnB,qB;MAKI,OAAO,qBAA
Q,C;K;0FAGnB,qB;MAKI,OAAO,EAxEA,qBAAQ,CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEA,qBAAQ,CA
wER,C;K;4FAGX,qB;MAKI,OAAO,EAxEA,qBAAQ,CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEA,qBAAQ,C
AwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEA,qBAAQ,CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEA,qBAAQ,
CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEA,qBAAQ,CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEA,qBAA
Q,CAwER,C;K;4FAGX,qB;MAKI,OAAO,EAxEA,qBAAQ,CAwER,C;K;IAOP,kC;MAAQ,0BAAO,CAAP,I;K;IA
MR,oC;MAAQ,0BAAO,CAAP,I;K;IAMR,oC;MAAQ,0BAAO,CAAP,I;K;IAMR,oC;MAAQ,0BAAO,CAAP,I;K;I
AMR,oC;MAAQ,0BAAO,CAAP,I;K;IAMR,oC;MAAQ,0BAAO,CAAP,I;K;IAMR,oC;MAAQ,0BAAO,CAAP,I;K
;IAMR,oC;MAAQ,0BAAO,CAAP,I;K;IAMR,oC;MAAQ,0BAAO,CAAP,I;K;IA8TZ,yD;MAcI,sBAAS,cAAT,EA
AyB,SAAzB,EAAoC,OAApC,C;K;IAGJ,yD;MAYI,mBAAK,SAAL,EAAgB,OAAhB,C;MACA,qBAAQ,SAAR,E
AAmB,OAAnB,C;K;IAGJ,yD;MAYI,mBAAK,SAAL,EAAgB,OAAhB,C;MACA,sBAAQ,SAAR,EAAmB,OAAn
B,C;K;IAGJ,0D;MAYI,mBAAK,SAAL,EAAgB,OAAhB,C;MACA,sBAAQ,SAAR,EAAmB,OAAnB,C;K;IAGJ,0
D;MAYI,mBAAK,SAAL,EAAgB,OAAhB,C;MACA,sBAAQ,SAAR,EAAmB,OAAnB,C;K;IAGJ,0D;MAYI,mBA
AK,SAAL,EAAgB,OAAhB,C;MACA,sBAAQ,SAAR,EAAmB,OAAnB,C;K;IAGJ,0D;MAYI,mBAAK,SAAL,EA
AgB,OAAhB,C;MACA,sBAAQ,SAAR,EAAmB,OAAnB,C;K;IAGJ,0D;MAYI,mBAAK,SAAL,EAAgB,OAAhB,
C;MACA,sBAAQ,SAAR,EAAmB,OAAnB,C;K;IA2B0B,oD;MAAA,wB;QAAW,2BAAK,KAAL,C;O;K;IAJzC,m
C;MAII,OAAO,qBAAa,gBAAb,EAAmB,gCAAnB,C;K;IAOgB,8C;MAAA,wB;QAAW,wBAAK,KAAL,C;O;K;I
AJtC,gC;MAII,OAAO,+BAAU,gBAAV,GAAgB,6BAAhB,C;K;IAOgB,8C;MAAA,wB;QAAW,wBAAK,KAAL,C
;O;K;IAJtC,gC;MAII,OAAO,kBAAU,gBAAV,EAAgB,6BAAhB,C;K;IAOkB,kD;MAAA,wB;QAAW,0BAAK,KA
AL,C;O;K;IAJxC,kC;MAII,OAAO,kCAAY,gBAAZ,GAAkB,+BAAlB,C;K;IAOiB,gD;MAAA,wB;QAAW,yBAA
K,KAAL,C;O;K;IAJvC,iC;MAII,OAAO,kCAAW,gBAAX,GAAiB,8BAAjB,C;K;IAOe,4C;MAAA,wB;QAAW,uB
AAK,KAAL,C;O;K;IAJrC,+B;MAII,OAAO,gCAAS,gBAAT,GAAe,4BAAf,C;K;IAOgB,8C;MAAA,wB;QAAW,
wBAAK,KAAL,C;O;K;IAJtC,gC;MAII,OAAO,kBAAU,gBAAV,EAAgB,6BAAhB,C;K;IAOiB,gD;MAAA,wB;Q
AAW,yBAAK,KAAL,C;O;K;IAJvC,iC;MAII,OAAO,gCAAW,gBAAX,GAAiB,8BAAjB,C;K;wFA2CX,yB;MAA

A,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,k
BAAY,mBAAoB,QAApB,C;QAyqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,OAA
V,C;UM3+QnB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;QNi0PA,OA4qBO,W;O;KAxrBX,C;0FAeA,yB;
MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QA
C1B,kBAAY,mBAAoB,QAApB,C;QAyqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,CAAU,
OAAV,C;UM1/QnB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;QNg1PA,OA4qBO,W;O;KAxrBX,C;0FAe
A,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,
C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAyqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0qB/B,C
AAU,OAAV,C;UMzgRnB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;QN+1PA,OA4qBO,W;O;KAxrBX,C;
0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,E
AAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAyqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C,SA0q
B/B,CAAU,OAAV,C;UMxhRnB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;QN82PA,OA4qBO,W;O;KAxr
BX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAlB,YAAY,gBAAZ,CAAkB,E
AAc,EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAyqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1qB8C
,SA0qB/B,CAAU,OAAV,C;UMviRnB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;QN63PA,OA4qBO,W;O;
KAxrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAlB,YAAY,gBAAZ,CAA
kB,EAAc,EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAyqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA1q
B8C,SA0qB/B,CAAU,OAAV,C;UMtjRnB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;QN44PA,OA4qBO,
W;O;KAxrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAlB,YAAY,gBAAZ,
CAAkB,EAAc,EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAyqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,
WA1qB8C,SA0qB/B,CAAU,OAAV,C;UMrkRnB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;QN25PA,OA4
qBO,W;O;KAxrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAiC,cAAlB,YAAY,gB
AAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAyqBH,Q;QAAhB,iD;UAAgB,cAAhB,e;U
ACI,WA1qB8C,SA0qB/B,CAAU,OAAV,C;UMplRnB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;QN06PA,
OA4qBO,W;O;KAxrBX,C;0FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MA4qBA,oC;MAAA,gC;MA5qBA,uC;
QAWI,eAAiC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBAAY,mBAAoB,QAApB,C;QAyqBH,Q
;QAAhB,iD;UAAgB,cAAhB,0B;UACI,WA1qB8C,SA0qB/B,CAAU,oBAAV,C;UMnmRnB,wBAAI,IAAK,MAAT
,EAAgB,IAAK,OAArB,C;;QNy7PA,OA4qBO,W;O;KAxrBX,C;4FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;M
AAA,yC;QAWI,eAAiC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBAAc,mBAAoB,QAApB,C;QA
mQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAY,aApQoC,WAoQhC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C
;;QApQhB,OAsQO,W;O;KAlRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAl
B,YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBAAc,mBAAuB,QAAvB,C;QAoQL,Q;QAAhB,iD;UAAgB,c
AAhB,e;UACI,WAAY,aArQuC,WAqQnC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QArQhB,OAuQO,W;O;KAn
RX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAlB,YAAY,gBAAZ,CAAkB,E
AAc,EAAd,C;QAC1B,kBAAc,mBAAwB,QAAxB,C;QAqQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAY,aAt
QwC,WAsQpC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QAtQhB,OAwQO,W;O;KApRX,C;8FAeA,yB;MAAA,0
D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBA
Ac,mBAAsB,QAAtB,C;QAsQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAY,aAvQsC,WAuQlC,CAAY,OAAZ,
CAAJ,EAA0B,OAA1B,C;;QAvQhB,OAyQO,W;O;KArRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAA
A,yC;QAWI,eAAiC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBAAc,mBAAuB,QAAvB,C;QAuQ
L,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAY,aAxQuC,WAwQnC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;Q
AxQhB,OA0QO,W;O;KAtRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAlB,
YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBAAc,mBAAwB,QAAxB,C;QAwQL,Q;QAAhB,iD;UAAgB,c
AAhB,e;UACI,WAAY,aAzQwC,WAyQpC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QAzQhB,OA2QO,W;O;KA
vRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAlB,YAAY,gBAAZ,CAAkB,E
AAc,EAAd,C;QAC1B,kBAAc,mBAAyB,QAAzB,C;QAyQL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAY,aA1
QyC,WA0QrC,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QA1QhB,OA4QO,W;O;KAxRX,C;8FAeA,yB;MAAA,0
D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eAAiC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBA

Ac,mBAA0B,QAA1B,C;QA0QL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAY,aA3Q0C,WA2QtC,CAAY,OAA
Z,CAAJ,EAA0B,OAA1B,C;;QA3QhB,OA6QO,W;O;KAzRX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;M
A6QA,oC;MAAA,gC;MA7QA,yC;QAWI,eAAiC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBAAc
,mBAAuB,QAAvB,C;QA2QL,Q;QAAhB,iD;UAAgB,cAAhB,0B;UACI,WAAY,aA5QuC,WA4QnC,CAAY,oBAA
Z,CAAJ,EAA0B,oBAA1B,C;;QA5QhB,OA8QO,W;O;KA1RX,C;8FAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;M
AAA,yD;QAUI,eAAiC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBAAc,mBAAoB,QAApB,C;QA
6QL,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAY,aA9QoC,WA8QhC,CAAY,OAAZ,CAAJ,EA9QiD,cA8QvB,
CAAe,OAAf,CAA1B,C;;QA9QhB,OAgRO,W;O;KA3RX,C;8FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA
,yD;QAUI,eAAiC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBAAc,mBAAoB,QAApB,C;QA+QL,
Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAAY,aAhRoC,WAgRhC,CAAY,OAAZ,CAAJ,EAhRiD,cAgRvB,CAAe,
OAAf,CAA1B,C;;QAhRhB,OAkRO,W;O;KA7RX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;Q
AUI,eAAiC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBAAc,mBAAoB,QAApB,C;QAiRL,Q;QA
AhB,iD;UAAgB,cAAhB,e;UACI,WAAY,aAlRoC,WAkRhC,CAAY,OAAZ,CAAJ,EAlRiD,cAkRvB,CAAe,OAAf,
CAA1B,C;;QAlRhB,OAoRO,W;O;KA/RX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAUI,eA
AiC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBAAc,mBAAoB,QAApB,C;QAmRL,Q;QAAhB,iD
;UAAgB,cAAhB,e;UACI,WAAY,aApRoC,WAoRhC,CAAY,OAAZ,CAAJ,EApRiD,cAoRvB,CAAe,OAAf,CAA1
B,C;;QApRhB,OAsRO,W;O;KAjSX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAUI,eAAiC,c
AAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBAAc,mBAAoB,QAApB,C;QAqRL,Q;QAAhB,iD;UAA
gB,cAAhB,e;UACI,WAAY,aAtRoC,WAsRhC,CAAY,OAAZ,CAAJ,EAtRiD,cAsRvB,CAAe,OAAf,CAA1B,C;;Q
AtRhB,OAwRO,W;O;KAnSX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAUI,eAAiC,cAAlB,
YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBAAc,mBAAoB,QAApB,C;QAuRL,Q;QAAhB,iD;UAAgB,cA
AhB,e;UACI,WAAY,aAxRoC,WAwRhC,CAAY,OAAZ,CAAJ,EAxRiD,cAwRvB,CAAe,OAAf,CAA1B,C;;QAxR
hB,OA0RO,W;O;KArSX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAUI,eAAiC,cAAlB,YAA
Y,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBAAc,mBAAoB,QAApB,C;QAyRL,Q;QAAhB,iD;UAAgB,cAAhB,
e;UACI,WAAY,aA1RoC,WA0RhC,CAAY,OAAZ,CAAJ,EA1RiD,cA0RvB,CAAe,OAAf,CAA1B,C;;QA1RhB,O
A4RO,W;O;KAvSX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yD;QAUI,eAAiC,cAAlB,YAAY,gB
AAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBAAc,mBAAoB,QAApB,C;QA2RL,Q;QAAhB,iD;UAAgB,cAAhB,e;UA
CI,WAAY,aA5RoC,WA4RhC,CAAY,OAAZ,CAAJ,EA5RiD,cA4RvB,CAAe,OAAf,CAA1B,C;;QA5RhB,OA8RO
,W;O;KAzSX,C;+FAcA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MA8RA,oC;MAAA,gC;MA9RA,yD;QAUI,eAAiC
,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,C;QAC1B,kBAAc,mBAAoB,QAApB,C;QA6RL,Q;QAAhB,iD;UA
AgB,cAAhB,0B;UACI,WAAY,aA9RoC,WA8RhC,CAAY,oBAAZ,CAAJ,EA9RiD,cA8RvB,CAAe,oBAAf,CAA1
B,C;;QA9RhB,OAgSO,W;O;KA3SX,C;gGAcA,+C;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SA
AhB,M;QACI,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,W;K;kGAGX,+C;MAUoB
,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,O
AA1B,C;;MAEhB,OAAO,W;K;kGAGX,+C;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M
;QACI,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,W;K;iGAGX,+C;MAUoB,Q;MA
AhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,
C;;MAEhB,OAAO,W;K;kGAGX,+C;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,
WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,W;K;kGAGX,+C;MAUoB,Q;MAAhB,w
BAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MA
EhB,OAAO,W;K;kGAGX,+C;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAA
Y,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,W;K;kGAGX,+C;MAUoB,Q;MAAhB,wBAAg
B,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,
OAAO,W;K;iGAGX,yB;MAAA,oC;MAAA,gC;MAAA,sD;QAUoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cA
AhB,UAAgB,SAAhB,O;UACI,WAAY,aAAI,YAAY,oBAAZ,CAAJ,EAA0B,oBAA1B,C;;QAEhB,OAAO,W;O;K
AbX,C;kGAgBA,+D;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,Y
AAY,OAAZ,CAAJ,EAA0B,eAAe,OAAf,CAA1B,C;;MAEhB,OAAO,W;K;kGAGX,+D;MAUoB,Q;MAAhB,wBA
AgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,eAAe,OAAf,CAA

1B,C;;MAEhB,OAAO,W;K;mGAGX,+D;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;Q
ACI,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,eAAe,OAAf,CAA1B,C;;MAEhB,OAAO,W;K;mGAGX,+D;MA
UoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA
0B,eAAe,OAAf,CAA1B,C;;MAEhB,OAAO,W;K;mGAGX,+D;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAg
B,cAAA,SAAhB,M;QACI,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,eAAe,OAAf,CAA1B,C;;MAEhB,OAAO,
W;K;mGAGX,+D;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,YA
AY,OAAZ,CAAJ,EAA0B,eAAe,OAAf,CAA1B,C;;MAEhB,OAAO,W;K;mGAGX,+D;MAUoB,Q;MAAhB,wBA
AgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,eAAe,OAAf,CAA
1B,C;;MAEhB,OAAO,W;K;mGAGX,+D;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;Q
ACI,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,eAAe,OAAf,CAA1B,C;;MAEhB,OAAO,W;K;mGAGX,yB;MA
AA,oC;MAAA,gC;MAAA,sE;QAUoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UAC
I,WAAY,aAAI,YAAY,oBAAZ,CAAJ,EAA0B,eAAe,oBAAf,CAA1B,C;;QAEhB,OAAO,W;O;KAbX,C;2FAgBA,
6C;MASoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAe,UAAU,OAAV,C;QM3+Q
nB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;MN6+QA,OAAO,W;K;8FAGX,6C;MASoB,Q;MAAhB,wBA
AgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAe,UAAU,OAAV,C;QM1/QnB,wBAAI,IAAK,MAAT,EA
AgB,IAAK,OAArB,C;;MN4/QA,OAAO,W;K;8FAGX,6C;MASoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cA
AA,SAAhB,M;QACI,WAAe,UAAU,OAAV,C;QMzgRnB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;MN2g
RA,OAAO,W;K;8FAGX,6C;MASoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAe,
UAAU,OAAV,C;QMxhRnB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;MN0hRA,OAAO,W;K;8FAGX,6C;
MASoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAe,UAAU,OAAV,C;QMviRnB,w
BAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;MNyiRA,OAAO,W;K;8FAGX,6C;MASoB,Q;MAAhB,wBAAgB,
SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAe,UAAU,OAAV,C;QMtjRnB,wBAAI,IAAK,MAAT,EAAgB,I
AAK,OAArB,C;;MNwjRA,OAAO,W;K;8FAGX,6C;MASoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SA
AhB,M;QACI,WAAe,UAAU,OAAV,C;QMrkRnB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;MNukRA,OA
AO,W;K;8FAGX,6C;MASoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAe,UAAU,
OAAV,C;QMplRnB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;MNslRA,OAAO,W;K;8FAGX,yB;MAAA,o
C;MAAA,gC;MAAA,oD;QASoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,W
AAe,UAAU,oBAAV,C;UMnmRnB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;QNqmRA,OAAO,W;O;KAZ
X,C;gGAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAYI,aAAa,mBAAsC,cAAlB,YAAY,gBAAZ,CA
AkB,EAAc,EAAd,CAAtC,C;QAsJG,Q;QAAhB,iD;UAAgB,cAAhB,e;UArJuB,MAsJP,aAAI,OAAJ,EAtJe,aAsJF,C
AAc,OAAd,CAAb,C;;QAtJhB,OAAuB,M;O;KAb3B,C;kGAgBA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2
C;QAaI,aAAa,mBAAyC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,CAAzC,C;QAsJG,Q;QAAhB,iD;UAAgB,c
AAhB,e;UArJuB,MAsJP,aAAI,OAAJ,EAtJe,aAsJF,CAAc,OAAd,CAAb,C;;QAtJhB,OAAuB,M;O;KAd3B,C;kGAi
BA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAaI,aAAa,mBAA0C,cAAlB,YAAY,gBAAZ,CAAkB,EAA
c,EAAd,CAA1C,C;QAsJG,Q;QAAhB,iD;UAAgB,cAAhB,e;UArJuB,MAsJP,aAAI,OAAJ,EAtJe,aAsJF,CAAc,OA
Ad,CAAb,C;;QAtJhB,OAAuB,M;O;KAd3B,C;kGAiBA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAaI,a
AAa,mBAAwC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,CAAxC,C;QAsJG,Q;QAAhB,iD;UAAgB,cAAhB,e;
UArJuB,MAsJP,aAAI,OAAJ,EAtJe,aAsJF,CAAc,OAAd,CAAb,C;;QAtJhB,OAAuB,M;O;KAd3B,C;kGAiBA,yB;
MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAaI,aAAa,mBAAyC,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAA
d,CAAzC,C;QAsJG,Q;QAAhB,iD;UAAgB,cAAhB,e;UArJuB,MAsJP,aAAI,OAAJ,EAtJe,aAsJF,CAAc,OAAd,CA
Ab,C;;QAtJhB,OAAuB,M;O;KAd3B,C;kGAiBA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAaI,aAAa,m
BAA0C,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,CAA1C,C;QAsJG,Q;QAAhB,iD;UAAgB,cAAhB,e;UArJuB
,MAsJP,aAAI,OAAJ,EAtJe,aAsJF,CAAc,OAAd,CAAb,C;;QAtJhB,OAAuB,M;O;KAd3B,C;kGAiBA,yB;MAAA,0
D;MAAA,yD;MAAA,uE;MAAA,2C;QAaI,aAAa,mBAA2C,cAAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,CAA3C
,C;QAsJG,Q;QAAhB,iD;UAAgB,cAAhB,e;UArJuB,MAsJP,aAAI,OAAJ,EAtJe,aAsJF,CAAc,OAAd,CAAb,C;;QA
tJhB,OAAuB,M;O;KAd3B,C;kGAiBA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAaI,aAAa,mBAA4C,c
AAlB,YAAY,gBAAZ,CAAkB,EAAc,EAAd,CAA5C,C;QAsJG,Q;QAAhB,iD;UAAgB,cAAhB,e;UArJuB,MAsJP,a
AAI,OAAJ,EAtJe,aAsJF,CAAc,OAAd,CAAb,C;;QAtJhB,OAAuB,M;O;KAd3B,C;kGAiBA,yB;MAAA,uD;MAA

A,0D;MAAA,yD;MAAA,uE;MAwJA,oC;MAAA,gC;MAxJA,2C;QAaI,aAAa,mBAA2D,cAApC,YAAiB,aAAL,gB
AAK,EAAa,GAAb,CAAjB,CAAoC,EAAc,EAAd,CAA3D,C;QAsJG,Q;QAAhB,iD;UAAgB,cAAhB,0B;UArJuB,
MAsJP,aAAI,oBAAJ,EAtJe,aAsJF,CAAc,oBAAd,CAAb,C;;QAtJhB,OAAuB,M;O;KAd3B,C;oGAiBA,iD;MAUo
B,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CA
Ab,C;;MAEhB,OAAO,W;K;sGAGX,iD;MAWoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;Q
ACI,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;sGAGX,iD;MAWoB,Q;MAAhB,w
BAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,
OAAO,W;K;sGAGX,iD;MAWoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aA
AI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;sGAGX,iD;MAWoB,Q;MAAhB,wBAAgB,SAAhB,
gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;sG
AGX,iD;MAWoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,OAAJ,EAAa,
cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;sGAGX,iD;MAWoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAA
A,SAAhB,M;QACI,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;sGAGX,iD;MAWoB
,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CA
Ab,C;;MAEhB,OAAO,W;K;sGAGX,yB;MAAA,oC;MAAA,gC;MAAA,wD;QAWoB,Q;QAAhB,wBAAgB,SAAh
B,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,WAAY,aAAI,oBAAJ,EAAa,cAAc,oBAAd,CAAb,C;;QAEhB,OA
AO,W;O;KAdX,C;IAiBA,8C;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,I
AAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAA
Y,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;Q
ACI,WAAY,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,
SAAb,M;QACI,WAAY,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QA
Aa,WAAA,SAAb,M;QACI,WAAY,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,wBAAa,SA
Ab,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;MAAb,
wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB
,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,
gD;MAIiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAb,UAAa,SAAb,O;QACI,WAAY,WAAI,iBAAJ,C;;MAEhB,O
AAO,W;K;IAGX,8B;MAII,OAAO,wBAAa,eAAW,YAAY,gBAAZ,CAAX,CAAb,C;K;IAGX,gC;MAII,OAAO,0B
AAa,eAAc,YAAY,gBAAZ,CAAd,CAAb,C;K;IAGX,gC;MAII,OAAO,0BAAa,eAAe,YAAY,gBAAZ,CAAf,CAAb
,C;K;IAGX,gC;MAII,OAAO,0BAAa,eAAa,YAAY,gBAAZ,CAAb,CAAb,C;K;IAGX,gC;MAII,OAAO,0BAAa,eA
Ac,YAAY,gBAAZ,CAAd,CAAb,C;K;IAGX,gC;MAII,OAAO,0BAAa,eAAe,YAAY,gBAAZ,CAAf,CAAb,C;K;IA
GX,gC;MAII,OAAO,0BAAa,eAAgB,YAAY,gBAAZ,CAAhB,CAAb,C;K;IAGX,gC;MAII,OAAO,0BAAa,eAAiB,
YAAY,gBAAZ,CAAjB,CAAb,C;K;IAGX,gC;MAII,OAAO,0BAAa,eAAc,YAAiB,eAAL,gBAAK,EAAa,GAAb,C
AAjB,CAAd,CAAb,C;K;IAGX,2B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aAC
A,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,qBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;
MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL,C
AAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBA
AN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,SA
AK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;a
ACA,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6
B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL
,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gB
AAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,S
AAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K
;aACA,C;UAAK,cAAO,UAAK,CAAL,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,
6B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,UAAK,CAA
L,CAAP,C;UAAL,K;;UACa,uBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,6B;MAIiB,IAAN,I;MAAA,QAAM,g
BAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,sBAAK,CAAL,EAAP,C;UAAL,K;;UACa,uBAAL
,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,kC;MAII,OAAO,iBAAe,aAAL,SAAK,CAAf,C;K;IAGX,oC;MAKiB,Q;M

ADb,WAAW,iBAAgB,gBAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAmB,IAAK,WAAI,IA
AJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBAAiB,gBAAjB,C;MACX,wBAAa,SAAb,gB;Q
AAa,WAAA,SAAb,M;QAAmB,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,i
BAAe,gBAAf,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAmB,IAAK,WAAI,IAAJ,C;;MACxB,OA
AO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBAAgB,gBAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAA
b,M;QAAmB,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBAAiB,gBAAjB,
C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAmB,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,
oC;MAKiB,Q;MADb,WAAW,iBAAkB,gBAAlB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAmB,I
AAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q;MADb,WAAW,iBAAmB,gBAAnB,C;MACX,wB
AAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAmB,IAAK,WAAI,IAAJ,C;;MACxB,OAAO,I;K;IAGX,oC;MAKiB,Q
;MADb,WAAW,iBAAgB,gBAAhB,C;MACX,wBAAa,SAAb,gB;QAAa,WAAb,UAAa,SAAb,O;QAAmB,IAAK,
WAAI,iBAAJ,C;;MACxB,OAAO,I;K;IAGX,0B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;
UAAL,K;aACA,C;UAAK,aAAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,+BAAa,qBAAiB,YAAY,gBAAZ,CA
AjB,CAAb,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;U
AAL,K;aACA,C;UAAK,aAAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAoB,YAAY,gBAAZ,CAA
pB,CAAb,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UA
AL,K;aACA,C;UAAK,aAAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAqB,YAAY,gBAAZ,CAAr
B,CAAb,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAA
L,K;aACA,C;UAAK,aAAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAmB,YAAY,gBAAZ,CAAnB
,CAAb,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAAL
,K;aACA,C;UAAK,aAAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAoB,YAAY,gBAAZ,CAApB,C
AAb,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAAL,K
;aACA,C;UAAK,aAAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAqB,YAAY,gBAAZ,CAArB,CA
Ab,C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAAL,K;a
ACA,C;UAAK,aAAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAsB,YAAY,gBAAZ,CAAtB,CAAb,
C;UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAAL,K;aAC
A,C;UAAK,aAAM,UAAK,CAAL,CAAN,C;UAAL,K;;UACQ,iCAAa,qBAAuB,YAAY,gBAAZ,CAAvB,CAAb,C;
UAHL,K;;MAAP,W;K;IAOJ,4B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAAL,K;aACA,
C;UAAK,aAAM,sBAAK,CAAL,EAAN,C;UAAL,K;;UACQ,iCAAa,qBAAoB,YAAiB,eAAL,gBAAK,EAAa,GAA
b,CAAjB,CAApB,CAAb,C;UAHL,K;;MAAP,W;K;oFAOJ,yB;MAAA,+D;MAwaA,gD;MAxaA,uC;QAMW,kBA
AU,gB;QAsaD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAva6B,SAualB,CAAU,OAAV,C;UACC,OAAZ,WAAY,
EAAO,IAAP,C;;QAxahB,OA0aO,W;O;KAhbX,C;sFASA,yB;MAAA,+D;MA0aA,gD;MA1aA,uC;QAMW,kBAA
U,gB;QAwaD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAza6B,SAyalB,CAAU,OAAV,C;UACC,OAAZ,WAAY,E
AAO,IAAP,C;;QA1ahB,OA4aO,W;O;KAlbX,C;sFASA,yB;MAAA,+D;MA4aA,gD;MA5aA,uC;QAMW,kBAAU,
gB;QA0aD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA3a6B,SA2alB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EA
AO,IAAP,C;;QA5ahB,OA8aO,W;O;KApbX,C;sFASA,yB;MAAA,+D;MA8aA,gD;MA9aA,uC;QAMW,kBAAU,g
B;QA4aD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA7a6B,SA6alB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAA
O,IAAP,C;;QA9ahB,OAgbO,W;O;KAtbX,C;sFASA,yB;MAAA,+D;MAgbA,gD;MAhbA,uC;QAMW,kBAAU,gB;
QA8aD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WA/a6B,SA+alB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,I
AAP,C;;QAhbhB,OAkbO,W;O;KAxbX,C;sFASA,yB;MAAA,+D;MAkbA,gD;MAlbA,uC;QAMW,kBAAU,gB;QA
gbD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAjb6B,SAiblB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAA
P,C;;QAlbhB,OAobO,W;O;KA1bX,C;sFASA,yB;MAAA,+D;MAobA,gD;MApbA,uC;QAMW,kBAAU,gB;QAkb
D,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAnb6B,SAmblB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAA
P,C;;QApbhB,OAsbO,W;O;KA5bX,C;sFASA,yB;MAAA,+D;MAsbA,gD;MAtbA,uC;QAMW,kBAAU,gB;QAobD
,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WArb6B,SAqblB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C
;;QAtbhB,OAwbO,W;O;KA9bX,C;sFASA,yB;MAAA,+D;MAwbA,oC;MAAA,gD;MAAA,gC;MAxbA,uC;QAM
W,kBAAU,gB;QAsbD,Q;QAAhB,iD;UAAgB,cAAhB,0B;UACI,WAvb6B,SAublB,CAAU,oBAAV,C;UACC,OAA
Z,WAAY,EAAO,IAAP,C;;QAxbhB,OA0bO,W;O;KAhcX,C;sFASA,yB;MAAA,+D;MA0bA,gD;MA1bA,uC;QAU

W,kBAAU,gB;QAwbD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,WAzb6B,SAyblB,CAAU,OAAV,C;UACC,OAAZ,
WAAY,EAAO,IAAP,C;;QA1bhB,OA4bO,W;O;KAtcX,C;kGAaA,yB;MAAA,+D;MAsJA,gD;MAtJA,uC;QAYW,
kBAAiB,gB;QAqJR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WAtJoC,SAsJzB,EAAU,cAAV,EA
AU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAvJhB,OAyJO,W;O;KArKX,C;oGAe
A,yB;MAAA,+D;MAyJA,gD;MAzJA,uC;QAYW,kBAAiB,gB;QAwJR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,c
AAhB,e;UACI,WAzJoC,SAyJzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAA
O,IAAP,C;;QA1JhB,OA4JO,W;O;KAxKX,C;oGAeA,yB;MAAA,+D;MA4JA,gD;MA5JA,uC;QAYW,kBAAiB,gB
;QA2JR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WA5JoC,SA4JzB,EAAU,cAAV,EAAU,sBAA
V,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA7JhB,OA+JO,W;O;KA3KX,C;oGAeA,yB;MA
AA,+D;MA+JA,gD;MA/JA,uC;QAYW,kBAAiB,gB;QA8JR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;U
ACI,WA/JoC,SA+JzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;
QAhKhB,OAkKO,W;O;KA9KX,C;oGAeA,yB;MAAA,+D;MAkKA,gD;MAlKA,uC;QAYW,kBAAiB,gB;QAiKR,
gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WAlKoC,SAkKzB,EAAU,cAAV,EAAU,sBAAV,WAA
mB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAnKhB,OAqKO,W;O;KAjLX,C;oGAeA,yB;MAAA,+D;
MAqKA,gD;MArKA,uC;QAYW,kBAAiB,gB;QAoKR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,
WArKoC,SAqKzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA
tKhB,OAwKO,W;O;KApLX,C;oGAeA,yB;MAAA,+D;MAwKA,gD;MAxKA,uC;QAYW,kBAAiB,gB;QAuKR,gB
;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,WAxKoC,SAwKzB,EAAU,cAAV,EAAU,sBAAV,WAAm
B,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAzKhB,OA2KO,W;O;KAvLX,C;oGAeA,yB;MAAA,+D;M
A2KA,gD;MA3KA,uC;QAYW,kBAAiB,gB;QA0KR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,e;UACI,W
A3KoC,SA2KzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA5
KhB,OA8KO,W;O;KA1LX,C;oGAeA,yB;MAAA,+D;MA8KA,oC;MAAA,gD;MAAA,gC;MA9KA,uC;QAYW,kB
AAiB,gB;QA6KR,gB;QADhB,YAAY,C;QACZ,iD;UAAgB,cAAhB,0B;UACI,WA9KoC,SA8KzB,EAAU,cAAV,E
AAU,sBAAV,WAAmB,oBAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA/KhB,OAiLO,W;O;KA7LX,C;oG
AeA,yB;MAAA,+D;MAiLA,gD;MAjLA,uC;QAYW,kBAAiB,gB;QAgLR,gB;QADhB,YAAY,C;QACZ,iD;UAAg
B,cAAhB,e;UACI,WAjLoC,SAiLzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,E
AAO,IAAP,C;;QAlLhB,OAoLO,W;O;KAhMX,C;sGAeA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,Y
AAY,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,W
AAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAfX,C;uGAkBA,yB;MAAA,gD;
MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,
WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAA
O,W;O;KAfX,C;wGAkBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACZ,wBAAgB,SA
AhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OA
AZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAfX,C;wGAkBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,
M;QAFzB,YAAY,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAA
U,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAfX,C;wGAkBA,y
B;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAA
hB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;
QAEhB,OAAO,W;O;KAfX,C;wGAkBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACZ,
wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB
,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAfX,C;wGAkBA,yB;MAAA,gD;MAAA,oD;QA
WoB,UACS,M;QAFzB,YAAY,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,
cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAfX,C;
wGAkBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,
cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAA
O,IAAP,C;;QAEhB,OAAO,W;O;KAfX,C;wGAkBA,yB;MAAA,oC;MAAA,gD;MAAA,gC;MAAA,oD;QAWoB,U
ACS,M;QAFzB,YAAY,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,WAAW,WAA
U,cAAV,EAAU,sBAAV,WAAmB,oBAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAf

X,C;wGAkBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACZ,wBAAgB,SAAhB,gB;UA
AgB,cAAA,SAAhB,M;UACI,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,
EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAfX,C;uFAkBA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,
SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;Q
AEhB,OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAg
B,cAAA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O
;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M
;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;0FAWA,
yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,UA
AU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MA
AA,oD;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UAC
C,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;Q
AAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,E
AAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,S
AAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA
EhB,OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,
cAAA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;K
ARX,C;0FAWA,yB;MAAA,oC;MAAA,gD;MAAA,gC;MAAA,oD;QAIoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAA
gB,cAAhB,UAAgB,SAAhB,O;UACI,WAAW,UAAU,oBAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,
OAAO,W;O;KARX,C;0FAWA,yB;MAAA,gD;MAAA,oD;QAQoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cA
AA,SAAhB,M;UACI,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KA
ZX,C;oFAeA,yB;MAAA,wE;MAiOA,+D;MAjOA,yC;QASW,kBAAU,oB;QAiOD,Q;QAAhB,iD;UAAgB,cAAhB,
e;UACI,UAlOiD,WAkOvC,CAAY,OAAZ,C;UMv5UP,U;UADP,YNy5Ue,WMz5UH,WNy5UwB,GMz5UxB,C;U
ACL,IAAI,aAAJ,C;YACH,aNu5UuC,gB;YAA5B,WMt5UX,aNs5UgC,GMt5UhC,EAAS,MAAT,C;YACA,e;;YAE
A,c;;UNm5UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QApOT,OAsOO,W;O;KA/OX,C;sFAYA,yB;MAAA,wE;MAsO
A,+D;MAtOA,yC;QASW,kBAAU,oB;QAsOD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAvOoD,WAuO1C,CAAY
,OAAZ,C;UMx6UP,U;UADP,YN06Ue,WM16UH,WN06UwB,GM16UxB,C;UACL,IAAI,aAAJ,C;YACH,aNw6U
uC,gB;YAA5B,WMv6UX,aNu6UgC,GMv6UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNo6UA,iB;UACA,IAAK,
WAAI,OAAJ,C;;QAzOT,OA2OO,W;O;KApPX,C;sFAYA,yB;MAAA,wE;MA2OA,+D;MA3OA,yC;QASW,kBA
AU,oB;QA2OD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA5OqD,WA4O3C,CAAY,OAAZ,C;UMz7UP,U;UADP,
YN27Ue,WM37UH,WN27UwB,GM37UxB,C;UACL,IAAI,aAAJ,C;YACH,aNy7UuC,gB;YAA5B,WMx7UX,aNw
7UgC,GMx7UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNq7UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QA9OT,OAgP
O,W;O;KAzPX,C;sFAYA,yB;MAAA,wE;MAgPA,+D;MAhPA,yC;QASW,kBAAU,oB;QAgPD,Q;QAAhB,iD;UA
AgB,cAAhB,e;UACI,UAjPmD,WAiPzC,CAAY,OAAZ,C;UM18UP,U;UADP,YN48Ue,WM58UH,WN48UwB,G
M58UxB,C;UACL,IAAI,aAAJ,C;YACH,aN08UuC,gB;YAA5B,WMz8UX,aNy8UgC,GMz8UhC,EAAS,MAAT,C;
YACA,e;;YAEA,c;;UNs8UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAnPT,OAqPO,W;O;KA9PX,C;sFAYA,yB;MA
AA,wE;MAqPA,+D;MArPA,yC;QASW,kBAAU,oB;QAqPD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAtPoD,WA
sP1C,CAAY,OAAZ,C;UM39UP,U;UADP,YN69Ue,WM79UH,WN69UwB,GM79UxB,C;UACL,IAAI,aAAJ,C;Y
ACH,aN29UuC,gB;YAA5B,WM19UX,aN09UgC,GM19UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNu9UA,iB;U
ACA,IAAK,WAAI,OAAJ,C;;QAxPT,OA0PO,W;O;KAnQX,C;sFAYA,yB;MAAA,wE;MA0PA,+D;MA1PA,yC;Q
ASW,kBAAU,oB;QA0PD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA3PqD,WA2P3C,CAAY,OAAZ,C;UM5+UP,
U;UADP,YN8+Ue,WM9+UH,WN8+UwB,GM9+UxB,C;UACL,IAAI,aAAJ,C;YACH,aN4+UuC,gB;YAA5B,WM
3+UX,aN2+UgC,GM3+UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNw+UA,iB;UACA,IAAK,WAAI,OAAJ,C;;Q
A7PT,OA+PO,W;O;KAxQX,C;sFAYA,yB;MAAA,wE;MA+PA,+D;MA/PA,yC;QASW,kBAAU,oB;QA+PD,Q;Q
AAhB,iD;UAAgB,cAAhB,e;UACI,UAhQsD,WAgQ5C,CAAY,OAAZ,C;UM7/UP,U;UADP,YN+/Ue,WM//UH,W
N+/UwB,GM//UxB,C;UACL,IAAI,aAAJ,C;YACH,aN6/UuC,gB;YAA5B,WM5/UX,aN4/UgC,GM5/UhC,EAAS,
MAAT,C;YACA,e;;YAEA,c;;UNy/UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAlQT,OAoQO,W;O;KA7QX,C;sFAY
A,yB;MAAA,wE;MAoQA,+D;MApQA,yC;QASW,kBAAU,oB;QAoQD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,U

ArQuD,WAqQ7C,CAAY,OAAZ,C;UM9gVP,U;UADP,YNghVe,WMhhVH,WNghVwB,GMhhVxB,C;UACL,IAA
I,aAAJ,C;YACH,aN8gVuC,gB;YAA5B,WM7gVX,aN6gVgC,GM7gVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN
0gVA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAvQT,OAyQO,W;O;KAlRX,C;sFAYA,yB;MAAA,wE;MAyQA,oC;M
AAA,+D;MAAA,gC;MAzQA,yC;QASW,kBAAU,oB;QAyQD,Q;QAAhB,iD;UAAgB,cAAhB,0B;UACI,UA1QoD
,WA0Q1C,CAAY,oBAAZ,C;UM/hVP,U;UADP,YNiiVe,WMjiVH,WNiiVwB,GMjiVxB,C;UACL,IAAI,aAAJ,C;
YACH,aN+hVuC,gB;YAA5B,WM9hVX,aN8hVgC,GM9hVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN2hVA,iB;
UACA,IAAK,WAAI,oBAAJ,C;;QA5QT,OA8QO,W;O;KAvRX,C;sFAYA,yB;MAAA,wE;MA8QA,+D;MA9QA,y
D;QAUW,kBAAU,oB;QA8QD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA/QiD,WA+QvC,CAAY,OAAZ,C;UMjj
VP,U;UADP,YNmjVe,WMnjVH,WNmjVwB,GMnjVxB,C;UACL,IAAI,aAAJ,C;YACH,aNijVuC,gB;YAA5B,W
MhjVX,aNgjVgC,GMhjVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN6iVA,iB;UACA,IAAK,WAjRyD,cAiRrD,C
AAe,OAAf,CAAJ,C;;QAjRT,OAmRO,W;O;KA7RX,C;sFAaA,yB;MAAA,wE;MAmRA,+D;MAnRA,yD;QAUW,
kBAAU,oB;QAmRD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UApRiD,WAoRvC,CAAY,OAAZ,C;UMnkVP,U;U
ADP,YNqkVe,WMrkVH,WNqkVwB,GMrkVxB,C;UACL,IAAI,aAAJ,C;YACH,aNmkVuC,gB;YAA5B,WMlkVX
,aNkkVgC,GMlkVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN+jVA,iB;UACA,IAAK,WAtRyD,cAsRrD,CAAe,O
AAf,CAAJ,C;;QAtRT,OAwRO,W;O;KAlSX,C;uFAaA,yB;MAAA,wE;MAwRA,+D;MAxRA,yD;QAUW,kBAAU,
oB;QAwRD,Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UAzRiD,WAyRvC,CAAY,OAAZ,C;UMrlVP,U;UADP,YNul
Ve,WMvlVH,WNulVwB,GMvlVxB,C;UACL,IAAI,aAAJ,C;YACH,aNqlVuC,gB;YAA5B,WMplVX,aNolVgC,G
MplVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNilVA,iB;UACA,IAAK,WA3RyD,cA2RrD,CAAe,OAAf,CAAJ,C
;;QA3RT,OA6RO,W;O;KAvSX,C;uFAaA,yB;MAAA,wE;MA6RA,+D;MA7RA,yD;QAUW,kBAAU,oB;QA6RD,
Q;QAAhB,iD;UAAgB,cAAhB,e;UACI,UA9RiD,WA8RvC,CAAY,OAAZ,C;UMvmVP,U;UADP,YNymVe,WMz
mVH,WNymVwB,GMzmVxB,C;UACL,IAAI,aAAJ,C;YACH,aNumVuC,gB;YAA5B,WMtmVX,aNsmVgC,GMt
mVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNmmVA,iB;UACA,IAAK,WAhSyD,cAgSrD,CAAe,OAAf,CAAJ,C;
;QAhST,OAkSO,W;O;KA5SX,C;uFAaA,yB;MAAA,wE;MAkSA,+D;MAlSA,yD;QAUW,kBAAU,oB;QAkSD,Q;
QAAhB,iD;UAAgB,cAAhB,e;UACI,UAnSiD,WAmSvC,CAAY,OAAZ,C;UMznVP,U;UADP,YN2nVe,WM3nVH
,WN2nVwB,GM3nVxB,C;UACL,IAAI,aAAJ,C;YACH,aNynVuC,gB;YAA5B,WMxnVX,aNwnVgC,GMxnVhC,E
AAS,MAAT,C;YACA,e;;YAEA,c;;UNqnVA,iB;UACA,IAAK,WArSyD,cAqSrD,CAAe,OAAf,CAAJ,C;;QArST,O
AuSO,W;O;KAjTX,C;uFAaA,yB;MAAA,wE;MAuSA,+D;MAvSA,yD;QAUW,kBAAU,oB;QAuSD,Q;QAAhB,iD
;UAAgB,cAAhB,e;UACI,UAxSiD,WAwSvC,CAAY,OAAZ,C;UM3oVP,U;UADP,YN6oVe,WM7oVH,WN6oVw
B,GM7oVxB,C;UACL,IAAI,aAAJ,C;YACH,aN2oVuC,gB;YAA5B,WM1oVX,aN0oVgC,GM1oVhC,EAAS,MAA
T,C;YACA,e;;YAEA,c;;UNuoVA,iB;UACA,IAAK,WA1SyD,cA0SrD,CAAe,OAAf,CAAJ,C;;QA1ST,OA4SO,W;
O;KAtTX,C;uFAaA,yB;MAAA,wE;MA4SA,+D;MA5SA,yD;QAUW,kBAAU,oB;QA4SD,Q;QAAhB,iD;UAAgB,
cAAhB,e;UACI,UA7SiD,WA6SvC,CAAY,OAAZ,C;UM7pVP,U;UADP,YN+pVe,WM/pVH,WN+pVwB,GM/pVx
B,C;UACL,IAAI,aAAJ,C;YACH,aN6pVuC,gB;YAA5B,WM5pVX,aN4pVgC,GM5pVhC,EAAS,MAAT,C;YACA,
e;;YAEA,c;;UNypVA,iB;UACA,IAAK,WA/SyD,cA+SrD,CAAe,OAAf,CAAJ,C;;QA/ST,OAiTO,W;O;KA3TX,C;
uFAaA,yB;MAAA,wE;MAiTA,+D;MAjTA,yD;QAUW,kBAAU,oB;QAiTD,Q;QAAhB,iD;UAAgB,cAAhB,e;UA
CI,UAlTiD,WAkTvC,CAAY,OAAZ,C;UM/qVP,U;UADP,YNirVe,WMjrVH,WNirVwB,GMjrVxB,C;UACL,IAAI
,aAAJ,C;YACH,aN+qVuC,gB;YAA5B,WM9qVX,aN8qVgC,GM9qVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN
2qVA,iB;UACA,IAAK,WApTyD,cAoTrD,CAAe,OAAf,CAAJ,C;;QApTT,OAsTO,W;O;KAhUX,C;uFAYA,yB;M
AAA,wE;MAsTA,oC;MAAA,+D;MAAA,gC;MAtTA,yD;QAUW,kBAAU,oB;QAsTD,Q;QAAhB,iD;UAAgB,cAA
hB,0B;UACI,UAvTiD,WAuTvC,CAAY,oBAAZ,C;UMjsVP,U;UADP,YNmsVe,WMnsVH,WNmsVwB,GMnsVx
B,C;UACL,IAAI,aAAJ,C;YACH,aNisVuC,gB;YAA5B,WMhsVX,aNgsVgC,GMhsVhC,EAAS,MAAT,C;YACA,e
;;YAEA,c;;UN6rVA,iB;UACA,IAAK,WAzTyD,cAyTrD,CAAe,oBAAf,CAAJ,C;;QAzTT,OA2TO,W;O;KArUX,C
;wFAaA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,U
AAU,YAAY,OAAZ,C;UMv5UP,U;UADP,YNy5Ue,WMz5UH,WNy5UwB,GMz5UxB,C;UACL,IAAI,aAAJ,C;Y
ACH,aNu5UuC,gB;YAA5B,WMt5UX,aNs5UgC,GMt5UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNm5UA,iB;U
ACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB
,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UMx6UP,U;UADP,YN06Ue,W
M16UH,WN06UwB,GM16UxB,C;UACL,IAAI,aAAJ,C;YACH,aNw6UuC,gB;YAA5B,WMv6UX,aNu6UgC,GMv

6UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNo6UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAd
X,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UA
CI,UAAU,YAAY,OAAZ,C;UMz7UP,U;UADP,YN27Ue,WM37UH,WN27UwB,GM37UxB,C;UACL,IAAI,aAAJ,
C;YACH,aNy7UuC,gB;YAA5B,WMx7UX,aNw7UgC,GMx7UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNq7UA,i
B;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QA
AhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UM18UP,U;UADP,YN48U
e,WM58UH,WN48UwB,GM58UxB,C;UACL,IAAI,aAAJ,C;YACH,aN08UuC,gB;YAA5B,WMz8UX,aNy8UgC,G
Mz8UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNs8UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;K
AdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;U
ACI,UAAU,YAAY,OAAZ,C;UM39UP,U;UADP,YN69Ue,WM79UH,WN69UwB,GM79UxB,C;UACL,IAAI,aAA
J,C;YACH,aN29UuC,gB;YAA5B,WM19UX,aN09UgC,GM19UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNu9UA
,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;Q
AAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UM5+UP,U;UADP,YN8+
Ue,WM9+UH,WN8+UwB,GM9+UxB,C;UACL,IAAI,aAAJ,C;YACH,aN4+UuC,gB;YAA5B,WM3+UX,aN2+Ug
C,GM3+UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNw+UA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;
O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,
M;UACI,UAAU,YAAY,OAAZ,C;UM7/UP,U;UADP,YN+/Ue,WM//UH,WN+/UwB,GM//UxB,C;UACL,IAAI,aA
AJ,C;YACH,aN6/UuC,gB;YAA5B,WM5/UX,aN4/UgC,GM5/UhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNy/UA,i
B;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,sD;QASoB,Q;QA
AhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UM9gVP,U;UADP,YNghV
e,WMhhVH,WNghVwB,GMhhVxB,C;UACL,IAAI,aAAJ,C;YACH,aN8gVuC,gB;YAA5B,WM7gVX,aN6gVgC,G
M7gVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN0gVA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;K
AdX,C;0FAiBA,yB;MAAA,oC;MAAA,+D;MAAA,gC;MAAA,sD;QASoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAA
gB,cAAhB,UAAgB,SAAhB,O;UACI,UAAU,YAAY,oBAAZ,C;UM/hVP,U;UADP,YNiiVe,WMjiVH,WNiiVwB,
GMjiVxB,C;UACL,IAAI,aAAJ,C;YACH,aN+hVuC,gB;YAA5B,WM9hVX,aN8hVgC,GM9hVhC,EAAS,MAAT,
C;YACA,e;;YAEA,c;;UN2hVA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QAET,OAAO,W;O;KAdX,C;0FAiBA,yB;MA
AA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OA
AZ,C;UMjjVP,U;UADP,YNmjVe,WMnjVH,WNmjVwB,GMnjVxB,C;UACL,IAAI,aAAJ,C;YACH,aNijVuC,gB;
YAA5B,WMhjVX,aNgjVgC,GMhjVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN6iVA,iB;UACA,IAAK,WAAI,eA
Ae,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAfX,C;0FAkBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAg
B,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UMnkVP,U;UADP,YNqkVe,WMrkVH,
WNqkVwB,GMrkVxB,C;UACL,IAAI,aAAJ,C;YACH,aNmkVuC,gB;YAA5B,WMlkVX,aNkkVgC,GMlkVhC,EA
AS,MAAT,C;YACA,e;;YAEA,c;;UN+jVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;K
AfX,C;2FAkBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;
UACI,UAAU,YAAY,OAAZ,C;UMrlVP,U;UADP,YNulVe,WMvlVH,WNulVwB,GMvlVxB,C;UACL,IAAI,aAAJ,
C;YACH,aNqlVuC,gB;YAA5B,WMplVX,aNolVgC,GMplVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNilVA,iB;
UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAfX,C;2FAkBA,yB;MAAA,+D;MAAA,sE;QA
UoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UMvmVP,U;UA
DP,YNymVe,WMzmVH,WNymVwB,GMzmVxB,C;UACL,IAAI,aAAJ,C;YACH,aNumVuC,gB;YAA5B,WMtmV
X,aNsmVgC,GMtmVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNmmVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,C
AAJ,C;;QAET,OAAO,W;O;KAfX,C;2FAkBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAgB,SAAhB,g
B;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UMznVP,U;UADP,YN2nVe,WM3nVH,WN2nVwB,
GM3nVxB,C;UACL,IAAI,aAAJ,C;YACH,aNynVuC,gB;YAA5B,WMxnVX,aNwnVgC,GMxnVhC,EAAS,MAAT,
C;YACA,e;;YAEA,c;;UNqnVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAfX,C;2FAk
BA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,
YAAY,OAAZ,C;UM3oVP,U;UADP,YN6oVe,WM7oVH,WN6oVwB,GM7oVxB,C;UACL,IAAI,aAAJ,C;YACH,a
N2oVuC,gB;YAA5B,WM1oVX,aN0oVgC,GM1oVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNuoVA,iB;UACA,I
AAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAfX,C;2FAkBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;Q

AAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UM7pVP,U;UADP,YN+p
Ve,WM/pVH,WN+pVwB,GM/pVxB,C;UACL,IAAI,aAAJ,C;YACH,aN6pVuC,gB;YAA5B,WM5pVX,aN4pVgC,
GM5pVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UNypVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,
OAAO,W;O;KAfX,C;2FAkBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAA
A,SAAhB,M;UACI,UAAU,YAAY,OAAZ,C;UM/qVP,U;UADP,YNirVe,WMjrVH,WNirVwB,GMjrVxB,C;UACL
,IAAI,aAAJ,C;YACH,aN+qVuC,gB;YAA5B,WM9qVX,aN8qVgC,GM9qVhC,EAAS,MAAT,C;YACA,e;;YAEA,c
;;UN2qVA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAfX,C;2FAkBA,yB;MAAA,oC;
MAAA,+D;MAAA,gC;MAAA,sE;QAUoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;
UACI,UAAU,YAAY,oBAAZ,C;UMjsVP,U;UADP,YNmsVe,WMnsVH,WNmsVwB,GMnsVxB,C;UACL,IAAI,a
AAJ,C;YACH,aNisVuC,gB;YAA5B,WMhsVX,aNgsVgC,GMhsVhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UN6rV
A,iB;UACA,IAAK,WAAI,eAAe,oBAAf,CAAJ,C;;QAET,OAAO,W;O;KAfX,C;0FAkBA,yB;MAAA,kC;MAAA,4
C;MAAA,wE;QAQW,sC;QAAA,8C;O;MARX,oDASQ,Y;QAA6C,OAAgB,qBAAhB,oBAAgB,C;O;MATrE,iDA
UQ,mB;QAAoC,gCAAY,OAAZ,C;O;MAV5C,gF;MAAA,yC;QAQI,2D;O;KARJ,C;4EAcA,yB;MAAA,gE;MAAA
,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UACI,WAAY,WAhViB,SAgVb,CA
AU,IAAV,CAAJ,C;;QAhVhB,OAiVO,W;O;KAxVX,C;8EAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa
,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UACI,WAAY,WAhViB,SAgVb,CAAU,IAAV,CAAJ,C;;QAhVh
B,OAiVO,W;O;KAxVX,C;8EAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAA
b,iD;UAAa,WAAb,e;UACI,WAAY,WAhViB,SAgVb,CAAU,IAAV,CAAJ,C;;QAhVhB,OAiVO,W;O;KAxVX,C;8
EAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UACI,
WAAY,WAhViB,SAgVb,CAAU,IAAV,CAAJ,C;;QAhVhB,OAiVO,W;O;KAxVX,C;8EAUA,yB;MAAA,gE;MA
AA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UACI,WAAY,WAhViB,SAgVb,
CAAU,IAAV,CAAJ,C;;QAhVhB,OAiVO,W;O;KAxVX,C;8EAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,e
AAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UACI,WAAY,WAhViB,SAgVb,CAAU,IAAV,CAAJ,C;;QA
hVhB,OAiVO,W;O;KAxVX,C;8EAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;Q
AAb,iD;UAAa,WAAb,e;UACI,WAAY,WAhViB,SAgVb,CAAU,IAAV,CAAJ,C;;QAhVhB,OAiVO,W;O;KAxVX,
C;8EAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,e;UA
CI,WAAY,WAhViB,SAgVb,CAAU,IAAV,CAAJ,C;;QAhVhB,OAiVO,W;O;KAxVX,C;8EAUA,yB;MAAA,gE;M
AiVA,oC;MAAA,gC;MAjVA,uC;QAOW,kBAAM,eAAa,gBAAb,C;QA+UA,Q;QAAb,iD;UAAa,WAAb,0B;UACI
,WAAY,WAhViB,SAgVb,CAAU,iBAAV,CAAJ,C;;QAhVhB,OAiVO,W;O;KAxVX,C;0FAUA,yB;MAAA,gE;M
AAA,uC;QAOW,kBAAa,eAAa,gBAAb,C;QAgHP,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UACI,WAAY,
WAjHwB,SAiHpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QAjHhB,OAkHO,W;O;KAzHX,C;4
FAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAa,eAAa,gBAAb,C;QAmHP,gB;QADb,YAAY,C;QACZ,iD;UAAa,
WAAb,e;UACI,WAAY,WApHwB,SAoHpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QApHhB,O
AqHO,W;O;KA5HX,C;4FAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAa,eAAa,gBAAb,C;QAsHP,gB;QADb,Y
AAY,C;QACZ,iD;UAAa,WAAb,e;UACI,WAAY,WAvHwB,SAuHpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IA
AnB,CAAJ,C;;QAvHhB,OAwHO,W;O;KA/HX,C;4FAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAa,eAAa,gBA
Ab,C;QAyHP,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UACI,WAAY,WA1HwB,SA0HpB,EAAU,cAAV,E
AAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QA1HhB,OA2HO,W;O;KAlIX,C;4FAUA,yB;MAAA,gE;MAAA,uC;Q
AOW,kBAAa,eAAa,gBAAb,C;QA4HP,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UACI,WAAY,WA7HwB,
SA6HpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QA7HhB,OA8HO,W;O;KArIX,C;2FAUA,yB;
MAAA,gE;MAAA,uC;QAOW,kBAAa,eAAa,gBAAb,C;QA+HP,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;U
ACI,WAAY,WAhIwB,SAgIpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QAhIhB,OAiIO,W;O;K
AxIX,C;4FAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAa,eAAa,gBAAb,C;QAkIP,gB;QADb,YAAY,C;QACZ,i
D;UAAa,WAAb,e;UACI,WAAY,WAnIwB,SAmIpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QA
nIhB,OAoIO,W;O;KA3IX,C;4FAUA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAa,eAAa,gBAAb,C;QAqIP,gB;QAD
b,YAAY,C;QACZ,iD;UAAa,WAAb,e;UACI,WAAY,WAtIwB,SAsIpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IA
AnB,CAAJ,C;;QAtIhB,OAuIO,W;O;KA9IX,C;4FAUA,yB;MAAA,gE;MAuIA,oC;MAAA,gC;MAvIA,uC;QAOW,
kBAAa,eAAa,gBAAb,C;QAwIP,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,0B;UACI,WAAY,WAzIwB,SAyIp

B,EAAU,cAAV,EAAU,sBAAV,WAAmB,iBAAnB,CAAJ,C;;QAzIhB,OA0IO,W;O;KAjJX,C;wGAUA,yB;MAAA
,+D;MAAA,uC;QAOW,kBAAoB,gB;QAklEd,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UAxkEmC,U;UAA
A,cAVQ,SAUR,EAwkET,cAxkES,EAwkET,sBAxkES,WAwkEA,IAxkEA,W;YAA6C,6B;;;QAVhF,OAWO,W;O;
KAlBX,C;4GAUA,yB;MAAA,oD;QA+kEiB,gB;QADb,YAAY,C;QACZ,iD;UAAa,WAAb,e;UAxkEmC,U;UAAA
,yBAwkET,cAxkES,EAwkET,sBAxkES,WAwkEA,IAxkEA,W;YAA6C,6B;;;QAChF,OAAO,W;O;KARX,C;8FA
WA,6C;MAQiB,UACiB,M;MAF9B,YAAY,C;MACZ,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,
WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;MAChB,OAAO,W;K;gGAGX,6C;MAQiB,UAC
iB,M;MAF9B,YAAY,C;MACZ,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,WAAU,cAAV,
EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;MAChB,OAAO,W;K;gGAGX,6C;MAQiB,UACiB,M;MAF9B,YAAY,
C;MACZ,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,WAAU,cAAV,EAAU,sBAAV,WAA
mB,IAAnB,CAAJ,C;;MAChB,OAAO,W;K;gGAGX,6C;MAQiB,UACiB,M;MAF9B,YAAY,C;MACZ,wBAAa,SA
Ab,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;
MAChB,OAAO,W;K;gGAGX,6C;MAQiB,UACiB,M;MAF9B,YAAY,C;MACZ,wBAAa,SAAb,gB;QAAa,WAAA
,SAAb,M;QACI,WAAY,WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;MAChB,OAAO,W;K;g
GAGX,6C;MAQiB,UACiB,M;MAF9B,YAAY,C;MACZ,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAA
Y,WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;MAChB,OAAO,W;K;gGAGX,6C;MAQiB,U
ACiB,M;MAF9B,YAAY,C;MACZ,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,WAAU,cAA
V,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;MAChB,OAAO,W;K;+FAGX,6C;MAQiB,UACiB,M;MAF9B,YAA
Y,C;MACZ,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,WAAU,cAAV,EAAU,sBAAV,WA
AmB,IAAnB,CAAJ,C;;MAChB,OAAO,W;K;gGAGX,yB;MAAA,oC;MAAA,gC;MAAA,oD;QAQiB,UACiB,M;Q
AF9B,YAAY,C;QACZ,wBAAa,SAAb,gB;UAAa,WAAb,UAAa,SAAb,O;UACI,WAAY,WAAI,WAAU,cAAV,EA
AU,sBAAV,WAAmB,iBAAnB,CAAJ,C;;QAChB,OAAO,W;O;KAVX,C;0FAaA,yB;MAAA,+D;MAAA,uC;QAO
W,kBAAa,gB;QAs4DJ,Q;QAAhB,iD;UAAgB,cAAhB,e;UA93DqB,U;UAAA,cARe,SAQf,CA83DQ,OA93DR,W;
YAAsC,6B;;;QAR3D,OASO,W;O;KAhBX,C;8FAUA,yB;MAAA,oD;QAm4DoB,Q;QAAhB,iD;UAAgB,cAAhB,e
;UA93DqB,U;UAAA,wBA83DQ,OA93DR,W;YAAsC,6B;;;QAC3D,OAAO,W;O;KANX,C;gFASA,6C;MAKiB,Q
;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,UAAU,IAAV,CAAJ,C;;MAChB,OAAO
,W;K;kFAGX,6C;MAKiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,UAAU,IAA
V,CAAJ,C;;MAChB,OAAO,W;K;kFAGX,6C;MAKiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAC
I,WAAY,WAAI,UAAU,IAAV,CAAJ,C;;MAChB,OAAO,W;K;kFAGX,6C;MAKiB,Q;MAAb,wBAAa,SAAb,gB;
QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,UAAU,IAAV,CAAJ,C;;MAChB,OAAO,W;K;kFAGX,6C;MAKiB,
Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,UAAU,IAAV,CAAJ,C;;MAChB,OAA
O,W;K;kFAGX,6C;MAKiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,UAAU,IA
AV,CAAJ,C;;MAChB,OAAO,W;K;kFAGX,6C;MAKiB,Q;MAAb,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QA
CI,WAAY,WAAI,UAAU,IAAV,CAAJ,C;;MAChB,OAAO,W;K;kFAGX,6C;MAKiB,Q;MAAb,wBAAa,SAAb,gB;
QAAa,WAAA,SAAb,M;QACI,WAAY,WAAI,UAAU,IAAV,CAAJ,C;;MAChB,OAAO,W;K;kFAGX,yB;MAAA,o
C;MAAA,gC;MAAA,oD;QAKiB,Q;QAAb,wBAAa,SAAb,gB;UAAa,WAAb,UAAa,SAAb,O;UACI,WAAY,WAA
I,UAAU,iBAAV,CAAJ,C;;QAChB,OAAO,W;O;KAPX,C;IAe4B,0C;MAAA,mB;QAAE,2C;O;K;IAL9B,8B;MAK
I,OAAO,qBAAiB,2BAAjB,C;K;IAQiB,4C;MAAA,mB;QAAE,+C;O;K;IAL9B,gC;MAKI,OAAO,qBAAiB,6BAAj
B,C;K;IAQiB,4C;MAAA,mB;QAAE,gD;O;K;IAL9B,gC;MAKI,OAAO,qBAAiB,6BAAjB,C;K;IAQiB,4C;MAAA,
mB;QAAE,8C;O;K;IAL9B,gC;MAKI,OAAO,qBAAiB,6BAAjB,C;K;IAQiB,4C;MAAA,mB;QAAE,+C;O;K;IAL9
B,gC;MAKI,OAAO,qBAAiB,6BAAjB,C;K;IAQiB,4C;MAAA,mB;QAAE,gD;O;K;IAL9B,gC;MAKI,OAAO,qBA
AiB,6BAAjB,C;K;IAQiB,4C;MAAA,mB;QAAE,iD;O;K;IAL9B,gC;MAKI,OAAO,qBAAiB,6BAAjB,C;K;IAQiB,
4C;MAAA,mB;QAAE,kD;O;K;IAL9B,gC;MAKI,OAAO,qBAAiB,6BAAjB,C;K;IAQiB,4C;MAAA,mB;QAAE,+
C;O;K;IAL9B,gC;MAKI,OAAO,qBAAiB,6BAAjB,C;K;IAGX,6B;MASI,OAA2B,SAAf,aAAL,SAAK,CAAe,C;K;
IAG/B,+B;MAQI,OAA2B,SAAf,eAAL,SAAK,CAAe,C;K;IAG/B,+B;MAQI,OAA2B,SAAf,eAAL,SAAK,CAAe,C
;K;IAG/B,+B;MAQI,OAA2B,SAAf,eAAL,SAAK,CAAe,C;K;IAG/B,+B;MAQI,OAA2B,SAAf,eAAL,SAAK,CAA
e,C;K;IAG/B,+B;MAQI,OAA2B,SAAf,eAAL,SAAK,CAAe,C;K;IAG/B,+B;MAQI,OAA2B,SAAf,eAAL,SAAK,C
AAe,C;K;IAG/B,+B;MAQI,OAA2B,SAAf,eAAL,SAAK,CAAe,C;K;IAG/B,+B;MAQI,OAA2B,SAAf,eAAL,SAA

K,CAAe,C;K;0FAG/B,yB;MAAA,2D;MAAA,+D;MAAA,sC;QAYc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QAC
X,wBAAU,SAAV,gB;UAAU,QAAA,SAAV,M;UACI,UAAU,SAAS,CAAT,C;UACV,IAAI,GAAI,WAAI,GAAJ,C
AAR,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAjBX,C;4FAoBA,yB;MAAA,2D;MAAA,+D;MAAA,s
C;QAWc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QACX,wBAAU,SAAV,gB;UAAU,QAAA,SAAV,M;UACI,UAA
U,SAAS,CAAT,C;UACV,IAAI,GAAI,WAAI,GAAJ,CAAR,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;K
AhBX,C;4FAmBA,yB;MAAA,2D;MAAA,+D;MAAA,sC;QAWc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QACX,w
BAAU,SAAV,gB;UAAU,QAAA,SAAV,M;UACI,UAAU,SAAS,CAAT,C;UACV,IAAI,GAAI,WAAI,GAAJ,CAA
R,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAhBX,C;4FAmBA,yB;MAAA,2D;MAAA,+D;MAAA,sC;
QAWc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QACX,wBAAU,SAAV,gB;UAAU,QAAA,SAAV,M;UACI,UAAU,
SAAS,CAAT,C;UACV,IAAI,GAAI,WAAI,GAAJ,CAAR,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAh
BX,C;4FAmBA,yB;MAAA,2D;MAAA,+D;MAAA,sC;QAWc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QACX,wBA
AU,SAAV,gB;UAAU,QAAA,SAAV,M;UACI,UAAU,SAAS,CAAT,C;UACV,IAAI,GAAI,WAAI,GAAJ,CAAR,C
;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAhBX,C;4FAmBA,yB;MAAA,2D;MAAA,+D;MAAA,sC;QA
Wc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QACX,wBAAU,SAAV,gB;UAAU,QAAA,SAAV,M;UACI,UAAU,SA
AS,CAAT,C;UACV,IAAI,GAAI,WAAI,GAAJ,CAAR,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAhBX
,C;4FAmBA,yB;MAAA,2D;MAAA,+D;MAAA,sC;QAWc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QACX,wBAA
U,SAAV,gB;UAAU,QAAA,SAAV,M;UACI,UAAU,SAAS,CAAT,C;UACV,IAAI,GAAI,WAAI,GAAJ,CAAR,C;
YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAhBX,C;4FAmBA,yB;MAAA,2D;MAAA,+D;MAAA,sC;QA
Wc,Q;QAFV,UAAU,c;QACV,WAAW,gB;QACX,wBAAU,SAAV,gB;UAAU,QAAA,SAAV,M;UACI,UAAU,SA
AS,CAAT,C;UACV,IAAI,GAAI,WAAI,GAAJ,CAAR,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAhBX
,C;4FAmBA,yB;MAAA,2D;MAAA,+D;MAAA,oC;MAAA,gC;MAAA,sC;QAWc,Q;QAFV,UAAU,c;QACV,WA
AW,gB;QACX,wBAAU,SAAV,gB;UAAU,QAAV,UAAU,SAAV,O;UACI,UAAU,SAAS,cAAT,C;UACV,IAAI,G
AAI,WAAI,GAAJ,CAAR,C;YACI,IAAK,WAAI,cAAJ,C;;QAEb,OAAO,I;O;KAhBX,C;IAmBA,qC;MAQI,UAAe,
aAAL,SAAK,C;MACX,YAAJ,GAAI,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,uC;MAQI,UAAe,eAAL,SAAK,C;
MACX,YAAJ,GAAI,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,uC;MAQI,UAAe,eAAL,SAAK,C;MACX,YA
AJ,GAAI,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,uC;MAQI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAI,EAA
U,KAAV,C;MACJ,OAAO,G;K;IAGX,uC;MAQI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAI,EAAU,KAAV,C;M
AJ,OAAO,G;K;IAGX,uC;MAQI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAI,EAAU,KAAV,C;MAJ,OAAO,G;
K;IAGX,uC;MAQI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAI,EAAU,KAAV,C;MAJ,OAAO,G;K;IAGX,uC;M
AQI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAI,EAAU,KAAV,C;MAJ,OAAO,G;K;IAGX,uC;MAQI,UAAe,eA
AL,SAAK,C;MACX,YAAJ,GAAI,EAAU,KAAV,C;MAJ,OAAO,G;K;IAGX,oC;MAMI,UAAe,aAAL,SAAK,C;
MACX,YAAJ,GAAI,EAAU,KAAV,C;MAJ,OAAO,G;K;IAGX,sC;MAMI,UAAe,eAAL,SAAK,C;MACX,YAAJ,
GAAI,EAAU,KAAV,C;MAJ,OAAO,G;K;IAGX,sC;MAMI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAI,EAAU,
KAAV,C;MAJ,OAAO,G;K;IAGX,sC;MAMI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAI,EAAU,KAAV,C;MA
CJ,OAAO,G;K;IAGX,sC;MAMI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAI,EAAU,KAAV,C;MACJ,OAAO,G;
K;IAGX,sC;MAMI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAI,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,sC;M
AMI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAI,EAAU,KAAV,C;MAJ,OAAO,G;K;IAGX,sC;MAMI,UAAe,eA
AL,SAAK,C;MACX,YAAJ,GAAI,EAAU,KAAV,C;MAJ,OAAO,G;K;IAGX,sC;MAMI,UAAe,eAAL,SAAK,C
;MACX,YAAJ,GAAI,EAAU,KAAV,C;MAJ,OAAO,G;K;IAGX,iC;MAAI,OAAO,wBAAa,qBAAiB,YAAY,gB
AAZ,CAAjB,CAAb,C;K;IAGX,mC;MAAI,OAAO,0BAAa,qBAAoB,YAAY,gBAAZ,CAApB,CAAb,C;K;IAGX,
mC;MAAI,OAAO,0BAAa,qBAAqB,YAAY,gBAAZ,CArB,CAAb,C;K;IAGX,mC;MAAI,OAAO,0BAAa,qBAA
mB,YAAY,gBAAZ,CAAnB,CAAb,C;K;IAGX,mC;MAAI,OAAO,0BAAa,qBAAoB,YAAY,gBAAZ,CApB,CA
Ab,C;K;IAGX,mC;MAAI,OAAO,0BAAa,qBAAqB,YAAY,gBAAZ,CArB,CAAb,C;K;IAGX,mC;MAAI,OAAO
,0BAAa,qBAAsB,YAAY,gBAAZ,CAtB,CAAb,C;K;IAGX,mC;MAAI,OAAO,0BAAa,qBAAuB,YAAY,gBAAZ
,CAAvB,CAAb,C;K;IAGX,mC;MAAI,OAAO,0BAAa,qBAAoB,YAAiB,eAAL,gBAAK,EAAa,GAAb,CAAjB,CA
ApB,CAAb,C;K;IAGX,iC;MAUI,UAAe,aAAL,SAAK,C;MACX,YAAJ,OAAI,EAAU,KAAV,C;MACJ,OAAO,G;
K;IAGX,mC;MAUI,UAAe,eAAL,SAAK,C;MACX,YAAJ,OAAI,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,mC;
MAUI,UAAe,eAAL,SAAK,C;MACX,YAAJ,OAAI,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,mC;MAUI,UAAe,

eAAL,SAAK,C;MACX,OAAJ,GAAI,EAAO,KAAP,C;MACJ,OAAO,G;K;IAGX,mC;MAUI,UAAe,eAAL,SAAK,
C;MACX,OAAJ,GAAI,EAAO,KAAP,C;MACJ,OAAO,G;K;IAGX,mC;MAUI,UAAe,eAAL,SAAK,C;MACX,OA
AJ,GAAI,EAAO,KAAP,C;MACJ,OAAO,G;K;IAGX,mC;MAUI,UAAe,eAAL,SAAK,C;MACX,OAAJ,GAAI,EA
AO,KAAP,C;MACJ,OAAO,G;K;IAGX,mC;MAUI,UAAe,eAAL,SAAK,C;MACX,OAAJ,GAAI,EAAO,KAAP,C;
MACJ,OAAO,G;K;IAGX,mC;MAUI,UAAe,eAAL,SAAK,C;MACX,OAAJ,GAAI,EAAO,KAAP,C;MACJ,OAAO,
G;K;4EAGX,gC;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAA
U,OAAV,CAAL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;8EAGX,gC;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB
;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;
8EAGX,gC;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,O
AAV,CAAL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;8EAGX,gC;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QA
AgB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;8EA
GX,gC;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV
,CAAL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;8EAGX,gC;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,
cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;8EAGX,
gC;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CA
AL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;8EAGX,gC;MAUoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cA
AA,SAAhB,M;QAAsB,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;8EAGX,yB;
MAAA,oC;MAAA,gC;MAAA,uC;QAUoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;
UAAsB,IAAI,CAAC,UAAU,oBAAV,CAAL,C;YAAyB,OAAO,K;;QACtD,OAAO,I;O;KAXX,C;IAcA,wB;MAMI
,OAAO,EAhpJA,qBAAQ,CAgpJR,C;K;IAGX,0B;MAMI,OAAO,EAjpJA,qBAAQ,CAipJR,C;K;IAGX,0B;MAMI,
OAAO,EAlpJA,qBAAQ,CAkpJR,C;K;IAGX,0B;MAMI,OAAO,EAnpJA,qBAAQ,CAmpJR,C;K;IAGX,0B;MAMI,
OAAO,EAppJA,qBAAQ,CAopJR,C;K;IAGX,0B;MAMI,OAAO,EArpJA,qBAAQ,CAqpJR,C;K;IAGX,0B;MAMI,
OAAO,EAtpJA,qBAAQ,CAspJR,C;K;IAGX,0B;MAMI,OAAO,EAvpJA,qBAAQ,CAupJR,C;K;IAGX,0B;MAMI,
OAAO,EAxpJA,qBAAQ,CAwpJR,C;K;8EAGX,gC;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SA
AhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;8EAGX,gC;MAMoB,Q;M
AAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;
MACrD,OAAO,K;K;+EAGX,gC;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,I
AAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;+EAGX,gC;MAMoB,Q;MAAhB,wBAAgB,
SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,
K;K;+EAGX,gC;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OA
AV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;+EAGX,gC;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QA
AgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;+EAGX,gC
;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UA
AwB,OAAO,I;;MACrD,OAAO,K;K;+EAGX,gC;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAA
hB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;+EAGX,yB;MAAA,oC;MA
AA,gC;MAAA,uC;QAMoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UAAsB,IAAI,U
AAU,oBAAV,CAAJ,C;YAAwB,OAAO,I;;QACrD,OAAO,K;O;KAPX,C;gFAUA,qB;MAKI,OAAO,gB;K;kFAGX
,qB;MAKI,OAAO,gB;K;kFAGX,qB;MAKI,OAAO,gB;K;kFAGX,qB;MAKI,OAAO,gB;K;kFAGX,qB;MAKI,OA
AO,gB;K;kFAGX,qB;MAKI,OAAO,gB;K;kFAGX,qB;MAKI,OAAO,gB;K;kFAGX,qB;MAKI,OAAO,gB;K;kFA
GX,qB;MAKI,OAAO,gB;K;kFAGX,gC;MAKoB,Q;MADhB,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAgB,cA
AA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;kFAGX,gC;MAKoB,Q;
MADhB,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C
;UAAwB,qB;;MAC9C,OAAO,K;K;mFAGX,gC;MAKoB,Q;MADhB,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QA
AgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;mFAGX,gC;MA
KoB,Q;MADhB,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,
CAAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;mFAGX,gC;MAKoB,Q;MADhB,YAAY,C;MACZ,wBAAgB,SAAhB,
gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;mFAGX,
gC;MAKoB,Q;MADhB,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,

OAAV,CAAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;mFAGX,gC;MAKoB,Q;MADhB,YAAY,C;MACZ,wBAAgB,
SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;
mFAGX,gC;MAKoB,Q;MADhB,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI
,UAAU,OAAV,CAAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;mFAGX,yB;MAAA,oC;MAAA,gC;MAAA,uC;QAKo
B,Q;QADhB,YAAY,C;QACZ,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UAAsB,IAAI,UAAU,oB
AAV,CAAJ,C;YAAwB,qB;;QAC9C,OAAO,K;O;KANX,C;8EASA,yC;MAUoB,Q;MADhB,kBAAkB,O;MAClB,
wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OA
AO,W;K;gFAGX,yC;MAUoB,Q;MADhB,kBAAkB,O;MAClB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;Q
AAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;gFAGX,yC;MAUoB,Q;MADhB,kBAAkB,
O;MAClB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;M
ACpC,OAAO,W;K;gFAGX,yC;MAUoB,Q;MADhB,kBAAkB,O;MAClB,wBAAgB,SAAhB,gB;QAAgB,cAAA,S
AAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;gFAGX,yC;MAUoB,Q;MADh
B,kBAAkB,O;MAClB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,O
AAvB,C;;MACpC,OAAO,W;K;gFAGX,yC;MAUoB,Q;MADhB,kBAAkB,O;MAClB,wBAAgB,SAAhB,gB;QAA
gB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;gFAGX,yC;MAUo
B,Q;MADhB,kBAAkB,O;MAClB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV
,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;gFAGX,yC;MAUoB,Q;MADhB,kBAAkB,O;MAClB,wBAAgB,SAAh
B,gB;QAAgB,cAAA,SAAhB,M;QAAsB,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;gFAGX,
yB;MAAA,oC;MAAA,gC;MAAA,gD;QAUoB,Q;QADhB,kBAAkB,O;QAClB,wBAAgB,SAAhB,gB;UAAgB,cA
AhB,UAAgB,SAAhB,O;UAAsB,cAAc,UAAU,WAAV,EAAuB,oBAAvB,C;;QACpC,OAAO,W;O;KAXX,C;4FAc
A,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MAClB,wBAAgB,SAAhB,gB;QAAgB,cAAA,S
AAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAnB,EAAgC,OAAhC,C;;MACpC,OAAO,W
;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MAClB,wBAAgB,SAAhB,gB;QAAgB,
cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAnB,EAAgC,OAAhC,C;;MACpC,
OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MAClB,wBAAgB,SAAhB,g
B;QAAgB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAnB,EAAgC,OAAhC,C;
;MACpC,OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MAClB,wBAAgB,
SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAnB,EAAgC,O
AAhC,C;;MACpC,OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MAClB,w
BAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAnB,E
AAgC,OAAhC,C;;MACpC,OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;
MAClB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,
WAAnB,EAAgC,OAAhC,C;;MACpC,OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kB
AAkB,O;MAClB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,
WAAmB,WAAnB,EAAgC,OAAhC,C;;MACpC,OAAO,W;K;8FAGX,yC;MAYoB,UAA8B,M;MAF9C,YAAY,C;
MACZ,kBAAkB,O;MAClB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,cAAc,WAAU,cAAV,EAA
U,sBAAV,WAAmB,WAAnB,EAAgC,OAAhC,C;;MACpC,OAAO,W;K;8FAGX,yB;MAAA,oC;MAAA,gC;MAA
A,gD;QAYoB,UAA8B,M;QAF9C,YAAY,C;QACZ,kBAAkB,O;QAClB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,U
AAgB,SAAhB,O;UAAsB,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAnB,EAAgC,oBAAhC,C;;QACpC,
OAAO,W;O;KAbX,C;wFAgBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kBAAkB,O;Q
AClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAI,oBAAJ,OAAV,EAAwB,WAAxB,C;;QAE
lB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kBAAkB,O;
QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAI,oBAAJ,OAAV,EAAwB,WAAxB,C;;Q
AElB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kBAAkB,
O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAI,oBAAJ,OAAV,EAAwB,WAAxB,C;;
QAElB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kBAAk
B,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAI,oBAAJ,OAAV,EAAwB,WAAxB,
C;;QAElB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kBA

AkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAI,oBAAJ,OAAV,EAAwB,WAAx
B,C;;QAElB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kB
AAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAI,oBAAJ,OAAV,EAAwB,WAA
xB,C;;QAElB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,k
BAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAI,oBAAJ,OAAV,EAAwB,WA
AxB,C;;QAElB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,
kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,YAAJ,EAAI,oBAAJ,OAAV,EAAwB,W
AAxB,C;;QAElB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,8D;MAAA,oC;MAAA,gD;QAYoC,Q;QAHhC,YAA
Y,wB;QACZ,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,sBAAI,YAAJ,EAAI,oBAAJ,QAA
V,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAdX,C;sGAiBA,yB;MAAA,8D;MAAA,gD;QAUI,YAAY,wB;QAC
Z,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,
WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;wGAmBA,yB;MAAA,8D;MAAA,gD;QAUI,YAAY,wB;QA
CZ,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6
B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;wGAmBA,yB;MAAA,8D;MAAA,gD;QAUI,YAAY,wB;Q
ACZ,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA
6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;wGAmBA,yB;MAAA,8D;MAAA,gD;QAUI,YAAY,wB;
QACZ,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EA
A6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;wGAmBA,yB;MAAA,8D;MAAA,gD;QAUI,YAAY,w
B;QACZ,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,
EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;wGAmBA,yB;MAAA,8D;MAAA,gD;QAUI,YAAY
,wB;QACZ,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAj
B,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;wGAmBA,yB;MAAA,8D;MAAA,gD;QAUI,YA
AY,wB;QACZ,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,C
AAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;wGAmBA,yB;MAAA,8D;MAAA,gD;QAUI,
YAAY,wB;QACZ,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ
,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;wGAmBA,yB;MAAA,8D;MAAA,oC;MA
AA,gD;QAUI,YAAY,wB;QACZ,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAAi
B,sBAAI,KAAJ,EAAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAhBX,C;oFAmBA,6B;MAIoB,Q;M
AAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,OAAO,OAAP,C;;K;sFAG1B,6B;MAIoB,Q;MAA
hB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,OAAO,OAAP,C;;K;sFAG1B,6B;MAIoB,Q;MAAhB,
wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,OAAO,OAAP,C;;K;sFAG1B,6B;MAIoB,Q;MAAhB,wB
AAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,OAAO,OAAP,C;;K;sFAG1B,6B;MAIoB,Q;MAAhB,wBAA
gB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,OAAO,OAAP,C;;K;sFAG1B,6B;MAIoB,Q;MAAhB,wBAAgB,
SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,OAAO,OAAP,C;;K;sFAG1B,6B;MAIoB,Q;MAAhB,wBAAgB,SA
AhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,OAAO,OAAP,C;;K;sFAG1B,6B;MAIoB,Q;MAAhB,wBAAgB,SAAh
B,gB;QAAgB,cAAA,SAAhB,M;QAAsB,OAAO,OAAP,C;;K;sFAG1B,yB;MAAA,oC;MAAA,gC;MAAA,oC;QAI
oB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UAAsB,OAAO,oBAAP,C;;O;KAJ1B,C;
kGAOA,6B;MAOiB,UAAa,M;MAD1B,YAAY,C;MACZ,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAmB,QA
AO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;K;oGAGvB,6B;MAOiB,UAAa,M;MAD1B,YAAY,C;MACZ,wBA
Aa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAmB,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;K;oGAGvB,6B;
MAOiB,UAAa,M;MAD1B,YAAY,C;MACZ,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAmB,QAAO,cAAP,E
AAO,sBAAP,WAAgB,IAAhB,C;;K;oGAGvB,6B;MAOiB,UAAa,M;MAD1B,YAAY,C;MACZ,wBAAa,SAAb,gB;
QAAa,WAAA,SAAb,M;QAAmB,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;K;oGAGvB,6B;MAOiB,UAA
a,M;MAD1B,YAAY,C;MACZ,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAmB,QAAO,cAAP,EAAO,sBAAP,
WAAgB,IAAhB,C;;K;oGAGvB,6B;MAOiB,UAAa,M;MAD1B,YAAY,C;MACZ,wBAAa,SAAb,gB;QAAa,WAA
A,SAAb,M;QAAmB,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;K;oGAGvB,6B;MAOiB,UAAa,M;MAD1B
,YAAY,C;MACZ,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;QAAmB,QAAO,cAAP,EAAO,sBAAP,WAAgB,IA
AhB,C;;K;oGAGvB,6B;MAOiB,UAAa,M;MAD1B,YAAY,C;MACZ,wBAAa,SAAb,gB;QAAa,WAAA,SAAb,M;

QAAmB,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;K;oGAGvB,yB;MAAA,oC;MAAA,gC;MAAA,oC;QA
OiB,UAAa,M;QAD1B,YAAY,C;QACZ,wBAAa,SAAb,gB;UAAa,WAAb,UAAa,SAAb,O;UAAmB,QAAO,cAAP,
EAAO,sBAAP,WAAgB,iBAAhB,C;;O;KAPvB,C;IAUA,wB;MAaiB,Q;MAFb,IApsLO,qBAAQ,CAosLf,C;QAAe,
MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;
QACR,MmB/1aG,MAAO,KnB+1aE,GmB/1aF,EnB+1aO,CmB/1aP,C;;MnBi2ad,OAAO,G;K;IAGX,0B;MAaiB,Q;
MAFb,IAxtLO,qBAAQ,CAwtLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,
CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmB93aG,MAAO,KnB83aE,GmB93aF,EnB83aO,CmB93aP,C;;
MnBg4ad,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IA1uLO,qBAAQ,CA0uLf,C;QAAe,MAAM,6B;MACrB,UAA
U,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,sBAAM,CA
AN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IApvLO,qBAAQ,CAovLf,C;Q
AAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAA
L,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IA9vLO,
qBAAQ,CA8vLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI
,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWi
B,Q;MAFb,IAxwLO,qBAAQ,CAwwLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,a
AAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO
,G;K;IAGX,0B;MAWiB,Q;MAFb,IAlxLO,qBAAQ,CAkxLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,
C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,oBAAM,CAAN,KAAJ,C;UAA
a,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAaiB,Q;MAFb,IA9xLO,qBAAQ,CA8xLf,C;QAAe,MAAM,6B;MA
CrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmB5+a
G,MAAO,KnB4+aE,GmB5+aF,EnB4+aO,CmB5+aP,C;;MnB8+ad,OAAO,G;K;IAGX,0B;MAaiB,Q;MAFb,IA1yL
O,qBAAQ,CA0yLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QA
CI,QAAQ,UAAK,CAAL,C;QACR,MmBr/aG,MAAO,KnBq/aE,GmBr/aF,EnBq/aO,CmBr/aP,C;;MnBu/ad,OAAO,
G;K;IAGX,0B;MAWiB,Q;MAFb,IA5yLO,qBAAQ,CA4yLf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;
MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAA
M,C;;MAEvB,OAAO,G;K;gFAGX,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAh4LO,qBAAQ,CAg4Lf,C;UA
Ae,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;U
AAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAA
L,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;Q
AGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAn5LO,qBAAQ,CAm5Lf,C
;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,
C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,C
AAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;
;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAt6LO,qBAAQ,CAs6Lf,
C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB
,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,
CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,
C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAz7LO,qBAAQ,CAy7
Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CA
AjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UA
AK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAA
W,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA58LO,qBAAQ,C
A48Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,
CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,
UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,W
AAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA/9LO,qBAAQ,
CA+9Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAA
a,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ

,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,
WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAl/LO,qBAA
Q,CAk/Lf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cA
Aa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAA
Q,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV
,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IArgMO,qBA
AQ,CAqgMf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,
cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,Q
AAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YA
CV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,oC;MAAA,sC;QAW
I,IAxhMO,qBAAQ,CAwhMf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,
C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,S
AAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YA
CI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;4FA2BA,yB;MAAA,8D;MAAA,sC;QAOI,IA/m
MO,qBAAQ,CA+mMf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACr
B,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;
UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAA
U,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA9nMO,qBA
AQ,CA8nMf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cA
Aa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAA
Q,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV
,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA7oMO,qBAAQ,CA6o
Mf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAj
B,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK
,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,
C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA5pMO,qBAAQ,CA4pMf,C;UAA
e,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAo
B,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;
UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGn
B,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA3qMO,qBAAQ,CA2qMf,C;UAAe,OAAO,I
;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O
;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAA
Q,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;
O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IA1rMO,qBAAQ,CA0rMf,C;UAAe,OAAO,I;QACtB,cA
Ac,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,e
AAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,
CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApB
X,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAzsMO,qBAAQ,CAysMf,C;UAAe,OAAO,I;QACtB,cAAc,UAA
K,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SA
AS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;
UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAu
BA,yB;MAAA,8D;MAAA,sC;QAOI,IAxtMO,qBAAQ,CAwtMf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C
;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C
;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI
,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAA
A,8D;MAAA,oC;MAAA,sC;QAOI,IAvuMO,qBAAQ,CAuuMf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;
QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,
C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAA

I,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;gFAuBA,yB;MAAA,sE;MAAA,8D;MmBt9bA,iB;MnBs9bA,sC;QAeiB,Q;QAFb,IAp0MO,qBAAQ,CAo0Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB/9bG,MAAO,KnB+9bO,QmB/9bP,EnB+9biB,CmB/9bjB,C;;QnBi+bd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB5+bA,iB;MnB4+bA,sC;QAeiB,Q;QAFb,IAl1MO,qBAAQ,CAk1Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBr/bG,MAAO,KnBq/bO,QmBr/bP,EnBq/biB,CmBr/bjB,C;;QnBu/bd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBlgcA,iB;MnBkgcA,sC;QAeiB,Q;QAFb,IAh2MO,qBAAQ,CAg2Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB3gcG,MAAO,KnB2gcO,QmB3gcP,EnB2gciB,CmB3gcjB,C;;QnB6gcd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBxhcA,iB;MnBwhcA,sC;QAeiB,Q;QAFb,IA92MO,qBAAQ,CA82Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBjicG,MAAO,KnBiicO,QmBjicP,EnBiiciB,CmBjicjB,C;;QnBmicd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB9icA,iB;MnB8icA,sC;QAeiB,Q;QAFb,IA53MO,qBAAQ,CA43Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBvjcG,MAAO,KnBujcO,QmBvjcP,EnBujciB,CmBvjcjB,C;;QnByjcd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBpkcA,iB;MnBokcA,sC;QAeiB,Q;QAFb,IA14MO,qBAAQ,CA04Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB7kcG,MAAO,KnB6kcO,QmB7kcP,EnB6kciB,CmB7kcjB,C;;QnB+kcd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB1lcA,iB;MnB0lcA,sC;QAeiB,Q;QAFb,IAx5MO,qBAAQ,CAw5Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBnmcG,MAAO,KnBmmcO,QmBnmcP,EnBmmciB,CmBnmcjB,C;;QnBqmcd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBhncA,iB;MnBgncA,sC;QAeiB,Q;QAFb,IAt6MO,qBAAQ,CAs6Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBzncG,MAAO,KnByncO,QmBzncP,EnBynciB,CmBzncjB,C;;QnB2ncd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MmBtocA,iB;MnBsocA,sC;QAeiB,Q;QAFb,IAp7MO,qBAAQ,CAo7Mf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmB/ocG,MAAO,KnB+ocO,QmB/ocP,EnB+ociB,CmB/ocjB,C;;QnBipcd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB5pcA,iB;MnB4pcA,sC;QAeiB,Q;QAFb,IA1gNO,qBAAQ,CA0gNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBrqcG,MAAO,KnBqqcO,QmBrqcP,EnBqqciB,CmBrqcjB,C;;QnBuqcd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB7rcA,iB;MnB6rcA,sC;QAeiB,Q;QAFb,IAxhNO,qBAAQ,CAwhNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBtscG,MAAO,KnBsscO,QmBtscP,EnBssciB,CmBtscjB,C;;QnBwscd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBntcA,iB;MnBmtcA,sC;QAeiB,Q;QAFb,IAtiNO,qBAAQ,CAsiNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB5tcG,MAAO,KnB4tcO,QmB5tcP,EnB4tciB,CmB5tcjB,C;;QnB8tcd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBzucA,iB;MnByucA,sC;QAeiB,Q;QAFb,IApjNO,qBAAQ,CAojNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBlvcG,MAAO,KnBkvcO,QmBlvcP,EnBkvciB,CmBlvcjB,C;;QnBovcd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmB/vcA,iB;MnB+vcA,sC;QAeiB,Q;QAFb,IAlkNO,qBAAQ,CAkkNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBxwcG,MAAO,KnBwwcO,QmBxwcP,EnBwwciB,CmBxwcjB,C;;QnB0wcd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBrxcA,iB;MnBqxcA,sC;QAeiB,Q;QAFb,IAhlNO,qBAAQ,CAglNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,W

mB9xcG,MAAO,KnB8xcO,QmB9xcP,EnB8xciB,CmB9xcjB,C;;QnBgycd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmB3ycA,iB;MnB2ycA,sC;QAeiB,Q;QAFb,IA9lNO,qBAAQ,CA8lNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBpzcG,MAAO,KnBozcO,QmBpzcP,EnBozciB,CmBpzcjB,C;;QnBszcd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBj0cA,iB;MnBi0cA,sC;QAeiB,Q;QAFb,IA5mNO,qBAAQ,CA4mNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB10cG,MAAO,KnB00cO,QmB10cP,EnB00ciB,CmB10cjB,C;;QnB40cd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MmBv1cA,iB;MnBu1cA,sC;QAeiB,Q;QAFb,IA1nNO,qBAAQ,CA0nNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmBh2cG,MAAO,KnBg2cO,QmBh2cP,EnBg2ciB,CmBh2cjB,C;;QnBk2cd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA9sNO,qBAAQ,CA8sNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA5tNO,qBAAQ,CA4tNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA1uNO,qBAAQ,CA0uNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAxvNO,qBAAQ,CAwvNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAtwNO,qBAAQ,CAswNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IApxNO,qBAAQ,CAoxNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA9zNO,qBAAQ,CA8zNf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;4FAsBA,yB;MAAA,8D;MmBxidA,iB;MnBwidA,sC;QAaiB,Q;QAFb,IAp5NO,qBAAQ,CAo5Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB/idG,MAAO,KnB+idO,QmB/idP,EnB+idiB,CmB/idjB,C;;QnBijdd,OAAO,Q;O;KAjBX,C;4FAoBA,yB;MAAA,8D;MmB5jdA,iB;MnB4jdA,sC;QAaiB,Q;QAFb,IAh6NO,qBAAQ,CAg6Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBnkdG,MAAO,KnBmkdO,QmBnkdP,EnBmkdiB,CmBnkdjB,C;;QnBqkdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBhldA,iB;MnBgldA,sC;QAaiB,Q;QAFb,IA56NO,qBAAQ,CA46Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBvldG,MAAO,KnBuldO,QmBvldP,EnBuldiB,CmBvldjB,C;;QnByldd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBpmdA,iB;MnBomdA,sC;QAaiB,Q;QAFb,IAx7NO,qBAAQ,CAw7Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,

CAAL,CAAT,C;UACR,WmB3mdG,MAAO,KnB2mdO,QmB3mdP,EnB2mdiB,CmB3mdjB,C;;QnB6mdd,OAAO,
Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBxndA,iB;MnBwndA,sC;QAaiB,Q;QAFb,IAp8NO,qBAAQ,CAo8Nf,
C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,
SAAS,UAAK,CAAL,CAAT,C;UACR,WmB/ndG,MAAO,KnB+ndO,QmB/ndP,EnB+ndiB,CmB/ndjB,C;;QnBiodd,
OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB5odA,iB;MnB4odA,sC;QAaiB,Q;QAFb,IAh9NO,qBAAQ,C
Ag9Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,
QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBnpdG,MAAO,KnBmpdO,QmBnpdP,EnBmpdiB,CmBnpdjB,C;
;QnBqpdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBhqdA,iB;MnBgqdA,sC;QAaiB,Q;QAFb,IA59NO,
qBAAQ,CA49Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV
,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBvqdG,MAAO,KnBuqdO,QmBvqdP,EnBuqdiB,CmB
vqdjB,C;;QnByqdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBprdA,iB;MnBordA,sC;QAaiB,Q;QAFb,I
Ax+NO,qBAAQ,CAw+Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAA
U,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB3rdG,MAAO,KnB2rdO,QmB3rdP,EnB2rd
iB,CmB3rdjB,C;;QnB6rdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,oC;MAAA,8D;MmBxsdA,iB;MnBwsdA,s
C;QAaiB,Q;QAFb,IAp/NO,qBAAQ,CAo/Nf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QA
CF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmB/sdG,MAAO,KnB+sd
O,QmB/sdP,EnB+sdiB,CmB/sdjB,C;;QnBitdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBvudA,iB;MnB
uudA,sC;QAaiB,Q;QAFb,IAxkOO,qBAAQ,CAwkOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAA
T,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB9udG,MAAO,
KnB8udO,QmB9udP,EnB8udiB,CmB9udjB,C;;QnBgvdd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB3v
dA,iB;MnB2vdA,sC;QAaiB,Q;QAFb,IAplOO,qBAAQ,CAolOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CA
AL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBlwdG,
MAAO,KnBkwdO,QmBlwdP,EnBkwdiB,CmBlwdjB,C;;QnBowdd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8
D;MmB/wdA,iB;MnB+wdA,sC;QAaiB,Q;QAFb,IAhmOO,qBAAQ,CAgmOf,C;UAAe,OAAO,I;QACtB,eAAe,SA
AS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UA
CR,WmBtxdG,MAAO,KnBsxdO,QmBtxdP,EnBsxdiB,CmBtxdjB,C;;QnBwxdd,OAAO,Q;O;KAjBX,C;+FAoBA,y
B;MAAA,8D;MmBnydA,iB;MnBmydA,sC;QAaiB,Q;QAFb,IA5mOO,qBAAQ,CA4mOf,C;UAAe,OAAO,I;QACt
B,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,C
AAT,C;UACR,WmB1ydG,MAAO,KnB0ydO,QmB1ydP,EnB0ydiB,CmB1ydjB,C;;QnB4ydd,OAAO,Q;O;KAjBX,
C;+FAoBA,yB;MAAA,8D;MmBvzdA,iB;MnBuzdA,sC;QAaiB,Q;QAFb,IAxnOO,qBAAQ,CAwnOf,C;UAAe,OA
AO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAA
K,CAAL,CAAT,C;UACR,WmB9zdG,MAAO,KnB8zdO,QmB9zdP,EnB8zdiB,CmB9zdjB,C;;QnBg0dd,OAAO,Q;
O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmB30dA,iB;MnB20dA,sC;QAaiB,Q;QAFb,IApoOO,qBAAQ,CAooOf,C;
UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,S
AAS,UAAK,CAAL,CAAT,C;UACR,WmBl1dG,MAAO,KnBk1dO,QmBl1dP,EnBk1diB,CmBl1djB,C;;QnBo1dd,
OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmB/1dA,iB;MnB+1dA,sC;QAaiB,Q;QAFb,IAhpOO,qBAAQ,C
AgpOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,
QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBt2dG,MAAO,KnBs2dO,QmBt2dP,EnBs2diB,CmBt2djB,C;;Q
nBw2dd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBn3dA,iB;MnBm3dA,sC;QAaiB,Q;QAFb,IA5pOO,q
BAAQ,CA4pOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,i
B;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB13dG,MAAO,KnB03dO,QmB13dP,EnB03diB,CmB1
3djB,C;;QnB43dd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,oC;MAAA,8D;MmBv4dA,iB;MnBu4dA,sC;QAaiB
,Q;QAFb,IAxqOO,qBAAQ,CAwqOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;
QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmB94dG,MAAO,KnB84dO,QmB
94dP,EnB84diB,CmB94djB,C;;QnBg5dd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;Q
AFb,IA1vOO,qBAAQ,CA0vOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,
aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WA
AW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAtwOO,qBAAQ,C

AswOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,
QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,
Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAlxOO,qBAAQ,CAkxOf,C;UAAe,OAAO,I
;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CA
AL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,
yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA9xOO,qBAAQ,CA8xOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,U
AAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IA
AI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,s
C;QAWiB,Q;QAFb,IA1yOO,qBAAQ,CA0yOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;Q
ACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KA
AJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAt
zOO,qBAAQ,CAszOf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,C
AAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;
QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAl0OO,qBAAQ,CAk0Of,C
;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,S
AAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAj
BX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA90OO,qBAAQ,CA80Of,C;UAAe,OAAO,I;QACtB,
eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAA
T,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAA
A,oC;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IA11OO,qBAAQ,CA01Of,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,
sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,
IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,sE;MAA
A,8D;MAAA,kD;QAaiB,Q;QAFb,IAh7OO,qBAAQ,CAg7Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,C
AAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UA
AW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;
0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA97OO,qBAAQ,CA87Of,C;UAAe,MAAM,6B;Q
ACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAA
L,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QA
GnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA58OO,qBAAQ,C
A48Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UA
CI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,C
AAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAai
B,Q;QAFb,IA19OO,qBAAQ,CA09Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+
B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAA
kB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;
MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAx+OO,qBAAQ,CAw+Of,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UA
AK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAA
I,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAn
BX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAt/OO,qBAAQ,CAs/Of,C;UAAe,MAAM,
6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,
CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;
;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IApgPO,qBAA
Q,CAogPf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;
UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC
,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QA
aiB,Q;QAFb,IAlhPO,qBAAQ,CAkhPf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,
+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EA
AkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,s

E;MAAA,oC;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAhiPO,qBAAQ,CAgiPf,C;UAAe,MAAM,6B;QACrB,eAAe
,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C
;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAA
O,Q;O;KAnBX,C;oGAsBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IApnPO,qBAAQ,CAonPf,C;UAAe,OAA
O,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,
CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;
;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAhoPO,qBAAQ,CAgoPf,
C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,
SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YA
CI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA5oPO,qB
AAQ,CA4oPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;
UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC
,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QA
Fb,IAxpPO,qBAAQ,CAwpPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,a
AAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB
,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD
;QAWiB,Q;QAFb,IApqPO,qBAAQ,CAoqPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QAC
F,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,E
AAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,
8D;MAAA,kD;QAWiB,Q;QAFb,IAhrPO,qBAAQ,CAgrPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,
CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,S
AAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAo
BA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA5rPO,qBAAQ,CA4rPf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,
UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,I
AAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;K
AjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAxsPO,qBAAQ,CAwsPf,C;UAAe,OAAO,I;QACt
B,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,C
AAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGn
B,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,oC;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAptPO,qBAAQ,CAot
Pf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QA
AQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,
C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;IAoBA,8B;MASiB,Q;MAFb,IAtyPO,qBAAQ,CAsyPf,C;QA
Ae,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;
QACR,MmBj8eG,MAAO,KnBi8eE,GmBj8eF,EnBi8eO,CmBj8eP,C;;MnBm8ed,OAAO,G;K;IAGX,gC;MASiB,Q;
MAFb,IAtzPO,qBAAQ,CAszPf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CA
AV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmB59eG,MAAO,KnB49eE,GmB59eF,EnB49eO,CmB59eP,C;;Mn
B89ed,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAp0PO,qBAAQ,CAo0Pf,C;QAAe,OAAO,I;MACtB,UAAU,UAA
K,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,sBAAM,CAAN,KA
AJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA10PO,qBAAQ,CA00Pf,C;QAAe,OA
AO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,
IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAh1PO,qBAAQ,CA
g1Pf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAA
K,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,I
At1PO,qBAAQ,CAs1Pf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;Q
ACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;M
AOiB,Q;MAFb,IA51PO,qBAAQ,CA41Pf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,a
AAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,oBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,
OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IAp2PO,qBAAQ,CAo2Pf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAA

L,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBljfG,MAAO,KnBkjfE,GmBlj
fF,EnBkjfO,CmBljfP,C;;MnBojfd,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IA52PO,qBAAQ,CA42Pf,C;QAAe,OA
AO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,
MmBvjfG,MAAO,KnBujfE,GmBvjfF,EnBujfO,CmBvjfP,C;;MnByjfd,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA
12PO,qBAAQ,CA02Pf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;Q
ACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,wC;M
AWiB,Q;MAFb,IA57PO,qBAAQ,CA47Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MA
Ab,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GA
A6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAt8PO,qBAAQ,CAs8Pf,C;
QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CA
AL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,O
AAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAh9PO,qBAAQ,CAg9Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,C
AAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,E
AAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IA1
9PO,qBAAQ,CA09Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;
QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAA
oC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAp+PO,qBAAQ,CAo+Pf,C;QAAe,MAAM,6B;
MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,
UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0
C;MAWiB,Q;MAFb,IA9+PO,qBAAQ,CA8+Pf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;
MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,
GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAx/PO,qBAAQ,CAw/P
f,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,
CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9
C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAlgQO,qBAAQ,CAkgQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAA
K,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAA
R,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,I
A5gQO,qBAAQ,CA4gQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,
iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CAAjC,C;U
AAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,8C;MAOiB,Q;MAFb,IA1lQO,qBAAQ,CA0lQf,C;QAAe,OAAO,I;
MACtB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI
,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,g
D;MAOiB,Q;MAFb,IAhmQO,qBAAQ,CAgmQf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;M
AAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,G
AA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAtmQO,qBAAQ,CAsmQ
f,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,C
AAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,
OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IA5mQO,qBAAQ,CA4mQf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,C
AAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,E
AAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAln
QO,qBAAQ,CAknQf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QA
CI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,
MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAxnQO,qBAAQ,CAwnQf,C;QAAe,OAAO,I;MACt
B,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAA
W,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MA
OiB,Q;MAFb,IA9nQO,qBAAQ,CA8nQf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,a
AAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,
CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IApoQO,qBAAQ,CAooQf,C;QA

Ae,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;
QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,
G;K;IAGX,gD;MAOiB,Q;MAFb,IA1oQO,qBAAQ,CA0oQf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;
MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cA
Ab,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,wB;MAaiB,Q;MAFb,IA9tQO,qBA
AQ,CA8tQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QA
AQ,UAAK,CAAL,C;QACR,MmBrqfG,MAAO,KnBqqfE,GmBrqfF,EnBqqfO,CmBrqfP,C;;MnBuqfd,OAAO,G;K;I
AGX,0B;MAaiB,Q;MAFb,IAlvQO,qBAAQ,CAkvQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MAC
G,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBpsfG,MAAO,KnBosfE,GmBpsfF,EnB
osfO,CmBpsfP,C;;MnBssfd,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IApwQO,qBAAQ,CAowQf,C;QAAe,MAA
M,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QAC
R,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IA9wQO,q
BAAQ,CA8wQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI
,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWi
B,Q;MAFb,IAxxQO,qBAAQ,CAwxQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,a
AAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO
,G;K;IAGX,0B;MAWiB,Q;MAFb,IAlyQO,qBAAQ,CAkyQf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,
C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAA
M,C;;MAEvB,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IA5yQO,qBAAQ,CA4yQf,C;QAAe,MAAM,6B;MACrB,
UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,oBAAM
,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0B;MAaiB,Q;MAFb,IAxzQO,qBAAQ,CAwzQf,C
;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,C
AAL,C;QACR,MmBlzfG,MAAO,KnBkzfE,GmBlzfF,EnBkzfO,CmBlzfP,C;;MnBozfd,OAAO,G;K;IAGX,0B;MAa
iB,Q;MAFb,IAp0QO,qBAAQ,CAo0Qf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,a
AAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmB3zfG,MAAO,KnB2zfE,GmB3zfF,EnB2zfO,CmB3zfP
,C;;MnB6zfd,OAAO,G;K;IAGX,0B;MAWiB,Q;MAFb,IAt0QO,qBAAQ,CAs0Qf,C;QAAe,MAAM,6B;MACrB,U
AAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,C
AAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;gFAGX,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA15QO,qB
AAQ,CA05Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAA
I,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,
QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;Y
ACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA76QO
,qBAAQ,CA66Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,I
AAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UA
CI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,
C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IAh8
QO,qBAAQ,CAg8Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACr
B,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;
UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAA
U,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAWI,IA
n9QO,qBAAQ,CAm9Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;Q
ACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,
M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,U
AAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAW
I,IAt+QO,qBAAQ,CAs+Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;
QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAA
b,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,
UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QA

WI,IAz/QO,qBAAQ,CAy/Qf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,
C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,S
AAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YA
CI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,sC;
QAWI,IA5gRO,qBAAQ,CA4gRf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SA
AK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAA
a,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;
YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAAA,s
C;QAWI,IA/hRO,qBAAQ,CA+hRf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,S
AAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OA
Aa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,
C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;kFA2BA,yB;MAAA,sE;MAAA,8D;MAA
A,oC;MAAA,sC;QAWI,IAljRO,qBAAQ,CAkjRf,C;UAAe,MAAM,6B;QACrB,cAAc,UAAK,CAAL,C;QACd,gBA
AqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aA
AU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,
CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;4FA2BA,yB;MAAA,8D;M
AAA,sC;QAOI,IAzoRO,qBAAQ,CAyoRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAA
L,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,
OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KA
AJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QA
OI,IAxpRO,qBAAQ,CAwpRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;
QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAA
b,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,
UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAvqRO,
qBAAQ,CAuqRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI
,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,Q
AAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YA
CV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAtrRO,qBAAQ,CAsr
Rf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB
,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,
CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,
C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IArsRO,qBAAQ,CAqsRf,C;UAAe,
OAAO,I;QACtB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,
OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UA
CR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,O
AAO,O;O;KApBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAptRO,qBAAQ,CAotRf,C;UAAe,OAAO,I;QAC
tB,cAAc,UAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC
3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SA
AS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KA
pBX,C;8FAuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAnuRO,qBAAQ,CAmuRf,C;UAAe,OAAO,I;QACtB,cAAc,UA
AK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,S
AAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,
C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8F
AuBA,yB;MAAA,8D;MAAA,sC;QAOI,IAlvRO,qBAAQ,CAkvRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,
C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,
C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IA
AI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;8FAuBA,yB;M
AAA,8D;MAAA,oC;MAAA,sC;QAOI,IAjwRO,qBAAQ,CAiwRf,C;UAAe,OAAO,I;QACtB,cAAc,UAAK,CAAL,

C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAA
T,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,UAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,I
AAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;gFAuBA,yB;
MAAA,sE;MAAA,8D;MmB5xgBA,iB;MnB4xgBA,sC;QAeiB,Q;QAFb,IA91RO,qBAAQ,CA81Rf,C;UAAe,MAA
M,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAA
K,CAAL,CAAT,C;UACR,WmBrygBG,MAAO,KnBqygBO,QmBrygBP,EnBqygBiB,CmBrygBjB,C;;QnBuygBd,O
AAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBlzgBA,iB;MnBkzgBA,sC;QAeiB,Q;QAFb,IA52RO
,qBAAQ,CA42Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CA
AV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB3zgBG,MAAO,KnB2zgBO,QmB3zgBP,EnB2zg
BiB,CmB3zgBjB,C;;QnB6zgBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBx0gBA,iB;MnB
w0gBA,sC;QAeiB,Q;QAFb,IA13RO,qBAAQ,CA03Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,C
AAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBj1gBG,MA
AO,KnBi1gBO,QmBj1gBP,EnBi1gBiB,CmBj1gBjB,C;;QnBm1gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,s
E;MAAA,8D;MmB91gBA,iB;MnB81gBA,sC;QAeiB,Q;QAFb,IAx4RO,qBAAQ,CAw4Rf,C;UAAe,MAAM,6B;Q
ACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAA
L,CAAT,C;UACR,WmBv2gBG,MAAO,KnBu2gBO,QmBv2gBP,EnBu2gBiB,CmBv2gBjB,C;;QnBy2gBd,OAAO,
Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBp3gBA,iB;MnBo3gBA,sC;QAeiB,Q;QAFb,IAt5RO,qBA
AQ,CAs5Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,i
B;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB73gBG,MAAO,KnB63gBO,QmB73gBP,EnB63gBiB,
CmB73gBjB,C;;QnB+3gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmB14gBA,iB;MnB04gB
A,sC;QAeiB,Q;QAFb,IAp6RO,qBAAQ,CAo6Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,
C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBn5gBG,MAAO,K
nBm5gBO,QmBn5gBP,EnBm5gBiB,CmBn5gBjB,C;;QnBq5gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;
MAAA,8D;MmBh6gBA,iB;MnBg6gBA,sC;QAeiB,Q;QAFb,IAl7RO,qBAAQ,CAk7Rf,C;UAAe,MAAM,6B;QACr
B,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,C
AAT,C;UACR,WmBz6gBG,MAAO,KnBy6gBO,QmBz6gBP,EnBy6gBiB,CmBz6gBjB,C;;QnB26gBd,OAAO,Q;O
;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBt7gBA,iB;MnBs7gBA,sC;QAeiB,Q;QAFb,IAh8RO,qBAAQ,
CAg8Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;U
ACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB/7gBG,MAAO,KnB+7gBO,QmB/7gBP,EnB+7gBiB,Cm
B/7gBjB,C;;QnBi8gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MmB58gBA,iB;Mn
B48gBA,sC;QAeiB,Q;QAFb,IA98RO,qBAAQ,CA88Rf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL
,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmBr9gBG,
MAAO,KnBq9gBO,QmBr9gBP,EnBq9gBiB,CmBr9gBjB,C;;QnBu9gBd,OAAO,Q;O;KAnBX,C;kFAsBA,yB;MA
AA,sE;MAAA,8D;MmB7+gBA,iB;MnB6+gBA,sC;QAeiB,Q;QAFb,IApiSO,qBAAQ,CAoiSf,C;UAAe,MAAM,6B
;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CA
AL,CAAT,C;UACR,WmBt/gBG,MAAO,KnBs/gBO,QmBt/gBP,EnBs/gBiB,CmBt/gBjB,C;;QnBw/gBd,OAAO,Q;
O;KAnBX,C;kFAsBA,yB;MAAA,sE;MAAA,8D;MmBnghBA,iB;MnBmghBA,sC;QAeiB,Q;QAFb,IAljSO,qBAA
Q,CAkjSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;
UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB5ghBG,MAAO,KnB4ghBO,QmB5ghBP,EnB4ghBiB,C
mB5ghBjB,C;;QnB8ghBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBzhhBA,iB;MnByhhB
A,sC;QAeiB,Q;QAFb,IAhkSO,qBAAQ,CAgkSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,
C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBlihBG,MAAO,Kn
BkihBO,QmBlihBP,EnBkihBiB,CmBlihBjB,C;;QnBoihBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAA
A,8D;MmB/ihBA,iB;MnB+ihBA,sC;QAeiB,Q;QAFb,IA9kSO,qBAAQ,CA8kSf,C;UAAe,MAAM,6B;QACrB,eAA
e,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;
UACR,WmBxjhBG,MAAO,KnBwjhBO,QmBxjhBP,EnBwjhBiB,CmBxjhBjB,C;;QnB0jhBd,OAAO,Q;O;KAnBX,
C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBrkhBA,iB;MnBqkhBA,sC;QAeiB,Q;QAFb,IA5lSO,qBAAQ,CA4lSf,C
;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAA

Q,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB9khBG,MAAO,KnB8khBO,QmB9khBP,EnB8khBiB,CmB9khBjB,
C;;QnBglhBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmB3lhBA,iB;MnB2lhBA,sC;QAeiB,
Q;QAFb,IA1mSO,qBAAQ,CA0mSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B
;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBpmhBG,MAAO,KnBomhBO,
QmBpmhBP,EnBomhBiB,CmBpmhBjB,C;;QnBsmhBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8
D;MmBjnhBA,iB;MnBinhBA,sC;QAeiB,Q;QAFb,IAxnSO,qBAAQ,CAwnSf,C;UAAe,MAAM,6B;QACrB,eAAe,S
AAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;U
ACR,WmB1nhBG,MAAO,KnB0nhBO,QmB1nhBP,EnB0nhBiB,CmB1nhBjB,C;;QnB4nhBd,OAAO,Q;O;KAnBX,
C;mFAsBA,yB;MAAA,sE;MAAA,8D;MmBvohBA,iB;MnBuohBA,sC;QAeiB,Q;QAFb,IAtoSO,qBAAQ,CAsoSf,
C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QA
AQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBhphBG,MAAO,KnBgphBO,QmBhphBP,EnBgphBiB,CmBhphBj
B,C;;QnBkphBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MmB7phBA,iB;MnB6ph
BA,sC;QAeiB,Q;QAFb,IAppSO,qBAAQ,CAopSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAA
T,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmBtqhBG,MAAO
,KnBsqhBO,QmBtqhBP,EnBsqhBiB,CmBtqhBjB,C;;QnBwqhBd,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;
MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAxuSO,qBAAQ,CAwuSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UA
AK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAA
I,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8
D;MAAA,sC;QAaiB,Q;QAFb,IAtvSO,qBAAQ,CAsvSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,
CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,
CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,s
C;QAaiB,Q;QAFb,IApwSO,qBAAQ,CAowSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;
QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,K
AAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,
Q;QAFb,IAlxSO,qBAAQ,CAkxSf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;
QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YA
CI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,I
AhySO,qBAAQ,CAgySf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aA
AU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW
,C;;;QAGnB,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA9ySO,qB
AAQ,CA8ySf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,
iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGn
B,OAAO,Q;O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA5zSO,qBAAQ,CA4z
Sf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,Q
AAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;
O;KAnBX,C;mFAsBA,yB;MAAA,sE;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IA10SO,qBAAQ,CA00Sf,C;UAAe,
MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS
,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,
C;mFAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MAAA,sC;QAaiB,Q;QAFb,IAx1SO,qBAAQ,CAw1Sf,C;UAAe
,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAA
S,sBAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnB
X,C;4FAsBA,yB;MAAA,8D;MmB92hBA,iB;MnB82hBA,sC;QAaiB,Q;QAFb,IA96SO,qBAAQ,CA86Sf,C;UAAe,
OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,U
AAK,CAAL,CAAT,C;UACR,WmBr3hBG,MAAO,KnBq3hBO,QmBr3hBP,EnBq3hBiB,CmBr3hBjB,C;;QnBu3hB
d,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBl4hBA,iB;MnBk4hBA,sC;QAaiB,Q;QAFb,IA17SO,qBAA
Q,CA07Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UA
CI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBz4hBG,MAAO,KnBy4hBO,QmBz4hBP,EnBy4hBiB,CmB
z4hBjB,C;;QnB24hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBt5hBA,iB;MnBs5hBA,sC;QAaiB,Q;Q

AFb,IAt8SO,qBAAQ,CAs8Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,a
AAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB75hBG,MAAO,KnB65hBO,QmB75hB
P,EnB65hBiB,CmB75hBjB,C;;QnB+5hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB16hBA,iB;MnB0
6hBA,sC;QAaiB,Q;QAFb,IAl9SO,qBAAQ,CAk9Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,
C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBj7hBG,MAAO,K
nBi7hBO,QmBj7hBP,EnBi7hBiB,CmBj7hBjB,C;;QnBm7hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;Mm
B97hBA,iB;MnB87hBA,sC;QAaiB,Q;QAFb,IA99SO,qBAAQ,CA89Sf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UA
AK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,Wm
Br8hBG,MAAO,KnBq8hBO,QmBr8hBP,EnBq8hBiB,CmBr8hBjB,C;;QnBu8hBd,OAAO,Q;O;KAjBX,C;8FAoBA,
yB;MAAA,8D;MmBl9hBA,iB;MnBk9hBA,sC;QAaiB,Q;QAFb,IA1+SO,qBAAQ,CA0+Sf,C;UAAe,OAAO,I;QAC
tB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,C
AAT,C;UACR,WmBz9hBG,MAAO,KnBy9hBO,QmBz9hBP,EnBy9hBiB,CmBz9hBjB,C;;QnB29hBd,OAAO,Q;O
;KAjBX,C;8FAoBA,yB;MAAA,8D;MmBt+hBA,iB;MnBs+hBA,sC;QAaiB,Q;QAFb,IAt/SO,qBAAQ,CAs/Sf,C;U
AAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SA
AS,UAAK,CAAL,CAAT,C;UACR,WmB7+hBG,MAAO,KnB6+hBO,QmB7+hBP,EnB6+hBiB,CmB7+hBjB,C;;Q
nB++hBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB1/hBA,iB;MnB0/hBA,sC;QAaiB,Q;QAFb,IAlgTO
,qBAAQ,CAkgTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV
,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBjgiBG,MAAO,KnBigiBO,QmBjgiBP,EnBigiBiB,C
mBjgiBjB,C;;QnBmgiBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,oC;MAAA,8D;MmB9giBA,iB;MnB8giBA,s
C;QAaiB,Q;QAFb,IA9gTO,qBAAQ,CA8gTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QA
CF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,WmBrhiBG,MAAO,KnBqhi
BO,QmBrhiBP,EnBqhiBiB,CmBrhiBjB,C;;QnBuhiBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8D;MmB7iiBA
,iB;MnB6iiBA,sC;QAaiB,Q;QAFb,IAlmTO,qBAAQ,CAkmTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CA
AL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBpjiBG,
MAAO,KnBojiBO,QmBpjiBP,EnBojiBiB,CmBpjiBjB,C;;QnBsjiBd,OAAO,Q;O;KAjBX,C;8FAoBA,yB;MAAA,8
D;MmBjkiBA,iB;MnBikiBA,sC;QAaiB,Q;QAFb,IA9mTO,qBAAQ,CA8mTf,C;UAAe,OAAO,I;QACtB,eAAe,SA
AS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UA
CR,WmBxkiBG,MAAO,KnBwkiBO,QmBxkiBP,EnBwkiBiB,CmBxkiBjB,C;;QnB0kiBd,OAAO,Q;O;KAjBX,C;+
FAoBA,yB;MAAA,8D;MmBrliBA,iB;MnBqliBA,sC;QAaiB,Q;QAFb,IA1nTO,qBAAQ,CA0nTf,C;UAAe,OAAO,I
;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CA
AL,CAAT,C;UACR,WmB5liBG,MAAO,KnB4liBO,QmB5liBP,EnB4liBiB,CmB5liBjB,C;;QnB8liBd,OAAO,Q;O;
KAjBX,C;+FAoBA,yB;MAAA,8D;MmBzmiBA,iB;MnBymiBA,sC;QAaiB,Q;QAFb,IAtoTO,qBAAQ,CAsoTf,C;U
AAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SA
AS,UAAK,CAAL,CAAT,C;UACR,WmBhniBG,MAAO,KnBgniBO,QmBhniBP,EnBgniBiB,CmBhniBjB,C;;QnBk
niBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmB7niBA,iB;MnB6niBA,sC;QAaiB,Q;QAFb,IAlpTO,qBA
AQ,CAkpTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;U
ACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBpoiBG,MAAO,KnBooiBO,QmBpoiBP,EnBooiBiB,CmB
poiBjB,C;;QnBsoiBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBjpiBA,iB;MnBipiBA,sC;QAaiB,Q;QA
Fb,IA9pTO,qBAAQ,CA8pTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,a
AAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBxpiBG,MAAO,KnBwpiBO,QmBxpiBP
,EnBwpiBiB,CmBxpiBjB,C;;QnB0piBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBrqiBA,iB;MnBqqiB
A,sC;QAaiB,Q;QAFb,IA1qTO,qBAAQ,CA0qTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;
QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmB5qiBG,MAAO,KnB
4qiBO,QmB5qiBP,EnB4qiBiB,CmB5qiBjB,C;;QnB8qiBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MmBzr
iBA,iB;MnByriBA,sC;QAaiB,Q;QAFb,IAtrTO,qBAAQ,CAsrTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,C
AAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,WmBhsiB
G,MAAO,KnBgsiBO,QmBhsiBP,EnBgsiBiB,CmBhsiBjB,C;;QnBksiBd,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAA
A,oC;MAAA,8D;MmB7siBA,iB;MnB6siBA,sC;QAaiB,Q;QAFb,IAlsTO,qBAAQ,CAksTf,C;UAAe,OAAO,I;QAC

tB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,
EAAT,C;UACR,WmBptiBG,MAAO,KnBotiBO,QmBptiBP,EnBotiBiB,CmBptiBjB,C;;QnBstiBd,OAAO,Q;O;KAj
BX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IApxTO,qBAAQ,CAoxTf,C;UAAe,OAAO,I;QACtB,
eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAA
T,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAA
A,8D;MAAA,sC;QAWiB,Q;QAFb,IAhyTO,qBAAQ,CAgyTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAA
L,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAA
W,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWi
B,Q;QAFb,IA5yTO,qBAAQ,CA4yTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;
QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YA
CI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAxzTO,qB
AAQ,CAwzTf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB
;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,
OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAp0TO,qBAAQ,CAo0Tf,C;UAAe,
OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,U
AAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+
FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAh1TO,qBAAQ,CAg1Tf,C;UAAe,OAAO,I;QACtB,eAAe,S
AAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;U
ACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;
MAAA,sC;QAWiB,Q;QAFb,IA51TO,qBAAQ,CA41Tf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CA
AT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CA
AX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,8D;MAAA,sC;QAWiB,Q;Q
AFb,IAx2TO,qBAAQ,CAw2Tf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,
aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WA
AW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;+FAoBA,yB;MAAA,oC;MAAA,8D;MAAA,sC;QAWiB,Q;QAFb,IAp3TO
,qBAAQ,CAo3Tf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAA
V,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QA
GnB,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA18TO,qBAAQ,C
A08Tf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UA
CI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,C
AAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAai
B,Q;QAFb,IAx9TO,qBAAQ,CAw9Tf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+
B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAA
kB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;
MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAt+TO,qBAAQ,CAs+Tf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAA
K,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,
UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnB
X,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAp/TO,qBAAQ,CAo/Tf,C;UAAe,MAAM,6
B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,C
AAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;
QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IAlgUO,qBAAQ,
CAkgUf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;U
ACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,
CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAa
iB,Q;QAFb,IAhhUO,qBAAQ,CAghUf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,
+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EA
AkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,s
E;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA9hUO,qBAAQ,CA8hUf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,U

AAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IA
AI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KA
nBX,C;0FAsBA,yB;MAAA,sE;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,IA5iUO,qBAAQ,CA4iUf,C;UAAe,MAA
M,6B;QACrB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAA
K,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,
C;;;QAGnB,OAAO,Q;O;KAnBX,C;0FAsBA,yB;MAAA,sE;MAAA,oC;MAAA,8D;MAAA,kD;QAaiB,Q;QAFb,I
A1jUO,qBAAQ,CA0jUf,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,EAAT,C;QACF,+B;QAAb,aA
AU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,
CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;oGAsBA,yB;MAAA,8D;MAAA,kD
;QAWiB,Q;QAFb,IA9oUO,qBAAQ,CA8oUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QA
CF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,
EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAA
A,8D;MAAA,kD;QAWiB,Q;QAFb,IA1pUO,qBAAQ,CA0pUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CA
AL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAA
W,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;s
GAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAtqUO,qBAAQ,CAsqUf,C;UAAe,OAAO,I;QACtB,eAAe,S
AAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;U
ACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,
Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IAlrUO,qBAAQ,CAkrUf,C;UAAe,OAAO,I;
QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CA
AL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;Q
AGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA9rUO,qBAAQ,CA8rUf,C;
UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,S
AAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YA
CI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA1sUO,qB
AAQ,CA0sUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;
UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC
,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;QAWiB,Q;QA
Fb,IAttUO,qBAAQ,CAstUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QACF,+B;QAAb,aA
AU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,
CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,8D;MAAA,kD;
QAWiB,Q;QAFb,IAluUO,qBAAQ,CAkuUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,UAAK,CAAL,CAAT,C;QAC
F,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,UAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,E
AAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;sGAoBA,yB;MAAA,
oC;MAAA,8D;MAAA,kD;QAWiB,Q;QAFb,IA9uUO,qBAAQ,CA8uUf,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sB
AAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,EAAT,C;UACR,IA
AI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAj
BX,C;IAoBA,8B;MASiB,Q;MAFb,IAh0UO,qBAAQ,CAg0Uf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;
MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBvwjBG,MAAO,KnBuwjBE,GmB
vwjBF,EnBuwjBO,CmBvwjBP,C;;MnBywjBd,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IAh1UO,qBAAQ,CAg1Uf
,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,C
AAL,C;QACR,MmBlyjBG,MAAO,KnBkyjBE,GmBlyjBF,EnBkyjBO,CmBlyjBP,C;;MnBoyjBd,OAAO,G;K;IAG
X,gC;MAOiB,Q;MAFb,IA91UO,qBAAQ,CA81Uf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,+B
;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;
;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAp2UO,qBAAQ,CAo2Uf,C;QAAe,OAAO,I;MACtB,UAAU,U
AAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C
;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IA12UO,qBAAQ,CA02Uf,C;QAAe,OAAO,I
;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI

,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAh3UO,qBAAQ,CAg3Uf,
C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CA
AL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;MAOiB,Q;MAFb,IAt3UO
,qBAAQ,CAs3Uf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,Q
AAQ,UAAK,CAAL,C;QACR,IAAI,oBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,gC;
MASiB,Q;MAFb,IA93UO,qBAAQ,CA83Uf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAA
b,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmBx3jBG,MAAO,KnBw3jBE,GmBx3jBF,EnBw3jB
O,CmBx3jBP,C;;MnB03jBd,OAAO,G;K;IAGX,gC;MASiB,Q;MAFb,IAt4UO,qBAAQ,CAs4Uf,C;QAAe,OAAO,I;
MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,MmB
73jBG,MAAO,KnB63jBE,GmB73jBF,EnB63jBO,CmB73jBP,C;;MnB+3jBd,OAAO,G;K;IAGX,gC;MAOiB,Q;M
AFb,IAp4UO,qBAAQ,CAo4Uf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CA
AV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IA
GX,wC;MAWiB,Q;MAFb,IAt9UO,qBAAQ,CAs9Uf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MAC
G,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,C
AAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAh+UO,qBAAQ,
CAg+Uf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,
UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;
MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IA1+UO,qBAAQ,CA0+Uf,C;QAAe,MAAM,6B;MACrB,UAA
U,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAA
Q,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;
MAFb,IAp/UO,qBAAQ,CAo/Uf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,
CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAj
C,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IA9/UO,qBAAQ,CA8/Uf,C;QAAe,MA
AM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QAC
R,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;I
AGX,0C;MAWiB,Q;MAFb,IAxgVO,qBAAQ,CAwgVf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;M
ACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAA
b,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IAlhVO,qBAA
Q,CAkhVf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAA
Q,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,
C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,Q;MAFb,IA5hVO,qBAAQ,CA4hVf,C;QAAe,MAAM,6B;MACrB,UA
AU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SA
AQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0C;MAWiB,
Q;MAFb,IAtiVO,qBAAQ,CAsiVf,C;QAAe,MAAM,6B;MACrB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU
,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CA
AjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,8C;MAOiB,Q;MAFb,IApnVO,qBAAQ,CAonVf,C;QAAe,
OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,+B;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QA
CR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K
;IAGX,gD;MAOiB,Q;MAFb,IA1nVO,qBAAQ,CA0nVf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MAC
G,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,C
AAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAhoVO,qBAAQ,
CAgoVf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,U
AAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;
MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAtoVO,qBAAQ,CAsoVf,C;QAAe,OAAO,I;MACtB,UAAU,UA
AK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GA
AR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb
,IA5oVO,qBAAQ,CA4oVf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,i
B;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UA

AoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IAlpVO,qBAAQ,CAkpVf,C;QAAe,OAAO,I;M
ACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,U
AAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;
MAOiB,Q;MAFb,IAxpVO,qBAAQ,CAwpVf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAA
b,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA
6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAOiB,Q;MAFb,IA9pVO,qBAAQ,CA8pVf,C;
QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL
,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAA
O,G;K;IAGX,gD;MAOiB,Q;MAFb,IApqVO,qBAAQ,CAoqVf,C;QAAe,OAAO,I;MACtB,UAAU,UAAK,CAAL,C
;MACG,iC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,UAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,c
AAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,yB;MAMI,OAnvVO,qBAAQ,C;
K;IAsvVnB,2B;MAMI,OApvVO,qBAAQ,C;K;IAuvVnB,2B;MAMI,OArvVO,qBAAQ,C;K;IAwvVnB,2B;MAMI,
OAtvVO,qBAAQ,C;K;IAyvVnB,2B;MAMI,OAvvVO,qBAAQ,C;K;IA0vVnB,2B;MAMI,OAxvVO,qBAAQ,C;K;I
A2vVnB,2B;MAMI,OAzvVO,qBAAQ,C;K;IA4vVnB,2B;MAMI,OA1vVO,qBAAQ,C;K;IA6vVnB,2B;MAMI,OA
3vVO,qBAAQ,C;K;gFA8vVnB,gC;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB
,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;gFAGX,gC;MAMoB,Q;MAAhB,wBAAgB
,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO
,I;K;iFAGX,gC;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAA
V,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAg
B,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;M
AMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAw
B,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,
M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MAMoB,Q;MAAhB
,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MAC
rD,OAAO,I;K;iFAGX,gC;MAMoB,Q;MAAhB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,IAAI,U
AAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,yB;MAAA,oC;MAAA,gC;MAAA,uC;QA
MoB,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UAAsB,IAAI,UAAU,oBAAV,CAAJ,
C;YAAwB,OAAO,K;;QACrD,OAAO,I;O;KAPX,C;kFAUA,6B;MAMmC,Q;MAAhB,iD;QAAgB,cAAhB,e;QAAs
B,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAgB,cAAhB,e;QAAsB,OAAO,OAAP,
C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAgB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;
oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAgB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MAM
mC,Q;MAAhB,iD;QAAgB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,iD
;QAAgB,cAAhB,e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAgB,cAAhB,
e;QAAsB,OAAO,OAAP,C;;MAArC,gB;K;oFAGJ,6B;MAMmC,Q;MAAhB,iD;QAAgB,cAAhB,e;QAAsB,OAAO,
OAAP,C;;MAArC,gB;K;oFAGJ,yB;MAAA,oC;MAAA,gC;MAAA,oC;QAMmC,Q;QAAhB,iD;UAAgB,cAAhB,0
B;UAAsB,OAAO,oBAAP,C;;QAArC,gB;O;KANJ,C;gGASA,6B;MAn4KiB,gB;MADb,YAAY,C;MACZ,iD;QAA
a,WAAb,e;QAAmB,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MA44KnB,gB;K;kGAGJ,6B;MAr4KiB,gB;
MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAmB,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MA84Kn
B,gB;K;kGAGJ,6B;MAv4KiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAmB,QAAO,cAAP,EAAO,sB
AAP,WAAgB,IAAhB,C;;MAg5KnB,gB;K;kGAGJ,6B;MAz4KiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;
QAAmB,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAk5KnB,gB;K;kGAGJ,6B;MA34KiB,gB;MADb,YA
AY,C;MACZ,iD;QAAa,WAAb,e;QAAmB,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAo5KnB,gB;K;kG
AGJ,6B;MA74KiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAmB,QAAO,cAAP,EAAO,sBAAP,WAAg
B,IAAhB,C;;MAs5KnB,gB;K;kGAGJ,6B;MA/4KiB,gB;MADb,YAAY,C;MACZ,iD;QAAa,WAAb,e;QAAmB,QA
AO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAw5KnB,gB;K;kGAGJ,6B;MAj5KiB,gB;MADb,YAAY,C;MAC
Z,iD;QAAa,WAAb,e;QAAmB,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MA05KnB,gB;K;kGAGJ,yB;MA
AA,6B;MAAA,sC;MA15KA,oC;MAAA,gC;MA05KA,2BASiB,yB;QAn6KjB,oC;QAAA,gC;eAm6KiB,0B;UAAA
,4B;YAAE,aAAe,c;YA55KjB,gB;YADb,YAAY,C;YACZ,iD;cAAa,WAAb,0B;cAAmB,QAAO,cAAP,EAAO,sBA

AP,WAAgB,iBAAhB,C;;YA45KmB,W;W;S;OAAzB,C;MATjB,oC;QAn5KiB,gB;QADb,YAAY,C;QACZ,iD;UA
Aa,WAAb,0B;UAAmB,QAAO,cAAP,EAAO,sBAAP,WAAgB,iBAAhB,C;;QA45KnB,gB;O;KATJ,C;kFAYA,yB;
MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAlmWO,qBAAQ,CAkmWf,C;UACI,MAAM,mCAA8B,+
BAA9B,C;QACV,kBAAqB,UAAK,CAAL,C;QACJ,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EA
AuB,UAAK,KAAL,CAAvB,C;;QAElB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;Q
AgBqB,Q;QAHjB,IAhnWO,qBAAQ,CAgnWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,C
AAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QA
ElB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA9nWO,qBAA
Q,CA8nWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,
CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAElB,OAAO,W;O;KAnBX,C;oFAsB
A,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA5oWO,qBAAQ,CA4oWf,C;UACI,MAAM,mCAA
8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAA
V,EAAuB,UAAK,KAAL,CAAvB,C;;QAElB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,
uC;QAgBqB,Q;QAHjB,IA1pWO,qBAAQ,CA0pWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UA
AK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C
;;QAElB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAxqWO,qB
AAQ,CAwqWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBA
Ac,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAElB,OAAO,W;O;KAnBX,C;oF
AsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAtrWO,qBAAQ,CAsrWf,C;UACI,MAAM,mC
AA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,W
AAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAElB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MA
AA,uC;QAgBqB,Q;QAHjB,IApsWO,qBAAQ,CAosWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,
UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAv
B,C;;QAElB,OAAO,W;O;KAnBX,C;oFAsBA,yB;MAAA,4F;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QAg
BqB,Q;QAHjB,IAltWO,qBAAQ,CAktWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL
,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,wBAAV,EAAuB,sBAAK,KAAL,EAAvB,E;;QAEl
B,OAAO,W;O;KAnBX,C;gGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAxyWO,qBAAQ,
CAwyWf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAqB,UAAK,CAAL,C;QACJ,+B;QAAjB,iBAAc,C
AAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KA
nBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAtzWO,qBAAQ,CAszWf,C;UACI,M
AAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,
UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KAnBX,C;kGAsBA,yB;
MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAp0WO,qBAAQ,CAo0Wf,C;UACI,MAAM,mCAA8B,+B
AA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EA
AiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,
8D;MAAA,uC;QAgBqB,Q;QAHjB,IAl1WO,qBAAQ,CAk1Wf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kB
AAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8
B,UAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAg
BqB,Q;QAHjB,IAh2WO,qBAAQ,CAg2Wf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAA
L,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CA
A9B,C;;QAElB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA92
WO,qBAAQ,CA82Wf,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAA
jB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAElB,OAA
O,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA53WO,qBAAQ,CA43W
f,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;
UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KAnBX,C;k
GAsBA,yB;MAAA,4F;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA14WO,qBAAQ,CA04Wf,C;UACI,MAAM,
mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAA

U,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAA
A,4F;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,IAx5WO,qBAAQ,CAw5Wf,C;UACI,MA
AM,mCAA8B,+BAA9B,C;QACV,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,o
BAAU,KAAV,EAAiB,wBAAjB,EAA8B,sBAAK,KAAL,EAA9B,E;;QAElB,OAAO,W;O;KAnBX,C;4GAsBA,yB;
MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA9+WO,qBAAQ,CA8+Wf,C;UACI,OAAO,I;QACX,kBAAqB,UAA
K,CAAL,C;QACJ,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KA
AL,CAA9B,C;;QAElB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA5/WO,
qBAAQ,CA4/Wf,C;UACI,OAAO,I;QACX,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI
,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KAnBX,C;8GAsB
A,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IA1gXO,qBAAQ,CA0gXf,C;UACI,OAAO,I;QACX,kBAAkB,U
AAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,
KAAL,CAA9B,C;;QAElB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAxh
XO,qBAAQ,CAwhXf,C;UACI,OAAO,I;QACX,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;
UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KAnBX,C;8
GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAtiXO,qBAAQ,CAsiXf,C;UACI,OAAO,I;QACX,kBAAk
B,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UA
AK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,I
ApjXO,qBAAQ,CAojXf,C;UACI,OAAO,I;QACX,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,y
B;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KAnBX,C
;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QAHjB,IAlkXO,qBAAQ,CAkkXf,C;UACI,OAAO,I;QACX,kB
AAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8
B,UAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KAnBX,C;8GAsBA,yB;MAAA,8D;MAAA,uC;QAgBqB,Q;QA
HjB,IAhlXO,qBAAQ,CAglXf,C;UACI,OAAO,I;QACX,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CA
Ad,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KAn
BX,C;8GAsBA,yB;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,IA9lXO,qBAAQ,CA8lXf,C
;UACI,OAAO,I;QACX,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,KA
AV,EAAiB,wBAAjB,EAA8B,sBAAK,KAAL,EAA9B,E;;QAElB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;
MAAA,uC;QAiBqB,Q;QAHjB,IArrXO,qBAAQ,CAqrXf,C;UACI,OAAO,I;QACX,kBAAqB,UAAK,CAAL,C;QA
CJ,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAElB,OAAO,
W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAiBqB,Q;QAHjB,IApsXO,qBAAQ,CAosXf,C;UACI,OAA
O,I;QACX,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,
UAAK,KAAL,CAAvB,C;;QAElB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAiBqB,Q;QAHj
B,IAntXO,qBAAQ,CAmtXf,C;UACI,OAAO,I;QACX,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAA
d,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAElB,OAAO,W;O;KApBX,C;gGAuBA,yB
;MAAA,8D;MAAA,uC;QAiBqB,Q;QAHjB,IAluXO,qBAAQ,CAkuXf,C;UACI,OAAO,I;QACX,kBAAkB,UAAK,
CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;Q
AElB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAiBqB,Q;QAHjB,IAjvXO,qBAAQ,CAivXf,
C;UACI,OAAO,I;QACX,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,W
AAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAElB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAi
BqB,Q;QAHjB,IAhwXO,qBAAQ,CAgwXf,C;UACI,OAAO,I;QACX,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAj
B,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAElB,OAAO,W;O;KApBX,
C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAiBqB,Q;QAHjB,IA/wXO,qBAAQ,CA+wXf,C;UACI,OAAO,I;QACX,k
BAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,UAAK,KAA
L,CAAvB,C;;QAElB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D;MAAA,uC;QAiBqB,Q;QAHjB,IA9xXO,q
BAAQ,CA8xXf,C;UACI,OAAO,I;QACX,kBAAkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,
cAAc,UAAU,WAAV,EAAuB,UAAK,KAAL,CAAvB,C;;QAElB,OAAO,W;O;KApBX,C;gGAuBA,yB;MAAA,8D
;MAAA,oC;MAAA,gC;MAAA,uC;QAiBqB,Q;QAHjB,IA7yXO,qBAAQ,CA6yXf,C;UACI,OAAO,I;QACX,kBA
AkB,UAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,wBAAV,EAAuB,sBAAK,KAA

L,EAAvB,E;;QAElB,OAAO,W;O;KApBX,C;4FAuBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAe6B,UAEO,M;Q
AJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAqB,UAAI,Y
AAJ,EAAI,oBAAJ,O;QACrB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EA
AwB,WAAxB,C;;QAElB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAe0B,UAEU,
M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,UA
AI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAA
V,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAe0B,U
AEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB
,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,S
AAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAe0B
,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAA
kB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ
,SAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAe
0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kB
AAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sB
AAJ,SAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;
QAe0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB
,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,
sBAAJ,SAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,4F;MAAA,u
C;QAe0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QAC
rB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EA
AI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,4F;MAA
A,uC;QAe0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;Q
ACrB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,
EAAI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAnBX,C;8FAsBA,yB;MAAA,8D;MAAA,4F;M
AAA,oC;MAAA,gC;MAAA,uC;QAe0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAA
M,mCAA8B,+BAA9B,C;QACrB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,
cAAc,oBAAU,sBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,wBAAxB,E;;QAElB,OAAO,W;O;KAnBX,C;0GAsBA
,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAe6B,Q;QAFzB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAA
M,mCAA8B,+BAA9B,C;QACrB,kBAAqB,UAAI,YAAJ,EAAI,oBAAJ,O;QACrB,OAAO,SAAS,CAAhB,C;UACI,
cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;
4GAuBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAe0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAA
e,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C
;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAp
BX,C;4GAuBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAe0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C
;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CA
AhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;
O;KApBX,C;4GAuBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAe0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,C
AAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SA
AS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OA
AO,W;O;KApBX,C;4GAuBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAe0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,Q
AAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OA
AO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;Q
AEJ,OAAO,W;O;KApBX,C;4GAuBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAe0B,Q;QAFtB,YAAY,wB;QACZ
,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QA
ClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd
,qB;;QAEJ,OAAO,W;O;KApBX,C;4GAuBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAe0B,Q;QAFtB,YAAY,wB;
QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ

,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C
;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;4GAuBA,yB;MAAA,8D;MAAA,4F;MAAA,uC;QAe0B,Q;QAFtB,YA
AY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,UAAI,YAAJ,EAAI
,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,W
AA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;4GAuBA,yB;MAAA,8D;MAAA,4F;MAAA,oC;MAAA,gC;M
AAA,uC;QAe0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACr
B,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oBAAU,KAAV,EAAiB,s
BAAI,KAAJ,EAAjB,EAA6B,wBAA7B,E;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;sHAuBA,yB;MAAA,8D;MA
AA,uC;QAe6B,Q;QAFzB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAqB,UAAI,YA
AJ,EAAI,oBAAJ,O;QACrB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,E
AA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;wHAuBA,yB;MAAA,8D;MAAA,uC;QAe0B,Q;QAFt
B,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAC
lB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,
qB;;QAEJ,OAAO,W;O;KApBX,C;wHAuBA,yB;MAAA,8D;MAAA,uC;QAe0B,Q;QAFtB,YAAY,wB;QACZ,IAA
I,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB
,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;K
ApBX,C;wHAuBA,yB;MAAA,8D;MAAA,uC;QAe0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,
OAAO,I;QACtB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,K
AAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;wHAuBA,yB;M
AAA,8D;MAAA,uC;QAe0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAk
B,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAA
J,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;wHAuBA,yB;MAAA,8D;MAAA,uC;QAe
0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAkB,UAAI,YAAJ,EAAI,oB
AAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA
7B,C;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;wHAuBA,yB;MAAA,8D;MAAA,uC;QAe0B,Q;QAFtB,YAAY,w
B;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,S
AAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,O
AAO,W;O;KApBX,C;wHAuBA,yB;MAAA,8D;MAAA,uC;QAe0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CA
AZ,C;UAAe,OAAO,I;QACtB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cA
Ac,UAAU,KAAV,EAAiB,UAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;wH
AuBA,yB;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QAe0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAA
Z,C;UAAe,OAAO,I;QACtB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAA
c,oBAAU,KAAV,EAAiB,sBAAI,KAAJ,EAAjB,EAA6B,wBAA7B,E;UACd,qB;;QAEJ,OAAO,W;O;KApBX,C;w
GAuBA,yB;MAAA,8D;MAAA,uC;QAgB6B,UAEO,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,
OAAO,I;QACtB,kBAAqB,UAAI,YAAJ,EAAI,oBAAJ,O;QACrB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,U
AAI,cAAJ,EAAI,sBAAJ,SAAV,EAAAwB,WAAxB,C;;QAElB,OAAO,W;O;KApBX,C;0GAuBA,yB;MAAA,8D;M
AAA,uC;QAgB0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAkB,
UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,S
AAV,EAAAwB,WAAxB,C;;QAElB,OAAO,W;O;KApBX,C;0GAuBA,yB;MAAA,8D;MAAA,uC;QAgB0B,UAEU,
M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ
,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EAAAwB,WAAxB,C;;
QAElB,OAAO,W;O;KApBX,C;0GAuBA,yB;MAAA,8D;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAY,wB;QAC
Z,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,C
AAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EAAAwB,WAAxB,C;;QAElB,OAAO,W;O;KApB
X,C;0GAuBA,yB;MAAA,8D;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;U
AAe,OAAO,I;QACtB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UA
AU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EAAAwB,WAAxB,C;;QAElB,OAAO,W;O;KApBX,C;0GAuBA,yB;MAAA,
8D;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBA

AkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBA
AJ,SAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KApBX,C;0GAuBA,yB;MAAA,8D;MAAA,uC;QAgB0B,UA
EU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAkB,UAAI,YAAJ,EAAI,oB
AAJ,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EAAwB,WAAxB,
C;;QAElB,OAAO,W;O;KApBX,C;0GAuBA,yB;MAAA,8D;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAY,wB;Q
ACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAA
S,CAAhB,C;UACI,cAAc,UAAU,UAAI,cAAJ,EAAI,sBAAJ,SAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KA
pBX,C;0GAuBA,yB;MAAA,8D;MAAA,oC;MAAA,gC;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAY,wB;QACZ
,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAkB,UAAI,YAAJ,EAAI,oBAAJ,O;QAClB,OAAO,SAAS,CA
AhB,C;UACI,cAAc,oBAAU,sBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,wBAAxB,E;;QAElB,OAAO,W;O;KAp
BX,C;4FAuBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAgBoB,Q;QAHhB,IAtrZO,qBAAQ,CAsrZf,C;UAAe,OA
AO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB10oBO,W;QlB20o
BP,kBAAkB,O;QAClB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OA
AvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KApBX,C;8FAuBA,yB;MAAA,gD;MAAA,gE;MAA
A,gD;QAiBoB,Q;QAHhB,IAtsZO,qBAAQ,CAssZf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBA
AO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBl2oBO,W;QlBm2oBP,kBAAkB,O;QAClB,wBAAgB,SAAhB,gB;UA
AgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,O
AAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAiBoB,Q;QAHhB,IAttZO,qBAAQ,CAstZ
f,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB13oB
O,W;QlB23oBP,kBAAkB,O;QAClB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAA
V,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;M
AAA,gE;MAAA,gD;QAiBoB,Q;QAHhB,IAtuZO,qBAAQ,CAsuZf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBA
AvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBl5oBO,W;QlBm5oBP,kBAAkB,O;QAClB,wBAAgB
,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WA
AJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAiBoB,Q;QAHhB,IAtvZO
,qBAAQ,CAsvZf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;Q
AA5C,akB16oBO,W;QlB26oBP,kBAAkB,O;QAClB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAA
c,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB
;MAAA,gD;MAAA,gE;MAAA,gD;QAiBoB,Q;QAHhB,IAtwZO,qBAAQ,CAswZf,C;UAAe,OAAO,OAAO,OAAP
,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBl8oBO,W;QlBm8oBP,kBAAkB,O;QA
ClB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MA
AO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAiBoB,Q;
QAHhB,IAtxZO,qBAAQ,CAsxZf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,
C;QAA+B,8B;QAA5C,akB19oBO,W;QlB29oBP,kBAAkB,O;QAClB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAh
B,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX
,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAiBoB,Q;QAHhB,IAtyZO,qBAAQ,CAsyZf,C;UAAe,OAAO
,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBl/oBO,W;QlBm/oBP,k
BAAkB,O;QAClB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAc,UAAU,WAAV,EAAuB,OAAvB
,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;8FAwBA,yB;MAAA,gD;MAAA,gE;MAAA,o
C;MAAA,gC;MAAA,gD;QAiBoB,Q;QAHhB,IAtzZO,qBAAQ,CAszZf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,k
BAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB1gpBO,W;QlB2gpBP,kBAAkB,O;QAClB,wBA
AgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,cAAc,UAAU,WAAV,EAAuB,oBAAvB,C;UACd,MA
AO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;0GAwBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAAcI,IA94Z
O,qBAAQ,CA84Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B
;QAA5C,akBlipBO,W;QlBmipBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,U
AAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;4GAwBA,yB;MAAA,gD;
MAAA,gE;MAAA,gD;QAeI,IA/5ZO,qBAAQ,CA+5Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,
mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB3jpBO,W;QlB4jpBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAA

U,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAh7ZO,qBAAQ,CAg7Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBplpBO,W;QlBqlpBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAj8ZO,qBAAQ,CAi8Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB7mpBO,W;QlB8mpBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAl9ZO,qBAAQ,CAk9Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBtopBO,W;QlBuopBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAn+ZO,qBAAQ,CAm+Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB/qpBO,W;QlBgrpBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAp/ZO,qBAAQ,CAo/Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBxtpBO,W;QlBytpBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAr/ZO,qBAAQ,CAq/Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBjvpBO,W;QlBkvpBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAt/ZO,qBAAQ,CAs/Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBxwpBO,W;QlBywpBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAv/ZO,qBAAQ,CAu/Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB5xpBO,W;QlB6xpBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAx/ZO,qBAAQ,CAw/Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBhzpBO,W;QlBizpBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAz/ZO,qBAAQ,CAy/Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBp0pBO,W;QlBq0pBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IA1/ZO,qBAAQ,CA0/Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBx1pBO,W;QlBy1pBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IA3/ZO,qBAAQ,CA2/Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB52pBO,W;QlB62pBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IA5/ZO,qBAAQ,CA4/Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBh4pBO,W;QlBi4pBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IA7/ZO,qBAAQ,CA6/Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBp5pBO,W;QlBq5pBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IA9/ZO,qBAAQ,CA8/Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBx6pBO,W;QlBy6pBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IA//ZO,qBAAQ,CA+/Zf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB57pBO,W;QlB67pBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAhgaO,qBAAQ,CAggaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBh9pBO,W;QlBi9pBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAjgaO,qBAAQ,CAigaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBp+pBO,W;QlBq+pBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAlgaO,qBAAQ,CAkgaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBx/pBO,W;QlBy/pBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAngaO,qBAAQ,CAmgaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB5gqBO,W;QlB6gqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IApgaO,qBAAQ,CAogaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBhhqBO,W;QlBihqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IArgaO,qBAAQ,CAqgaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBphqBO,W;QlBqhqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAtgaO,qBAAQ,CAsgaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBxiqBO,W;QlByiqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAvgaO,qBAAQ,CAugaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB5jqBO,W;QlB6jqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAxgaO,qBAAQ,CAwgaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBhlqBO,W;QlBilqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAzgaO,qBAAQ,CAygaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBplqBO,W;QlBqlqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IA1gaO,qBAAQ,CA0gaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBxmqBO,W;QlBymqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IA3gaO,qBAAQ,CA2gaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB5nqBO,W;QlB6nqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IA5gaO,qBAAQ,CA4gaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBhpqBO,W;QlBipqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IA7gaO,qBAAQ,CA6gaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBpqqBO,W;QlBqqqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IA9gaO,qBAAQ,CA8gaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBxrqBO,W;QlByrqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IA/gaO,qBAAQ,CA+gaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB5sqBO,W;QlB6sqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAhhaO,qBAAQ,CAghaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBhuqBO,W;QlBiuqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAjhaO,qBAAQ,CAihaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBpvqBO,W;QlBqvqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAlhaO,qBAAQ,CAkhaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akBxwqBO,W;QlBywqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IAnhaO,qBAAQ,CAmhaf,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,akB5xqBO,W;QlB6xqBP,kBAAkB,O;QAClB,wD;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;4GAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAeI,IApha

AtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OA
AO,M;O;KAlBX,C;kGAqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAWI,IAzsaO,qBAAQ,CAysaf,C;UAAe,OA
AO,W;QACtB,sBAAkB,UAAK,CAAL,CAAlB,C;QACsC,kBAAzB,eAAmB,gBAAnB,C;QAAiC,sBAAI,aAAJ,C;
QAA9C,akBt5pBO,W;QlBu5pBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,aAAV,EAAuB,UAAK
,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAlBX,C;kGAqBA,yB;MAAA,qD;MAAA
,gE;MAAA,oC;MAAA,gC;MAAA,uC;QAWI,IAttaO,qBAAQ,CAstaf,C;UAAe,OAAO,W;QACtB,sBAAkB,UAA
K,CAAL,CAAlB,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,0BAAJ,C;QAA3C,akB36pBO,W;QlB4
6pBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,oBAAU,0BAAV,EAAuB,sBAAK,KAAL,EAAvB,E;UAC
d,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KAlBX,C;8GAqBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAcI,I
A9yaO,qBAAQ,CA8yaf,C;UAAe,OAAO,W;QACtB,sBAAqB,UAAK,CAAL,CAArB,C;QACgC,kBAAnB,eAAa,g
BAAb,C;QAA2B,sBAAI,aAAJ,C;QAAxC,akBn8pBO,W;QlBo8pBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gB
AAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OA
AO,M;O;KArBX,C;gHAwBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IA5zaO,qBAAQ,CA4zaf,C;UAAe,OA
AO,W;QACtB,sBAAkB,UAAK,CAAL,CAAlB,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,aAAJ,C;
QAA3C,akBz9pBO,W;QlB09pBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB
,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;M
AAA,qD;MAAA,gE;MAAA,uC;QAYI,IA10aO,qBAAQ,CA00af,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CA
AL,CAAlB,C;QACoC,kBAAvB,eAAiB,gBAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,akB/+pBO,W;QlBg/pBP,iB
AAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;U
ACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QA
YI,IAx1aO,qBAAQ,CAw1af,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAlB,C;QACkC,kBAArB,eA
Ae,gBAAf,C;QAA6B,sBAAI,aAAJ,C;QAA1C,akBrgqBO,W;QlBsgqBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,
gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,O
AAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IAt2aO,qBAAQ,CAs2af,C;UAAe,O
AAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAlB,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,aAAJ,
C;QAA3C,akB3hqBO,W;QlB4hqBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAA
jB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;
MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IAp3aO,qBAAQ,CAo3af,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,C
AAL,CAAlB,C;QACoC,kBAAvB,eAAiB,gBAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,akBjjqBO,W;QlBkjqBP,iB
AAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;U
ACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QA
YI,IAl4aO,qBAAQ,CAk4af,C;UAAe,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAlB,C;QACqC,kBAAxB,eAA
kB,gBAAlB,C;QAAgC,sBAAI,aAAJ,C;QAA7C,akBvkqBO,W;QlBwkqBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UA
CI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAE
X,OAAO,M;O;KAnBX,C;gHAsBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAYI,IAh5aO,qBAAQ,CAg5af,C;UA
Ae,OAAO,W;QACtB,sBAAkB,UAAK,CAAL,CAAlB,C;QACsC,kBAAzB,eAAmB,gBAAnB,C;QAAiC,sBAAI,a
AAJ,C;QAA9C,akB7lqBO,W;QlB8lqBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,UAAU,KAAV,EAAiB,
aAAjB,EAA8B,UAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;gHAsBA
,yB;MAAA,qD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,uC;QAYI,IA95aO,qBAAQ,CA85af,C;UAAe,OAAO,W
;QACtB,sBAAkB,UAAK,CAAL,CAAlB,C;QACmC,kBAAtB,eAAgB,gBAAhB,C;QAA8B,sBAAI,0BAAJ,C;QAA
3C,akBnnqBO,W;QlBonqBP,iBAAc,CAAd,UAAsB,gBAAtB,U;UACI,gBAAc,oBAAU,KAAV,EAAiB,0BAAjB,E
AA8B,sBAAK,KAAL,EAA9B,E;UACd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KAnBX,C;8EAsBA,yB;M
A/zBA,gD;MAAA,gE;MA+zBA,gD;QAcW,sB;;UA7zBS,Q;UAHhB,IAtrZO,qBAAQ,CAsrZf,C;YAAe,qBAAO,O
Ag0BH,OAh0BG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA+zBzB,OA/zByB,
C;UAA5C,akB10oBO,W;UlB20oBP,kBA8zBmB,O;UA7zBnB,iD;YAAgB,cAAhB,e;YACI,cA4zBwB,SA5zBV,C
AAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAyzBP,yB;O;KAdJ,C;gF
AiBA,yB;MAzzBA,gD;MAAA,gE;MAyzBA,gD;QAeW,sB;;UAvzBS,Q;UAHhB,IAtsZO,qBAAQ,CAssZf,C;YAA
e,qBAAO,OA0zBH,OA1zBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAyzBzB,

OAzzByB,C;UAA5C,akBl2oBO,W;UlBm2oBP,kBAwzBmB,O;UAvzBnB,iD;YAAgB,cAAhB,e;YACI,cAszBwB,
SAtzBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAmzBP,yB;O;
KAfJ,C;gFAkBA,yB;MAnzBA,gD;MAAA,gE;MAmzBA,gD;QAeW,sB;;UAjzBS,Q;UAHhB,IAttZO,qBAAQ,CAs
tZf,C;YAAe,qBAAO,OAozBH,OApzBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,
sBAmzBzB,OAnzByB,C;UAA5C,akB13oBO,W;UlB23oBP,kBAkzBmB,O;UAjzBnB,iD;YAAgB,cAAhB,e;YACI,
cAgzBwB,SAhzBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA6
yBP,yB;O;KAfJ,C;gFAkBA,yB;MA7yBA,gD;MAAA,gE;MA6yBA,gD;QAeW,sB;;UA3yBS,Q;UAHhB,IAtuZO,q
BAAQ,CAsuZf,C;YAAe,qBAAO,OA8yBH,OA9yBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAA
b,C;UAA+B,sBA6yBzB,OA7yByB,C;UAA5C,akBl5oBO,W;UlBm5oBP,kBA4yBmB,O;UA3yBnB,iD;YAAgB,cA
AhB,e;YACI,cA0yBwB,SA1yBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBA
AO,M;;;QAuyBP,yB;O;KAfJ,C;gFAkBA,yB;MAvyBA,gD;MAAA,gE;MAuyBA,gD;QAeW,sB;;UAryBS,Q;UAH
hB,IAtvZO,qBAAQ,CAsvZf,C;YAAe,qBAAO,OAwyBH,OAxyBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAA
O,CAAP,IAAb,C;UAA+B,sBAuyBzB,OAvyByB,C;UAA5C,akB16oBO,W;UlB26oBP,kBAsyBmB,O;UAryBnB,i
D;YAAgB,cAAhB,e;YACI,cAoyBwB,SApyBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,
C;;UAEX,qBAAO,M;;;QAiyBP,yB;O;KAfJ,C;gFAkBA,yB;MAjyBA,gD;MAAA,gE;MAiyBA,gD;QAeW,sB;;UA/
xBS,Q;UAHhB,IAtwZO,qBAAQ,CAswZf,C;YAAe,qBAAO,OAkyBH,OAlyBG,C;YAAP,uB;;UACqB,kBAAvB,e
AAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAiyBzB,OAjyByB,C;UAA5C,akBl8oBO,W;UlBm8oBP,kBAgyBmB,O;
UA/xBnB,iD;YAAgB,cAAhB,e;YACI,cA8xBwB,SA9xBV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WA
AI,WAAJ,C;;UAEX,qBAAO,M;;;QA2xBP,yB;O;KAfJ,C;gFAkBA,yB;MA3xBA,gD;MAAA,gE;MA2xBA,gD;QA
eW,sB;;UAzxBS,Q;UAHhB,IAtxZO,qBAAQ,CAsxZf,C;YAAe,qBAAO,OA4xBH,OA5xBG,C;YAAP,uB;;UACqB
,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA2xBzB,OA3xByB,C;UAA5C,akB19oBO,W;UlB29oBP,kB
A0xBmB,O;UAzxBnB,iD;YAAgB,cAAhB,e;YACI,cAwxBwB,SAxxBV,CAAU,WAAV,EAAuB,OAAvB,C;YAC
d,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAqxBP,yB;O;KAfJ,C;gFAkBA,yB;MArxBA,gD;MAAA,gE;MA
qxBA,gD;QAeW,sB;;UAnxBS,Q;UAHhB,IAtyZO,qBAAQ,CAsyZf,C;YAAe,qBAAO,OAsxBH,OAtxBG,C;YAAP
,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAqxBzB,OArxByB,C;UAA5C,akBl/oBO,W;Ul
Bm/oBP,kBAoxBmB,O;UAnxBnB,iD;YAAgB,cAAhB,e;YACI,cAkxBwB,SAlxBV,CAAU,WAAV,EAAuB,OAA
vB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA+wBP,yB;O;KAfJ,C;gFAkBA,yB;MA/wBA,gD;M
AAA,gE;MAAA,oC;MAAA,gC;MA+wBA,gD;QAeW,sB;;UA7wBS,Q;UAHhB,IAtzZO,qBAAQ,CAszZf,C;YAAe
,qBAAO,OAgxBH,OAhxBG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA+wBzB,
OA/wByB,C;UAA5C,akB1gpBO,W;UlB2gpBP,kBA8wBmB,O;UA7wBnB,iD;YAAgB,cAAhB,0B;YACI,cA4wB
wB,SA5wBV,CAAU,WAAV,EAAuB,oBAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAywBP,
yB;O;KAfJ,C;4FAkBA,yB;MAzwBA,gD;MAAA,gE;MAywBA,gD;QAeW,6B;;UA1wBP,IA94ZO,qBAAQ,CA84
Zf,C;YAAe,4BAAO,OA0wBI,OA1wBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,s
BAywBlB,OAzwBkB,C;UAA5C,akBlipBO,W;UlBmipBP,kBAwwB0B,O;UAvwB1B,wD;YACI,cAswB+B,SAtw
BjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4B
AAO,M;;;QAmwBP,gC;O;KAfJ,C;8FAkBA,yB;MAnwBA,gD;MAAA,gE;MAmwBA,gD;QAgBW,6B;;UApwBP,I
A/5ZO,qBAAQ,CA+5Zf,C;YAAe,4BAAO,OAowBI,OApwBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CA
AP,IAAb,C;UAA+B,sBAmwBlB,OAnwBkB,C;UAA5C,akB3jpBO,W;UlB4jpBP,kBAkwB0B,O;UAjwB1B,wD;Y
ACI,cAgwB+B,SAhwBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAA
I,WAAJ,C;;UAEX,4BAAO,M;;;QA6vBP,gC;O;KAhBJ,C;8FAmBA,yB;MA7vBA,gD;MAAA,gE;MA6vBA,gD;Q
AgBW,6B;;UA9vBP,IAh7ZO,qBAAQ,CAg7Zf,C;YAAe,4BAAO,OA8vBI,OA9vBJ,C;YAAP,8B;;UACqB,kBAAv
B,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA6vBlB,OA7vBkB,C;UAA5C,akBplpBO,W;UlBqlpBP,kBA4vB0B,
O;UA3vB1B,wD;YACI,cA0vB+B,SA1vBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;Y
ACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAuvBP,gC;O;KAhBJ,C;8FAmBA,yB;MAvvBA,gD;MAAA,g
E;MAuvBA,gD;QAgBW,6B;;UAxvBP,IAj8ZO,qBAAQ,CAi8Zf,C;YAAe,4BAAO,OAwvBI,OAxvBJ,C;YAAP,8B
;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAuvBlB,OAvvBkB,C;UAA5C,akB7mpBO,W;UlB8
mpBP,kBAsvB0B,O;UArvB1B,wD;YACI,cAovB+B,SApvBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,K
AAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAivBP,gC;O;KAhBJ,C;8FAmBA,yB;MAj

vBA,gD;MAAA,gE;MAivBA,gD;QAgBW,6B;;UAlvBP,IAl9ZO,qBAAQ,CAk9Zf,C;YAAe,4BAAO,OAkvBI,OAl
vBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBAivBlB,OAjvBkB,C;UAA5C,akBt
opBO,W;UlBuopBP,kBAgvB0B,O;UA/uB1B,wD;YACI,cA8uB+B,SA9uBjB,CAAU,KAAV,EAAiB,WAAjB,EAA
8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QA2uBP,gC;O;KAhBJ,C;8FA
mBA,yB;MA3uBA,gD;MAAA,gE;MA2uBA,gD;QAgBW,6B;;UA5uBP,IAn+ZO,qBAAQ,CAm+Zf,C;YAAe,4BA
AO,OA4uBI,OA5uBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sBA2uBlB,OA3uB
kB,C;UAA5C,akB/ppBO,W;UlBgqpBP,kBA0uB0B,O;UAzuB1B,wD;YACI,cAwuB+B,SAxuBjB,CAAU,KAAV,E
AAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAquBP,g
C;O;KAhBJ,C;8FAmBA,yB;MAruBA,gD;MAAA,gE;MAquBA,gD;QAgBW,6B;;UAtuBP,IAp/ZO,qBAAQ,CAo/
Zf,C;YAAe,4BAAO,OAsuBI,OAtuBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,C;UAA+B,sB
AquBlB,OAruBkB,C;UAA5C,akBxrpBO,W;UlByrpBP,kBAouB0B,O;UAnuB1B,wD;YACI,cAkuB+B,SAluBjB,C
AAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,
M;;;QA+tBP,gC;O;KAhBJ,C;8FAmBA,yB;MA/tBA,gD;MAAA,gE;MA+tBA,gD;QAgBW,6B;;UAhuBP,IArgaO,q
BAAQ,CAqgaf,C;YAAe,4BAAO,OAguBI,OAhuBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mBAAO,CAAP,IAAb,
C;UAA+B,sBA+tBlB,OA/tBkB,C;UAA5C,akBjtpBO,W;UlBktpBP,kBA8tB0B,O;UA7tB1B,wD;YACI,cA4tB+B,S
A5tBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,UAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX
,4BAAO,M;;;QAytBP,gC;O;KAhBJ,C;8FAmBA,yB;MAztBA,gD;MAAA,gE;MAAA,oC;MAytBA,gD;QAgBW,6
B;;UA1tBP,IAthaO,qBAAQ,CAshaf,C;YAAe,4BAAO,OA0tBI,OA1tBJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,mB
AAO,CAAP,IAAb,C;UAA+B,sBAytBlB,OAztBkB,C;UAA5C,akB1upBO,W;UlB2upBP,kBAwtB0B,O;UAvtB1B,
wD;YACI,cAstB+B,SAttBjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,EAA9B,C;YACd,MAAO,W
AAI,WAAJ,C;;UAEX,4BAAO,M;;;QAmtBP,gC;O;KAhBJ,C;gFAmBA,+B;MAOoB,Q;MADhB,UAAe,C;MACf,
wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAG
X,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,
OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAg
B,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UA
Ae,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO
,G;K;kFAGX,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YA
AO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAh
B,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;
MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;
MAEJ,OAAO,G;K;kFAGX,+B;MAOoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAh
B,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;kFAGX,yB;MAAA,oC;MAAA,gC;MAAA,sC;QAO
oB,Q;QADhB,UAAe,C;QACf,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,YAAO,SAAS,oB
AAT,CAAP,I;;QAEJ,OAAO,G;O;KAVX,C;4FAaA,+B;MAOoB,Q;MADhB,UAAkB,G;MAClB,wBAAgB,SAAhB
,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADh
B,UAAkB,G;MAClB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OA
AO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MAClB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;Q
ACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MAClB,wBAAgB,
SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q
;MADhB,UAAkB,G;MAClB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MA
EX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MAClB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAh
B,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MAClB,wB
AAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;8FAGX,+B;MA
OoB,Q;MADhB,UAAkB,G;MAClB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,
C;;MAEX,OAAO,G;K;8FAGX,yB;MAAA,oC;MAAA,gC;MAAA,sC;QAOoB,Q;QADhB,UAAkB,G;QAClB,wBA
AgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,OAAO,SAAS,oBAAT,C;;QAEX,OAAO,G;O;KAVX,
C;gFAaA,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OA
AO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,

gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADh
B,UAAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,O
AAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;
QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAg
B,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,
Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;
MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB,wBAAgB,SAAhB,gB;QAAgB,cAAA,S
AAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAoB,C;MACpB
,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,yB;
MAAA,oC;MAAA,gC;MAAA,sC;QAUoB,Q;QADhB,UAAoB,C;QACpB,wBAAgB,SAAhB,gB;UAAgB,cAAhB,
UAAgB,SAAhB,O;UACI,OAAO,SAAS,oBAAT,C;;QAEX,OAAO,G;O;KAbX,C;kFAgBA,+B;MAUoB,Q;MADh
B,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,O
AAO,G;K;kFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QA
CI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,
SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MA
UoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CA
AP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,
SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;M
ACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;m
FAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,S
AAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;
QAAgB,cAAA,SAAhB,M;QACI,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,yB;MAAA,oC;MAA
A,gC;MAAA,sC;QAUoB,Q;QADhB,UAAe,C;QACf,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;U
ACI,YAAO,SAAS,oBAAT,CAAP,I;;QAEJ,OAAO,G;O;KAbX,C;mFAgBA,yB;MAAA,SASoB,gB;MATpB,sC;Q
AUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C
;;QAEJ,OAAO,G;O;KAbX,C;mFAgBA,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QACA,wBAAg
B,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAg
BA,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAh
B,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAgBA,yB;MAAA,SASoB,gB;MATpB,
sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CA
AP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAgBA,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QACA,wB
AAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;m
FAgBA,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,S
AAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAgBA,yB;MAAA,SASoB,gB;MA
TpB,sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAA
T,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C;mFAgBA,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QAC
A,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAb
X,C;mFAgBA,yB;MAAA,SASoB,gB;MATpB,oC;MAAA,gC;MAAA,sC;QAUoB,Q;QADhB,Y;QACA,wBAAgB,
SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,cAAO,SAAS,oBAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,
C;mFAiBA,6B;MGjyrBA,6B;MHiyrBA,sC;QAWoB,Q;QADhB,UGjyrBmC,cHiyrBnB,CGjyrBmB,C;QHkyrBnC,
wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MG/msBiD,cH+msBjD,GG/msB2D,KAAK,GH+msBzD,S
AAS,OAAT,CG/msBoE,KAAX,IAAf,C;;QHinsBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MGlzrBA,6B;MHkzrBA,s
C;QAWoB,Q;QADhB,UGlzrBmC,cHkzrBnB,CGlzrBmB,C;QHmzrBnC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAA
hB,M;UACI,MGhosBiD,cHgosBjD,GGhosB2D,KAAK,GHgosBzD,SAAS,OAAT,CGhosBoE,KAAX,IAAf,C;;Q
HkosBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MGn0rBA,6B;MHm0rBA,sC;QAWoB,Q;QADhB,UGn0rBmC,cHm0
rBnB,CGn0rBmB,C;QHo0rBnC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MGjpsBiD,cHipsBjD,GG
jpsB2D,KAAK,GHipsBzD,SAAS,OAAT,CGjpsBoE,KAAX,IAAf,C;;QHmpsBrD,OAAO,G;O;KAdX,C;mFAiBA,
yB;MGp1rBA,6B;MHo1rBA,sC;QAWoB,Q;QADhB,UGp1rBmC,cHo1rBnB,CGp1rBmB,C;QHq1rBnC,wBAAgB,

SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MGlqsBiD,cHkqsBjD,GGlqsB2D,KAAK,GHkqsBzD,SAAS,OAAT,
CGlqsBoE,KAAX,IAAf,C;;QHoqsBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MGr2rBA,6B;MHq2rBA,sC;QAWoB,
Q;QADhB,UGr2rBmC,cHq2rBnB,CGr2rBmB,C;QHs2rBnC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UA
CI,MGnrsBiD,cHmrsBjD,GGnrsB2D,KAAK,GHmrsBzD,SAAS,OAAT,CGnrsBoE,KAAX,IAAf,C;;QHqrsBrD,O
AAO,G;O;KAdX,C;mFAiBA,yB;MGt3rBA,6B;MHs3rBA,sC;QAWoB,Q;QADhB,UGt3rBmC,cHs3rBnB,CGt3rB
mB,C;QHu3rBnC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MGpssBiD,cHossBjD,GGpssB2D,KAA
K,GHossBzD,SAAS,OAAT,CGpssBoE,KAAX,IAAf,C;;QHsssBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MGv4rBA,
6B;MHu4rBA,sC;QAWoB,Q;QADhB,UGv4rBmC,cHu4rBnB,CGv4rBmB,C;QHw4rBnC,wBAAgB,SAAhB,gB;U
AAgB,cAAA,SAAhB,M;UACI,MGrtsBiD,cHqtsBjD,GGrtsB2D,KAAK,GHqtsBzD,SAAS,OAAT,CGrtsBoE,KAA
X,IAAf,C;;QHutsBrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MGx5rBA,6B;MHw5rBA,sC;QAWoB,Q;QADhB,UGx5
rBmC,cHw5rBnB,CGx5rBmB,C;QHy5rBnC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MGtusBiD,c
HsusBjD,GGtusB2D,KAAK,GHsusBzD,SAAS,OAAT,CGtusBoE,KAAX,IAAf,C;;QHwusBrD,OAAO,G;O;KAdX
,C;mFAiBA,yB;MAAA,oC;MAAA,gC;MGz6rBA,6B;MHy6rBA,sC;QAWoB,Q;QADhB,UGz6rBmC,cHy6rBnB,C
Gz6rBmB,C;QH06rBnC,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,MGvvsBiD,cHuvsBjD,
GGvvsB2D,KAAK,GHuvsBzD,SAAS,oBAAT,CGvvsBoE,KAAX,IAAf,C;;QHyvsBrD,OAAO,G;O;KAdX,C;mFA
iBA,yB;MoBv7rBA,+B;MpBu7rBA,sC;QAWoB,Q;QADhB,UoBt7rBqC,eAAW,oBpBs7rB/B,CoBt7rB+B,CAAX,
C;QpBu7rBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBrwsBmD,epBqwsBnD,GoBrwsB8D,K
AAK,KpBqwsB5D,SAAS,OAAT,CoBrwsBuE,KAAX,CAAhB,C;;QpBuwsBvD,OAAO,G;O;KAdX,C;mFAiBA,y
B;MoBx8rBA,+B;MpBw8rBA,sC;QAWoB,Q;QADhB,UoBv8rBqC,eAAW,oBpBu8rB/B,CoBv8rB+B,CAAX,C;Q
pBw8rBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBtxsBmD,epBsxsBnD,GoBtxsB8D,KAAK,
KpBsxsB5D,SAAS,OAAT,CoBtxsBuE,KAAX,CAAhB,C;;QpBwxsBvD,OAAO,G;O;KAdX,C;mFAiBA,yB;MoBz
9rBA,+B;MpBy9rBA,sC;QAWoB,Q;QADhB,UoBx9rBqC,eAAW,oBpBw9rB/B,CoBx9rB+B,CAAX,C;QpBy9rBr
C,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBvysBmD,epBuysBnD,GoBvysB8D,KAAK,KpBuys
B5D,SAAS,OAAT,CoBvysBuE,KAAX,CAAhB,C;;QpByysBvD,OAAO,G;O;KAdX,C;kFAiBA,yB;MoB1+rBA,+
B;MpB0+rBA,sC;QAWoB,Q;QADhB,UoBz+rBqC,eAAW,oBpBy+rB/B,CoBz+rB+B,CAAX,C;QpB0+rBrC,wBA
AgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBxzsBmD,epBwzsBnD,GoBxzsB8D,KAAK,KpBwzsB5D,S
AAS,OAAT,CoBxzsBuE,KAAX,CAAhB,C;;QpB0zsBvD,OAAO,G;O;KAdX,C;mFAiBA,yB;MoB3/rBA,+B;MpB
2/rBA,sC;QAWoB,Q;QADhB,UoB1/rBqC,eAAW,oBpB0/rB/B,CoB1/rB+B,CAAX,C;QpB2/rBrC,wBAAgB,SAAh
B,gB;UAAgB,cAAA,SAAhB,M;UACI,MoBz0sBmD,epBy0sBnD,GoBz0sB8D,KAAK,KpBy0sB5D,SAAS,OAAT,
CoBz0sBuE,KAAX,CAAhB,C;;QpB20sBvD,OAAO,G;O;KAdX,C;mFAiBA,yB;MoB5gsBA,+B;MpB4gsBA,sC;Q
AWoB,Q;QADhB,UoB3gsBqC,eAAW,oBpB2gsB/B,CoB3gsB+B,CAAX,C;QpB4gsBrC,wBAAgB,SAAhB,gB;U
AAgB,cAAA,SAAhB,M;UACI,MoB11sBmD,epB01sBnD,GoB11sB8D,KAAK,KpB01sB5D,SAAS,OAAT,CoB11
sBuE,KAAX,CAAhB,C;;QpB41sBvD,OAAO,G;O;KAdX,C;mFAiBA,yB;MoB7hsBA,+B;MpB6hsBA,sC;QAWoB
,Q;QADhB,UoB5hsBqC,eAAW,oBpB4hsB/B,CoB5hsB+B,CAAX,C;QpB6hsBrC,wBAAgB,SAAhB,gB;UAAgB,c
AAA,SAAhB,M;UACI,MoB32sBmD,epB22sBnD,GoB32sB8D,KAAK,KpB22sB5D,SAAS,OAAT,CoB32sBuE,K
AAX,CAAhB,C;;QpB62sBvD,OAAO,G;O;KAdX,C;kFAiBA,yB;MoB9isBA,+B;MpB8isBA,sC;QAWoB,Q;QADh
B,UoB7isBqC,eAAW,oBpB6isB/B,CoB7isB+B,CAAX,C;QpB8isBrC,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAh
B,M;UACI,MoB53sBmD,epB43sBnD,GoB53sB8D,KAAK,KpB43sB5D,SAAS,OAAT,CoB53sBuE,KAAX,CAAh
B,C;;QpB83sBvD,OAAO,G;O;KAdX,C;mFAiBA,yB;MAAA,oC;MAAA,gC;MoB/jsBA,+B;MpB+jsBA,sC;QAWo
B,Q;QADhB,UoB9jsBqC,eAAW,oBpB8jsB/B,CoB9jsB+B,CAAX,C;QpB+jsBrC,wBAAgB,SAAhB,gB;UAAgB,c
AAhB,UAAgB,SAAhB,O;UACI,MoB74sBmD,epB64sBnD,GoB74sB8D,KAAK,KpB64sB5D,SAAS,oBAAT,CoB
74sBuE,KAAX,CAAhB,C;;QpB+4sBvD,OAAO,G;O;KAdX,C;IAiBA,mC;MAIoB,UAMT,M;MANP,wBAAgB,S
AAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,eAAJ,C;UACI,MAAM,gCAAyB,2BAAwB,SAAxB,MAAzB,C;;
;MAId,OAAO,0D;K;wFAGX,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,g
B;QACb,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WA
AI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoB
A,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAgB,SAAhB
,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAA

O,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6
B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAh
B,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;QAGf,
OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;Q
AFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,O
AAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAA
Y,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,a
AAa,gB;QACb,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAA
M,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0
FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAgB,S
AAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,
MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MA
AA,6B;MAAA,uC;QAUoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAgB,SAAhB,gB;UAAgB,cAAA,S
AAhB,M;UACI,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;Q
AGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB
,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UACI,IAAI,UAA
U,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,E
AAY,MAAZ,C;O;KAjBX,C;0FAoBA,yB;MAAA,+D;MAAA,oC;MAAA,gC;MAAA,6B;MAAA,uC;QAUoB,Q;Q
AFhB,YAAY,gB;QACZ,aAAa,gB;QACb,wBAAgB,SAAhB,gB;UAAgB,cAAhB,UAAgB,SAAhB,O;UACI,IAAI,
UAAU,oBAAV,CAAJ,C;YACI,KAAM,WAAI,oBAAJ,C;;YAEN,MAAO,WAAI,oBAAJ,C;;;QAGf,OAAO,cAAK,
KAAL,EAAY,MAAZ,C;O;KAjBX,C;IAoBA,+B;MAkGI,WmBpgtBO,MAAO,KnBogtBG,gBmBpgtBH,EnBy6sB
H,KA2FkB,OmBpgtBf,C;MnBqgtBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,IA
AK,WA9FqB,GA8FP,UAAK,CAAL,CA9FO,EAAnB,KA8FqB,CAAM,CAAN,CA9FF,CA8FrB,C;;MA9FT,OAgG
O,I;K;IA7FX,iC;MAwGI,WmBphtBO,MAAO,KnBohtBG,gBmBphtBH,EnBm7sBH,KAiGkB,OmBphtBf,C;MnBq
htBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,IAAK,WApGqB,GAoGP,UAAK,C
AAL,CApGO,EAAnB,KAoGqB,CAAM,CAAN,CApGF,CAoGrB,C;;MApGT,OAsGO,I;K;IAnGX,iC;MA8GI,Wm
BpitBO,MAAO,KnBoitBG,gBmBpitBH,EnB67sBH,KAuGkB,OmBpitBf,C;MnBqitBd,WAAW,iBAAa,IAAb,C;M
ACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,IAAK,WA1GqB,GA0GP,UAAK,CAAL,CA1GO,EAAnB,KA0GqB,
CAAM,CAAN,CA1GF,CA0GrB,C;;MA1GT,OA4GO,I;K;IAzGX,iC;MAoHI,WmBpjtBO,MAAO,KnBojtBG,gBm
BpjtBH,EnBu8sBH,KA6GkB,OmBpjtBf,C;MnBqjtBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IA
AlB,M;QACI,IAAK,WAhHqB,GAgHP,UAAK,CAAL,CAhHO,EAAnB,KAgHqB,CAAM,CAAN,CAhHF,CAgHr
B,C;;MAhHT,OAkHO,I;K;IA/GX,iC;MA0HI,WmBpktBO,MAAO,KnBoktBG,gBmBpktBH,EnBi9sBH,KAmHkB,
OmBpktBf,C;MnBqktBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,IAAK,WAtHq
B,GAsHP,UAAK,CAAL,CAtHO,EAAnB,KAsHqB,CAAM,CAAN,CAtHF,CAsHrB,C;;MAtHT,OAwHO,I;K;IArH
X,iC;MAgII,WmBpltBO,MAAO,KnBoltBG,gBmBpltBH,EnB29sBH,KAyHkB,OmBpltBf,C;MnBqltBd,WAAW,i
BAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,IAAK,WA5HqB,GA4HP,UAAK,CAAL,CA5HO,
EAAnB,KA4HqB,CAAM,CAAN,CA5HF,CA4HrB,C;;MA5HT,OA8HO,I;K;IA3HX,iC;MAsII,WmBpmtBO,MAA
O,KnBomtBG,gBmBpmtBH,EnBq+sBH,KA+HkB,OmBpmtBf,C;MnBqmtBd,WAAW,iBAAa,IAAb,C;MACX,aA
AU,CAAV,MAAkB,IAAlB,M;QACI,IAAK,WAlIqB,GAkIP,UAAK,CAAL,CAlIO,EAAnB,KAkIqB,CAAM,CAA
N,CAlIF,CAkIrB,C;;MAlIT,OAoIO,I;K;IAjIX,iC;MA4II,WmBpntBO,MAAO,KnBontBG,gBmBpntBH,EnB++sB
H,KAqIkB,OmBpntBf,C;MnBqntBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,IAA
K,WAxIqB,GAwIP,UAAK,CAAL,CAxIO,EAAnB,KAwIqB,CAAM,CAAN,CAxIF,CAwIrB,C;;MAxIT,OA0IO,I;
K;IAvIX,iC;MAkJI,WmBpotBO,MAAO,KnBootBG,gBmBpotBH,EnBy/sBH,KA2IkB,OmBpotBf,C;MnBqotBd,W
AAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,IAAK,WA9IqB,GA8IP,sBAAK,CAAL,EA
9IO,EAAnB,KA8IqB,CAAM,CAAN,CA9IF,CA8IrB,C;;MA9IT,OAgJO,I;K;8EA7IX,yB;MAAA,gE;MmBlgtBA,i
B;MnBkgtBA,8C;QAQI,WmBpgtBO,MAAO,KnBogtBG,gBmBpgtBH,EnBogtBS,KAAM,OmBpgtBf,C;QnBqgtB
d,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CA
AV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;8EAgBA,yB;MAAA,gE;MmBlhtBA,i

B;MnBkhtBA,8C;QAQI,WmBphtBO,MAAO,KnBohtBG,gBmBphtBH,EnBohtBS,KAAM,OmBphtBf,C;QnBqhtB
d,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CA
AV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmBlitBA,i
B;MnBkitBA,8C;QAQI,WmBpitBO,MAAO,KnBoitBG,gBmBpitBH,EnBoitBS,KAAM,OmBpitBf,C;QnBqitBd,W
AAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,
EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;8EAgBA,yB;MAAA,gE;MmBljtBA,iB;M
nBkjtBA,8C;QAQI,WmBpjtBO,MAAO,KnBojtBG,gBmBpjtBH,EnBojtBS,KAAM,OmBpjtBf,C;QnBqjtBd,WAA
W,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EA
AmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmBlktBA,iB;MnB
kktBA,8C;QAQI,WmBpktBO,MAAO,KnBoktBG,gBmBpktBH,EnBoktBS,KAAM,OmBpktBf,C;QnBqktBd,WAA
W,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EA
AmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmBlltBA,iB;MnBk
ltBA,8C;QAQI,WmBpltBO,MAAO,KnBoltBG,gBmBpltBH,EnBoltBS,KAAM,OmBpltBf,C;QnBqltBd,WAAW,e
AAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAmB
,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmBlmtBA,iB;MnBkmt
BA,8C;QAQI,WmBpmtBO,MAAO,KnBomtBG,gBmBpmtBH,EnBomtBS,KAAM,OmBpmtBf,C;QnBqmtBd,WA
AW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,E
AAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmBlntBA,iB;Mn
BkntBA,8C;QAQI,WmBpntBO,MAAO,KnBontBG,gBmBpntBH,EnBontBS,KAAM,OmBpntBf,C;QnBqntBd,WA
AW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,E
AAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MAAA,oC;MmBlo
tBA,iB;MnBkotBA,8C;QAQI,WmBpotBO,MAAO,KnBootBG,gBmBpotBH,EnBootBS,KAAM,OmBpotBf,C;QnB
qotBd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAA
L,EAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;IAgBA,kC;MAqGoB,gB;MAHh
B,gBAAgB,gB;MAChB,WAAW,iBmB/utBJ,MAAO,KnB+utBsB,wBA5FzB,KA4FyB,EAAwB,EAAxB,CmB/utBt
B,EnB+utBmD,SmB/utBnD,CnB+utBH,C;MACX,QAAQ,C;MACQ,OA9FL,KA8FK,W;MAAhB,OAAgB,cAAhB,
C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhGqB,GAgGP,UAAK,UAAL,EAAK,k
BAAL,SAhGO,EAgGI,OAhGJ,CAgGrB,C;;MAhGT,OAkGO,I;K;IA/FX,kC;MA6GoB,gB;MAHhB,gBAAgB,gB;
MAChB,WAAW,iBmBjwtBJ,MAAO,KnBiwtBsB,wBApGzB,KAoGyB,EAAwB,EAAxB,CmBjwtBtB,EnBiwtBm
D,SmBjwtBnD,CnBiwtBH,C;MACX,QAAQ,C;MACQ,OAtGL,KAsGK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,y
B;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAxGqB,GAwGP,UAAK,UAAL,EAAK,kBAAL,SAx
GO,EAwGI,OAxGJ,CAwGrB,C;;MAxGT,OA0GO,I;K;IAvGX,kC;MAqHoB,gB;MAHhB,gBAAgB,gB;MAChB,
WAAW,iBmBnxtBJ,MAAO,KnBmxtBsB,wBA5GzB,KA4GyB,EAAwB,EAAxB,CmBnxtBtB,EnBmxtBmD,SmBn
xtBnD,CnBmxtBH,C;MACX,QAAQ,C;MACQ,OA9GL,KA8GK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAC
Z,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhHqB,GAgHP,UAAK,UAAL,EAAK,kBAAL,SAhHO,EAg
HI,OAhHJ,CAgHrB,C;;MAhHT,OAkHO,I;K;IA/GX,kC;MA6HoB,gB;MAHhB,gBAAgB,gB;MAChB,WAAW,iB
mBrytBJ,MAAO,KnBqytBsB,wBApHzB,KAoHyB,EAAwB,EAAxB,CmBrytBtB,EnBqytBmD,SmBrytBnD,CnBq
ytBH,C;MACX,QAAQ,C;MACQ,OAtHL,KAsHK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAA
K,SAAT,C;UAAoB,K;QACpB,IAAK,WAxHqB,GAwHP,UAAK,UAAL,EAAK,kBAAL,SAxHO,EAwHI,OAxHJ,
CAwHrB,C;;MAxHT,OA0HO,I;K;IAvHX,kC;MAqIoB,gB;MAHhB,gBAAgB,gB;MAChB,WAAW,iBmBvztBJ,M
AAO,KnBuztBsB,wBA5HzB,KA4HyB,EAAwB,EAAxB,CmBvztBtB,EnBuztBmD,SmBvztBnD,CnBuztBH,C;MA
CX,QAAQ,C;MACQ,OA9HL,KA8HK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;
UAAoB,K;QACpB,IAAK,WAhIqB,GAgIP,UAAK,UAAL,EAAK,kBAAL,SAhIO,EAgII,OAhIJ,CAgIrB,C;;MAhI
T,OAkIO,I;K;IA/HX,kC;MA6IoB,gB;MAHhB,gBAAgB,gB;MAChB,WAAW,iBmBz0tBJ,MAAO,KnBy0tBsB,wB
ApIzB,KAoIyB,EAAwB,EAAxB,CmBz0tBtB,EnBy0tBmD,SmBz0tBnD,CnBy0tBH,C;MACX,QAAQ,C;MACQ,
OAtIL,KAsIK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK
,WAxIqB,GAwIP,UAAK,UAAL,EAAK,kBAAL,SAxIO,EAwII,OAxIJ,CAwIrB,C;;MAxIT,OA0IO,I;K;IAvIX,kC;
MAqJoB,gB;MAHhB,gBAAgB,gB;MAChB,WAAW,iBmB31tBJ,MAAO,KnB21tBsB,wBA5IzB,KA4IyB,EAAwB

,EAAxB,CmB31tBtB,EnB21tBmD,SmB31tBnD,CnB21tBH,C;MACX,QAAQ,C;MACQ,OA9IL,KA8IK,W;MAAh
B,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhJqB,GAgJP,UAA
K,UAAL,EAAK,kBAAL,SAhJO,EAgJI,OAhJJ,CAgJrB,C;;MAhJT,OAkJO,I;K;IA/IX,kC;MA6JoB,gB;MAHhB,g
BAAgB,gB;MAChB,WAAW,iBmB72tBJ,MAAO,KnB62tBsB,wBApJzB,KAoJyB,EAAwB,EAAxB,CmB72tBtB,
EnB62tBmD,SmB72tBnD,CnB62tBH,C;MACX,QAAQ,C;MACQ,OAtJL,KAsJK,W;MAAhB,OAAgB,cAAhB,C;
QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAxJqB,GAwJP,UAAK,UAAL,EAAK,kBA
AL,SAxJO,EAwJI,OAxJJ,CAwJrB,C;;MAxJT,OA0JO,I;K;IAvJX,kC;MAqKoB,gB;MAHhB,gBAAgB,gB;MAChB
,WAAW,iBmB/3tBJ,MAAO,KnB+3tBsB,wBA5JzB,KA4JyB,EAAwB,EAAxB,CmB/3tBtB,EnB+3tBmD,SmB/3tB
nD,CnB+3tBH,C;MACX,QAAQ,C;MACQ,OA9JL,KA8JK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IA
AI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhKqB,GAgKP,sBAAK,UAAL,EAAK,kBAAL,UAhKO,EAgKI,
OAhKJ,CAgKrB,C;;MAhKT,OAkKO,I;K;+EA/JX,yB;MAAA,kF;MAAA,gE;MmB5utBA,iB;MnB4utBA,8C;QA
WoB,UAEY,M;QAL5B,gBAAgB,gB;QAChB,WAAW,emB/utBJ,MAAO,KnB+utBsB,wBAAN,KAAM,EAAwB,E
AAxB,CmB/utBtB,EnB+utBmD,SmB/utBnD,CnB+utBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,
C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAL,EAAK,kBA
AL,SAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KAfX,C;+EAkBA,yB;MAAA,kF;MAAA,gE;MmB9vtBA
,iB;MnB8vtBA,8C;QAWoB,UAEY,M;QAL5B,gBAAgB,gB;QAChB,WAAW,emBjwtBJ,MAAO,KnBiwtBsB,wB
AAN,KAAM,EAAwB,EAAxB,CmBjwtBtB,EnBiwtBmD,SmBjwtBnD,CnBiwtBH,C;QACX,QAAQ,C;QACQ,uB;
QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,U
AAK,UAAL,EAAK,kBAAL,SAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KAfX,C;+EAkBA,yB;MAAA,k
F;MAAA,gE;MmBhxtBA,iB;MnBgxtBA,8C;QAWoB,UAEY,M;QAL5B,gBAAgB,gB;QAChB,WAAW,emBnxtBJ
,MAAO,KnBmxtBsB,wBAAN,KAAM,EAAwB,EAAxB,CmBnxtBtB,EnBmxtBmD,SmBnxtBnD,CnBmxtBH,C;Q
ACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UAC
pB,IAAK,WAAI,UAAU,UAAK,UAAL,EAAK,kBAAL,SAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KAf
X,C;+EAkBA,yB;MAAA,kF;MAAA,gE;MmBlytBA,iB;MnBkytBA,8C;QAWoB,UAEY,M;QAL5B,gBAAgB,gB;
QAChB,WAAW,emBrytBJ,MAAO,KnBqytBsB,wBAAN,KAAM,EAAwB,EAAxB,CmBrytBtB,EnBqytBmD,SmB
rytBnD,CnBqytBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,S
AAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAL,EAAK,kBAAL,SAAV,EAAqB,OAArB,CAAJ,C;;
QAET,OAAO,I;O;KAfX,C;+EAkBA,yB;MAAA,kF;MAAA,gE;MmBpztBA,iB;MnBoztBA,8C;QAWoB,UAEY,M
;QAL5B,gBAAgB,gB;QAChB,WAAW,emBvztBJ,MAAO,KnBuztBsB,wBAAN,KAAM,EAAwB,EAAxB,CmBvzt
BtB,EnBuztBmD,SmBvztBnD,CnBuztBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;
UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAL,EAAK,kBAAL,SAAV,EA
AqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KAfX,C;+EAkBA,yB;MAAA,kF;MAAA,gE;MmBt0tBA,iB;MnBs0tBA
,8C;QAWoB,UAEY,M;QAL5B,gBAAgB,gB;QAChB,WAAW,emBz0tBJ,MAAO,KnBy0tBsB,wBAAN,KAAM,E
AAwB,EAAxB,CmBz0tBtB,EnBy0tBmD,SmBz0tBnD,CnBy0tBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAg
B,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,UAAK,UAAL,E
AAK,kBAAL,SAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KAfX,C;+EAkBA,yB;MAAA,kF;MAAA,gE;M
mBx1tBA,iB;MnBw1tBA,8C;QAWoB,UAEY,M;QAL5B,gBAAgB,gB;QAChB,WAAW,emB31tBJ,MAAO,KnB2
1tBsB,wBAAN,KAAM,EAAwB,EAAxB,CmB31tBtB,EnB21tBmD,SmB31tBnD,CnB21tBH,C;QACX,QAAQ,C;Q
ACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,
UAAU,UAAK,UAAL,EAAK,kBAAL,SAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KAfX,C;+EAkBA,yB;
MAAA,kF;MAAA,gE;MmB12tBA,iB;MnB02tBA,8C;QAWoB,UAEY,M;QAL5B,gBAAgB,gB;QAChB,WAAW,e
mB72tBJ,MAAO,KnB62tBsB,wBAAN,KAAM,EAAwB,EAAxB,CmB72tBtB,EnB62tBmD,SmB72tBnD,CnB62tB
H,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K
;UACpB,IAAK,WAAI,UAAU,UAAK,UAAL,EAAK,kBAAL,SAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O
;KAfX,C;+EAkBA,yB;MAAA,kF;MAAA,gE;MAAA,oC;MmB53tBA,iB;MnB43tBA,8C;QAWoB,UAEY,M;QAL
5B,gBAAgB,gB;QAChB,WAAW,emB/3tBJ,MAAO,KnB+3tBsB,wBAAN,KAAM,EAAwB,EAAxB,CmB/3tBtB,E
nB+3tBmD,SmB/3tBnD,CnB+3tBH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ
,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,sBAAK,UAAL,EAAK,kBAAL,UAAV,EAAqB,

OAArB,CAAJ,C;;QAET,OAAO,I;O;KAfX,C;IAkBA,kC;MAwFI,WmBh+tBO,MAAO,KnBg+tBG,gBmBh+tBH,E
nB+4tBH,KAiFkB,OmBh+tBf,C;MnBi+tBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;Q
ACI,IAAK,WApFqB,GAoFP,UAAK,CAAL,CApFO,EAAnB,KAoFqB,CAAM,CAAN,CApFF,CAoFrB,C;;MApF
T,OAsFO,I;K;IAnFX,kC;MA8FI,WmBh/tBO,MAAO,KnBg/tBG,gBmBh/tBH,EnBy5tBH,KAuFkB,OmBh/tBf,C;
MnBi/tBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,IAAK,WA1FqB,GA0FP,UAA
K,CAAL,CA1FO,EAAnB,KA0FqB,CAAM,CAAN,CA1FF,CA0FrB,C;;MA1FT,OA4FO,I;K;IAzFX,kC;MAoGI,W
mBhguBO,MAAO,KnBgguBG,gBmBhguBH,EnBm6tBH,KA6FkB,OmBhguBf,C;MnBiguBd,WAAW,iBAAa,IAA
b,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,IAAK,WAhGqB,GAgGP,UAAK,CAAL,CAhGO,EAAnB,KA
gGqB,CAAM,CAAN,CAhGF,CAgGrB,C;;MAhGT,OAkGO,I;K;IA/FX,kC;MA0GI,WmBhhuBO,MAAO,KnBghu
BG,gBmBhhuBH,EnB66tBH,KAmGkB,OmBhhuBf,C;MnBihuBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,
MAAkB,IAAlB,M;QACI,IAAK,WAtGqB,GAsGP,UAAK,CAAL,CAtGO,EAAnB,KAsGqB,CAAM,CAAN,CAtG
F,CAsGrB,C;;MAtGT,OAwGO,I;K;IArGX,kC;MAgHI,WmBhiuBO,MAAO,KnBgiuBG,gBmBhiuBH,EnBu7tBH,
KAyGkB,OmBhiuBf,C;MnBiiuBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,IAAK
,WA5GqB,GA4GP,UAAK,CAAL,CA5GO,EAAnB,KA4GqB,CAAM,CAAN,CA5GF,CA4GrB,C;;MA5GT,OA8G
O,I;K;IA3GX,kC;MAsHI,WmBhjuBO,MAAO,KnBgjuBG,gBmBhjuBH,EnBi8tBH,KA+GkB,OmBhjuBf,C;MnBij
uBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,IAAK,WAlHqB,GAkHP,UAAK,CA
AL,CAlHO,EAAnB,KAkHqB,CAAM,CAAN,CAlHF,CAkHrB,C;;MAlHT,OAoHO,I;K;IAjHX,kC;MA4HI,WmBh
kuBO,MAAO,KnBgkuBG,gBmBhkuBH,EnB28tBH,KAqHkB,OmBhkuBf,C;MnBikuBd,WAAW,iBAAa,IAAb,C;
MACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,IAAK,WAxHqB,GAwHP,UAAK,CAAL,CAxHO,EAAnB,KAwH
qB,CAAM,CAAN,CAxHF,CAwHrB,C;;MAxHT,OA0HO,I;K;IAvHX,kC;MAkII,WmBhluBO,MAAO,KnBgluBG,
gBmBhluBH,EnBq9tBH,KA2HkB,OmBhluBf,C;MnBiluBd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAk
B,IAAlB,M;QACI,IAAK,WA9HqB,GA8HP,sBAAK,CAAL,EA9HO,EA8HE,YA9HrB,KA8HqB,CAAM,CAAN,E
A9HF,CA8HrB,C;;MA9HT,OAgIO,I;K;+EA7HX,yB;MAAA,gE;MmB99tBA,iB;MnB89tBA,8C;QAQI,WmBh+tB
O,MAAO,KnBg+tBG,gBmBh+tBH,EnBg+tBS,KAAM,OmBh+tBf,C;QnBi+tBd,WAAW,eAAa,IAAb,C;QACX,aA
AU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAn
B,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB9+tBA,iB;MnB8+tBA,8C;QAQI,WmBh/tB
O,MAAO,KnBg/tBG,gBmBh/tBH,EnBg/tBS,KAAM,OmBh/tBf,C;QnBi/tBd,WAAW,eAAa,IAAb,C;QACX,aAAU
,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,C
AAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB9/tBA,iB;MnB8/tBA,8C;QAQI,WmBhguBO,M
AAO,KnBgguBG,gBmBhguBH,EnBgguBS,KAAM,OmBhguBf,C;QnBiguBd,WAAW,eAAa,IAAb,C;QACX,aAA
U,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,
CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB9guBA,iB;MnB8guBA,8C;QAQI,WmBhhuB
O,MAAO,KnBghuBG,gBmBhhuBH,EnBghuBS,KAAM,OmBhhuBf,C;QnBihuBd,WAAW,eAAa,IAAb,C;QACX,a
AAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAA
nB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB9huBA,iB;MnB8huBA,8C;QAQI,WmBhi
uBO,MAAO,KnBgiuBG,gBmBhiuBH,EnBgiuBS,KAAM,OmBhiuBf,C;QnBiiuBd,WAAW,eAAa,IAAb,C;QACX,a
AAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAA
nB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB9iuBA,iB;MnB8iuBA,8C;QAQI,WmBhju
BO,MAAO,KnBgjuBG,gBmBhjuBH,EnBgjuBS,KAAM,OmBhjuBf,C;QnBijuBd,WAAW,eAAa,IAAb,C;QACX,a
AAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAA
nB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MmB9juBA,iB;MnB8juBA,8C;QAQI,WmBhku
BO,MAAO,KnBgkuBG,gBmBhkuBH,EnBgkuBS,KAAM,OmBhkuBf,C;QnBikuBd,WAAW,eAAa,IAAb,C;QACX
,aAAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,UAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CA
AnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;+EAgBA,yB;MAAA,gE;MAAA,oC;MmB9kuBA,iB;MnB8kuBA,8C;Q
AQI,WmBhluBO,MAAO,KnBgluBG,gBmBhluBH,EnBgluBS,KAAM,OmBhluBf,C;QnBiluBd,WAAW,eAAa,IAA
b,C;QACX,aAAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,EAAV,EAAmB,kBAA
M,CAAN,EAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;IAgBA,4F;MAQ8D,yB;QAAA,YAA0B,I;MAAM,sB;QA
AA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;

QAAA,YAAoC,I;MAGvN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;
QAAgB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,
CAAR,IAAa,SAAS,KAA1B,C;UACW,gBAAP,MAAO,EAAc,OAAd,EAAuB,SAAvB,C;;UACJ,K;;MAEX,IAAI,S
AAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP
,OAAO,M;K;IAGX,8F;MAQwD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E
;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAGpN,Q;MAFhB,MAA
O,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,iCAA
U,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iB
AAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAE
X,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP
,C;MACP,OAAO,M;K;IAGX,8F;MAQyD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,U
AAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAwC,I;MAGtN,Q;MAF
hB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IA
AI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI
,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,
K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAA
O,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQuD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;
QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAsC,I;MAGlN,
Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;Q
ACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,
C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;
;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAA
O,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQwD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;
MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC
,I;MAGpN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAA,S
AAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAA
S,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAA
Q,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MA
CxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQyD,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,S
AAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAA
A,YAAwC,I;MAGtN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAhB,gB;QAAg
B,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,
IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAEP,MAAO,gB
AAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAAO,gBAAO,S
AAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQ0D,yB;QAAA,YAA0B,I;MAAM,s
B;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAA
O,yB;QAAA,YAAyC,I;MAGxN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wBAAgB,SAAh
B,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,Q
AAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CAAP,C;;YAE
P,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAiC,MAA
O,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQ2D,yB;QAAA,YAA0
B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YA
A0B,K;MAAO,yB;QAAA,YAA0C,I;MAG1N,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACZ,wB
AAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QA
CxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAAO,UAAU,OAAV,CA
AP,C;;YAEP,MAAO,gBAAO,OAAQ,WAAf,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;Q
AAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,8F;MAQwD,yB;Q
AAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;

QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAGpN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;
MACZ,wBAAgB,SAAhB,gB;QAAgB,cAAhB,UAAgB,SAAhB,O;QACI,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,g
BAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACI,IAAI,iBAAJ,C;YACI,MAAO,gBAA
O,UAAU,oBAAV,CAAP,C;;YAEP,MAAO,gBAAO,OAAP,C;;UACR,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ
,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,0F;M
AQyC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,
E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MAClN,OAAO,kBAAO,sBAAP,EAAwB,SAAxB,E
AAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAiF,W;K;IAG5F,4F;MAQ
kC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;M
AAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAC9M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAA
mC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAiF,W;K;IAG5F,4F;MAQmC
,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MA
AI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAwC,I;MAChN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAm
C,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAiF,W;K;IAG5F,4F;MAQiC,y
B;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI
,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAsC,I;MAC5M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,
MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAiF,W;K;IAG5F,4F;MAQkC,yB;
QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,y
B;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAuC,I;MAC9M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,M
AAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAiF,W;K;IAG5F,4F;MAQmC,yB;Q
AAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;
QAAA,YAA0B,K;MAAO,yB;QAAA,YAAwC,I;MAChN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MA
AnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAiF,W;K;IAG5F,4F;MAQoC,yB;QA
AA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;Q
AAA,YAA0B,K;MAAO,yB;QAAA,YAAyC,I;MAClN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAn
C,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAiF,W;K;IAG5F,4F;MAQqC,yB;QAAA,
YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAA
A,YAA0B,K;MAAO,yB;QAAA,YAA0C,I;MACpN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,E
AA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAiF,W;K;IAG5F,4F;MAQkC,yB;QAAA,YA
A0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,Y
AA0B,K;MAAO,yB;QAAA,YAAuC,I;MAC9M,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,EAA
2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAiF,W;K;IAQxE,4C;MAAA,mB;QAAE,OAAK
,qBAAL,eAAK,C;O;K;IAL3B,+B;MAII,IAp8fO,qBAAQ,CAo8ff,C;QAAe,OAAO,W;MACtB,kCAAgB,4BAAhB,
C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,yBAAL,eAAK,C;O;K;IAL3B,iC;MAII,IAp8fO,qBAAQ,CAo8ff,C;QA
Ae,OAAO,W;MACtB,kCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,0BAAL,eAAK,C;O;K;IAL3
B,iC;MAII,IAp8fO,qBAAQ,CAo8ff,C;QAAe,OAAO,W;MACtB,kCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;
QAAE,OAAK,wBAAL,eAAK,C;O;K;IAL3B,iC;MAII,IAp8fO,qBAAQ,CAo8ff,C;QAAe,OAAO,W;MACtB,kCA
AgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,yBAAL,eAAK,C;O;K;IAL3B,iC;MAII,IAp8fO,qBAAQ
,CAo8ff,C;QAAe,OAAO,W;MACtB,kCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,0BAAL,eAA
K,C;O;K;IAL3B,iC;MAII,IAp8fO,qBAAQ,CAo8ff,C;QAAe,OAAO,W;MACtB,kCAAgB,8BAAhB,C;K;IAQgB,8
C;MAAA,mB;QAAE,OAAK,2BAAL,eAAK,C;O;K;IAL3B,iC;MAII,IAp8fO,qBAAQ,CAo8ff,C;QAAe,OAAO,W;
MACtB,kCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,4BAAL,eAAK,C;O;K;IAL3B,iC;MAII,IA
p8fO,qBAAQ,CAo8ff,C;QAAe,OAAO,W;MACtB,kCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,
yBAAL,eAAK,C;O;K;IAL3B,iC;MAII,IAp8fO,qBAAQ,CAo8ff,C;QAAe,OAAO,W;MACtB,kCAAgB,8BAAhB,C
;K;IAUgB,4C;MAAA,mB;QAAE,OAAK,qBAAL,eAAK,C;O;K;IAP3B,+B;MAMI,IA9ggBO,qBAAQ,CA8ggBf,C
;QAAe,OAAO,e;MACtB,kCAAgB,4BAAhB,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK,yBAAL,eAAK,C;O;K;IA
P3B,iC;MAMI,IAhhgBO,qBAAQ,CAghgBf,C;QAAe,OAAO,e;MACtB,kCAAgB,8BAAhB,C;K;IAUgB,8C;MAA
A,mB;QAAE,OAAK,0BAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IAlhgBO,qBAAQ,CAkhgBf,C;QAAe,OAAO,e;MA

CtB,kCAAgB,8BAAhB,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK,wBAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IAph
gBO,qBAAQ,CAohgBf,C;QAAe,OAAO,e;MACtB,kCAAgB,8BAAhB,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK
,yBAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IAthgBO,qBAAQ,CAshgBf,C;QAAe,OAAO,e;MACtB,kCAAgB,8BAAh
B,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK,0BAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IAxhgBO,qBAAQ,CAwhg
Bf,C;QAAe,OAAO,e;MACtB,kCAAgB,8BAAhB,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK,2BAAL,eAAK,C;O;
K;IAP3B,iC;MAMI,IA1hgBO,qBAAQ,CA0hgBf,C;QAAe,OAAO,e;MACtB,kCAAgB,8BAAhB,C;K;IAUgB,8C;
MAAA,mB;QAAE,OAAK,4BAAL,eAAK,C;O;K;IAP3B,iC;MAMI,IA5hgBO,qBAAQ,CA4hgBf,C;QAAe,OAAO,
e;MACtB,kCAAgB,8BAAhB,C;K;IAUgB,8C;MAAA,mB;QAAE,OAAK,yBAAL,eAAK,C;O;K;IAP3B,iC;MAMI,
IA9hgBO,qBAAQ,CA8hgBf,C;QAAe,OAAO,e;MACtB,kCAAgB,8BAAhB,C;K;IAGJ,4B;MAOoB,Q;MAFhB,UA
AkB,G;MAClB,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;
MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAOoB,Q;MAFhB,UAA
kB,G;MAClB,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;M
AEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAOoB,Q;MAFhB,UAAk
B,G;MAClB,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MA
EJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAOoB,Q;MAFhB,UAAkB,
G;MAClB,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,
OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAOoB,Q;MAFhB,UAAkB,G;
MAClB,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,O
AAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAOoB,Q;MAFhB,UAAkB,G;M
AClB,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OA
AW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAMoB,Q;MAFhB,UAAkB,G;MA
ClB,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OAA
W,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAMoB,Q;MAFhB,UAAkB,G;MACl
B,YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OAAW,
UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAMoB,Q;MAFhB,UAAkB,G;MAClB,
YAAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OAAW,U
AAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAMoB,Q;MAFhB,UAAkB,G;MAClB,Y
AAiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OAAW,UA
AS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,8B;MAMoB,Q;MAFhB,UAAkB,G;MAClB,YA
AiB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OAAW,UAA
S,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MAMoB,Q;MAFhB,UAAkB,G;MAClB,YAA
iB,C;MACjB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;QACP,qB;;MAEJ,OAAW,UAAS,
CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,wB;MAMoB,Q;MADhB,UAAe,C;MACf,wBAAgB,
SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAMoB,Q;MADhB,UAA
e,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAM
oB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,OAAP,I;;MAEJ,O
AAO,G;K;IAGX,0B;MAMoB,Q;MADhB,Y;MACA,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,cAA
O,OAAP,C;;MAEJ,OAAO,G;K;IAGX,0B;MAMoB,Q;MADhB,UAAiB,G;MACjB,wBAAgB,SAAhB,gB;QAAgB,
cAAA,SAAhB,M;QACI,OAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAMoB,Q;MADhB,UAAkB,G;MAClB,wBAA
gB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;;MAEX,OAAO,G;K;IAGX,0B;MAKoB,Q;MADhB,U
AAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,O;;MAEX,OAAO,G;K;IAGX,0B;M
AKoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,YAAO,O;;MAEX,OA
AO,G;K;IAGX,0B;MAKoB,Q;MADhB,UAAe,C;MACf,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,Y
AAO,OAAP,I;;MAEJ,OAAO,G;K;IAGX,0B;MAKoB,Q;MADhB,Y;MACA,wBAAgB,SAAhB,gB;QAAgB,cAAA,
SAAhB,M;QACI,cAAO,OAAP,C;;MAEJ,OAAO,G;K;IAGX,0B;MAKoB,Q;MADhB,UAAiB,G;MACjB,wBAAg
B,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAKoB,Q;MADhB,UA
AkB,G;MAClB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,OAAO,O;;MAEX,OAAO,G;K;IcrmwBX,
oD;MAQuF,wC;K;IARvF,8CASI,Y;MAAuC,8B;K;IAT3C,gF;4FOOA,qB;MAOI,OAAO,sBAAI,CAAJ,C;K;4FAG

X,qB;MAOI,OAAO,sBAAI,CAAJ,C;K;4FAGX,qB;MAOI,OAAO,sBAAI,CAAJ,C;K;4FAGX,qB;MAOI,OAAO,s
BAAI,CAAJ,C;K;4FAGX,qB;MAOI,OAAO,sBAAI,CAAJ,C;K;IAGX,wC;MAII,IAAI,oCAAJ,C;QACI,OAAO,yB
AAS,OAAT,C;MACX,OAAO,qBAAQ,OAAR,KAAoB,C;K;IAWG,yC;MAAA,qB;QAAE,MAAM,8BAA0B,iDA
A8C,aAA9C,MAA1B,C;O;K;IAR1C,qC;MAMI,IAAI,8BAAJ,C;QACI,OAAO,sBAAI,KAAJ,C;MACX,OAAO,6B
AAgB,KAAhB,EAAuB,uBAAvB,C;K;0FAGX,4B;MAOI,OAAO,sBAAI,KAAJ,C;K;IAGX,2D;MAcqB,Q;MARjB
,IAAI,8BAAJ,C;QACI,OAAsB,KA8Lf,IAAS,CAAT,IA9Le,KA8LD,IAAS,iBA9LvB,SA8LuB,CAA3B,GA9LI,SA
8LkC,aA9LnB,KA8LmB,CAAtC,GA9L0B,YA8L4B,CA9LnC,KA8LmC,C;;MA7L7D,IAAI,QAAQ,CAAZ,C;QA
CI,OAAO,aAAa,KAAb,C;MACX,eAAe,oB;MACf,YAAY,C;MACZ,OAAO,QAAS,UAAhB,C;QACI,cAAc,QAA
S,O;QACvB,IAAI,WAAS,YAAT,EAAS,oBAAT,OAAJ,C;UACI,OAAO,O;;MAEf,OAAO,aAAa,KAAb,C;K;sGA
GX,yB;MAAA,8D;MAAA,iD;QAOI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,sBAAI,KAAJ,CAAtC,
GAAsD,aAAa,KAAb,C;O;KAPjE,C;IAUA,6C;MAcqB,Q;MARjB,IAAI,8BAAJ,C;QACI,OAAY,YAAL,SAAK,E
AAU,KAAV,C;MAChB,IAAI,QAAQ,CAAZ,C;QACI,OAAO,I;MACX,eAAe,oB;MACf,YAAY,C;MACZ,OAAO,
QAAS,UAAhB,C;QACI,cAAc,QAAS,O;QACvB,IAAI,WAAS,YAAT,EAAS,oBAAT,OAAJ,C;UACI,OAAO,O;;
MAEf,OAAO,I;K;sGAGX,yB;MAAA,sD;MAAA,mC;QAOI,OAAY,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;
gFAUA,gC;MAOW,sB;;QAyHS,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAzHH,SAyHO,C
AAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MA1HP,yB;K;wFAGJ,gC;MA6VoB,Q
;MADhB,WAAe,I;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAvVc,SAuVV,CAAU,OAAV,CAA
J,C;UACI,OAAO,O;;;MAxVf,OA2VO,I;K;wFAxVX,gC;MAOW,qB;;QA0VP,eAAoB,+BAAa,cAAb,C;QACpB,O
AAO,QAAS,cAAhB,C;UACI,cAAc,QAAS,W;UACvB,IA7Vc,SA6VV,CAAU,OAAV,CAAJ,C;YAAwB,oBAAO,
O;YAAP,sB;;;QAE5B,oBAAO,I;;;MA/VP,wB;K;IAGJ,6B;MAOQ,kBADE,SACF,Q;QAAW,OAAY,SAAL,SAAK
,C;;QAEnB,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UACI,MAAM,2BAAuB,sBAAvB,C;QACV,OAAO,QA
AS,O;;K;IAK5B,6B;MAMI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,gBAAvB,C;MACV,OAAO,sBAAK,CAAL,
C;K;mFAGX,yB;MAAA,iE;MAAA,uC;QAKoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IA
AI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,wDAAvB,C;O;KANV,C;oGASA,yB;M
AAA,iE;MAAA,uC;QAS8C,IAAnC,I;QAAAA,+B;;UAYS,U;UAAAA,6B;UAAhB,OAAgB,gBAAhB,C;YAAgB,2B;
YACZ,aAbwB,SAaX,CAAU,OAAV,C;YACb,IAAI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;;UAGR,8BAAO,I;;;QAl
BA,kC;QAAA,iB;UAAmC,MAAM,gCAAuB,mEAAvB,C;;QAAhD,OAAO,I;O;KATX,C;gHAYA,gC;MASoB,Q;
MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,aAAa,UAAU,OAAV,C;QACb,IAAI,cAAJ,C;UACI,O
AAO,M;;;MAGf,OAAO,I;K;IAGX,mC;MAKQ,kBADE,SACF,Q;QACI,IAAI,mBAAJ,C;UACI,OAAO,I;;UAEP,O
AAO,sBAAK,CAAL,C;;QAGX,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UACI,OAAO,I;QACX,OAAO,QA
AS,O;;K;IAK5B,mC;MAII,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,CAAL,C;K;+FAGpC,gC;MAIoB,Q;MA
AA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACr
D,OAAO,I;K;0FAGX,yB;MAAA,8D;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,sBA
AI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KALjE,C;IAQA,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,2BA
A3B,GAAsC,sBAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,uC;MAMiB,Q;MAFb,IAAI,8BAAJ,C;QAAkB,OAAO,SA
AK,eAAQ,OAAR,C;MAC9B,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,mBAAmB,KAAnB,C
;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UACI,OAAO,K;QACX,qB;;MAEJ,OAAO,E;K;IAGX,uC;MAKI,OAAO,w
BAAQ,OAAR,C;K;gGAGX,yB;MAAA,wE;MAAA,uC;QAKiB,Q;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAA
b,C;UAAa,sB;UACT,mBAAmB,KAAnB,C;UACA,IAAI,UAAU,IAAV,CAAJ,C;YACI,OAAO,K;UACX,qB;;QAE
J,OAAO,E;O;KAXX,C;gGAcA,gC;MAKiB,Q;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAC
T,IAAI,UAAU,IAAV,CAAJ,C;UACI,OAAO,K;QACX,qB;;MAEJ,OAAO,E;K;8FAGX,yB;MAAA,wE;MAAA,uC
;QAMiB,Q;QAFb,gBAAgB,E;QAChB,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,mBAAmB,K
AAnB,C;UACA,IAAI,UAAU,IAAV,CAAJ,C;YACI,YAAY,K;UAChB,qB;;QAEJ,OAAO,S;O;KAZX,C;8FAeA,g
C;MAII,eAAe,SAAK,sBAAa,cAAb,C;MACpB,OAAO,QAAS,cAAhB,C;QACI,IAAI,UAAU,QAAS,WAAnB,CA
AJ,C;UACI,OAAO,QAAS,Y;;;MAGxB,OAAO,E;K;IAGX,4B;MASQ,kBADE,SACF,Q;QAAW,OAAY,QAAL,S
AAK,C;;QAEnB,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UACI,MAAM,2BAAuB,sBAAvB,C;QACV,WAA
W,QAAS,O;QACpB,OAAO,QAAS,UAAhB,C;UACI,OAAO,QAAS,O;QACpB,OAAO,I;;K;IAKnB,4B;MAQI,IA
AI,mBAAJ,C;QACI,MAAM,2BAAuB,gBAAvB,C;MACV,OAAO,sBAAK,2BAAL,C;K;iFAGX,yB;MAAA,iE;M

AAA,gB;MAAA,8B;MAAA,uC;QAUoB,UAQT,M;QAVP,WAAe,I;QACf,YAAY,K;QACI,2B;QAAhB,OAAgB,c
AAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,OAAO,O;YACP,QAAQ,I;;;QAGhB,IAAI,CAA
C,KAAL,C;UAAY,MAAM,gCAAuB,wDAAvB,C;QAElB,OAAO,2E;O;KAlBX,C;iFAqBA,yB;MAAA,iE;MAAA,
uC;QAQI,eAAe,SAAK,sBAAa,cAAb,C;QACpB,OAAO,QAAS,cAAhB,C;UACI,cAAc,QAAS,W;UACvB,IAAI,U
AAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,kDAAvB,C;O;KAbV,C;IAgBA,2C;MAOiB,
Q;MAHb,IAAI,8BAAJ,C;QAAkB,OAAO,SAAK,mBAAY,OAAZ,C;MAC9B,gBAAgB,E;MAChB,YAAY,C;MA
CC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,mBAAmB,KAAnB,C;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UAC
I,YAAY,K;QAChB,qB;;MAEJ,OAAO,S;K;IAGX,2C;MAKI,OAAO,4BAAY,OAAZ,C;K;IAGX,kC;MAOQ,kBAD
E,SACF,Q;QAAW,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,iBAAO,CAAP,IAAL,C;;QAEvC,eAAe,oB;QA
Cf,IAAI,CAAC,QAAS,UAAd,C;UACI,OAAO,I;QACX,WAAW,QAAS,O;QACpB,OAAO,QAAS,UAAhB,C;UA
CI,OAAO,QAAS,O;QACpB,OAAO,I;;K;IAKnB,kC;MAMI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,iBAA
O,CAAP,IAAL,C;K;6FAGpC,gC;MAOoB,Q;MADhB,WAAe,I;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,y
B;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,OAAO,O;;;MAGf,OAAO,I;K;6FAGX,gC;MAMI,eAAe,SAAK,sBA
Aa,cAAb,C;MACpB,OAAO,QAAS,cAAhB,C;QACI,cAAc,QAAS,W;QACvB,IAAI,UAAU,OAAV,CAAJ,C;UAA
wB,OAAO,O;;MAEnC,OAAO,I;K;qFAGX,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C
;O;KARX,C;IAWA,sC;MAOI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,sBAAvB,C;MACV,OAAO,qBAAU,MAA
O,iBAAQ,cAAR,CAAjB,C;K;iGAGX,yB;MAAA,mC;MAAA,4D;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;K
APX,C;IAUA,4C;MAMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAO,qBAAU,MAAO,iBAAQ,cAAR,CAAjB,
C;K;IAGX,8B;MAKQ,kBADE,SACF,Q;QAAW,OAAY,UAAL,SAAK,C;;QAEnB,eAAe,oB;QACf,IAAI,CAAC,Q
AAS,UAAd,C;UACI,MAAM,2BAAuB,sBAAvB,C;QACV,aAAa,QAAS,O;QACtB,IAAI,QAAS,UAAb,C;UACI,
MAAM,gCAAyB,uCAAzB,C;QACV,OAAO,M;;K;IAKnB,8B;MAIiB,IAAN,I;MAAA,QAAM,cAAN,C;aACH,C;
UAAK,MAAM,2BAAuB,gBAAvB,C;aACX,C;UAAK,6BAAK,CAAL,C;UAAL,K;;UACQ,MAAM,gCAAyB,iCA
AzB,C;;MAHlB,W;K;qFAOJ,yB;MAAA,kF;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAMoB,UAST,M;QA
XP,aAAiB,I;QACjB,YAAY,K;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CA
AJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,qDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI
,CAAC,KAAL,C;UAAY,MAAM,gCAAuB,wDAAvB,C;QAElB,OAAO,6E;O;KAfX,C;IAkBA,oC;MAKQ,kBADE
,SACF,Q;QAAW,OAAW,mBAAQ,CAAZ,GAAe,sBAAK,CAAL,CAAf,GAA4B,I;;QAE1C,eAAe,oB;QACf,IAAI,
CAAC,QAAS,UAAd,C;UACI,OAAO,I;QACX,aAAa,QAAS,O;QACtB,IAAI,QAAS,UAAb,C;UACI,OAAO,I;QA
CX,OAAO,M;;K;IAKnB,oC;MAII,OAAW,mBAAQ,CAAZ,GAAe,sBAAK,CAAL,CAAf,GAA4B,I;K;iGAGvC,g
C;MAMoB,Q;MAFhB,aAAiB,I;MACjB,YAAY,K;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI
,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAAW,OAAO,I;UAClB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,
CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;IAGX,8B;MAoBsC,UAGT,MAHS,EAarB,M;MN/pBb,IA
AI,EMsoBI,KAAK,CNtoBT,CAAJ,C;QACI,cMqoBc,sD;QNpoBd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MMqoBV
,IAAI,MAAK,CAAT,C;QAAY,OAAO,mB;MACnB,Q;MACA,IAAI,oCAAJ,C;QACI,iBAAiB,iBAAO,CAAP,I;Q
ACjB,IAAI,cAAc,CAAlB,C;UACI,OAAO,W;QACX,IAAI,eAAc,CAAlB,C;UACI,OAAO,OAAO,kBAAP,C;QAC
X,OAAO,iBAAa,UAAb,C;QACP,IAAI,8BAAJ,C;UACI,IAAI,sCAAJ,C;YAC0B,qB;YAAtB,iBAAc,CAAd,wB;cA
CI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;;YAEI,wCAAa,CAAb,C;YAAb,OAAa,gBAAb,C;cAAa,wB;cACT,IAA
K,WAAI,IAAJ,C;;;UAEb,OAAO,I;;;QAIX,OAAO,gB;;MAEX,YAAY,C;MACC,6B;MAAb,OAAa,gBAAb,C;QA
Aa,0B;QACT,IAAI,SAAS,CAAb,C;UAAgB,IAAK,WAAI,MAAJ,C;;UAAe,qB;;MAExC,OAAY,qBAAL,IAAK,C
;K;IAGhB,kC;MNrqBI,IAAI,EM6qBI,KAAK,CN7qBT,CAAJ,C;QACI,cM4qBc,sD;QN3qBd,MAAM,gCAAyB,O
AAQ,WAAjC,C;;MM4qBV,OAAO,kBAAgB,gBAAV,iBAAO,CAAP,IAAU,EAAc,CAAd,CAAhB,C;K;kGAGX,y
B;MAAA,4C;MAAA,qD;MAAA,uC;QAMI,IAAI,CAAC,mBAAL,C;UACI,eAAe,+BAAa,cAAb,C;UACf,OAAO,
QAAS,cAAhB,C;YACI,IAAI,CAAC,UAAU,QAAS,WAAnB,CAAL,C;cACI,OAAO,gBAAK,QAAS,YAAT,GAA
uB,CAAvB,IAAL,C;;;;QAInB,OAAO,W;O;KAdX,C;0FAiBA,yB;MAAA,+D;MAAA,uC;QAQiB,Q;QAFb,eAAe,
K;QACf,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ,
C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAEnB,OAAO,I;O;
KAfX,C;oFAkBA,yB;MAAA,+D;MAAA,uC;QAMW,kBAAS,gB;QA2FA,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,
C;UAAgB,yB;UAAM,IA3FU,SA2FN,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAI,OAAJ,C;;QA3F1D,OA4F

O,W;O;KAlGX,C;kGASA,yB;MAAA,+D;MAikCA,wE;MAjkCA,uC;QAQW,kBAAgB,gB;QAgkCV,gB;QADb,Y
AAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UApjCT,IAZmC,SAY/B,CAojCkB,oBAAmB,cAAnB,EAAm
B,sBAAnB,UApjClB,EAojC+C,IApjC/C,CAAJ,C;YAA2C,sBAojCQ,IApjCR,C;;QAZ/C,OAcO,W;O;KAtBX,C;sG
AWA,yB;MAsjCA,wE;MAtjCA,oD;QA6jCiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;U
ApjCT,IAAI,UAojCkB,oBAAmB,cAAnB,EAAmB,sBAAnB,UApjClB,EAojC+C,IApjC/C,CAAJ,C;YAA2C,sBAo
jCQ,IApjCR,C;;QAE/C,OAAO,W;O;KAXX,C;wGAcA,yB;MAAA,+D;MAAA,sC;QAMW,kBAAmB,gB;QASV,Q
;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,YAAJ,C;YAAkB,WAAY,WAAI,OAAJ,C;;QAT
pD,OAUO,W;O;KAhBX,C;4GASA,4C;MAMoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,I
AAI,YAAJ,C;UAAkB,WAAY,WAAI,OAAJ,C;;MACpD,OAAO,W;K;0FAGX,yB;MAAA,+D;MAAA,uC;QAMW
,kBAAY,gB;QA4BH,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CA5BS,SA4BR,CAAU,
OAAV,CAAL,C;YAAyB,WAAY,WAAI,OAAJ,C;;QA5B3D,OA6BO,W;O;KAnCX,C;IASA,oC;MAMI,OAAO,6
BAAgB,gBAAhB,C;K;IAGX,mD;MAMoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,e
AAJ,C;UAAqB,WAAY,WAAI,OAAJ,C;;MACvD,OAAO,W;K;8FAGX,6C;MAMoB,Q;MAAA,2B;MAAhB,OAA
gB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAC3
D,OAAO,W;K;wFAGX,6C;MAMoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,
OAAV,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;IAGX,sC;MAII,IAAI,OAAQ,UAAZ,C;
QAAuB,OhB7wBe,W;;MgB8wBtC,OAA6D,SAAtD,SAAK,iBAAQ,OAAQ,MAAhB,EAAuB,OAAQ,aAAR,GAA
uB,CAAvB,IAAvB,CAAiD,C;K;IAGjE,sC;MAOkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MA
CnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACtB,WAAW,iBAAa,IAAb,C;MACG,yB;MAAd,OAAc,cAAd,C;
QAAc,uB;QACV,IAAK,WAAI,sBAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,8B;MAgBiB,Q;MN91Bb,IAAI,
EMs1BI,KAAK,CNt1BT,CAAJ,C;QACI,cMq1Bc,sD;QNp1Bd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MMq1BV,IA
AI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,oCAAJ,C;QACI,IAAI,KAAK,cAAT,C;UAAe,OAAO,mB;Q
ACtB,IAAI,MAAK,CAAT,C;UAAY,OAAO,OAAO,mBAAP,C;;MAEvB,YAAY,C;MACZ,WAAW,iBAAa,CAAb
,C;MACE,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UAC
I,K;;MAER,OAAY,qBAAL,IAAK,C;K;IAGhB,kC;MAeqC,IAGhB,I;MNx3BjB,IAAI,EM82BI,KAAK,CN92BT,C
AAJ,C;QACI,cM62Bc,sD;QN52Bd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MM62BV,IAAI,MAAK,CAAT,C;QAA
Y,OAAO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;
QAAY,OAAO,OAAO,kBAAP,C;MACnB,WAAW,iBAAa,CAAb,C;MACX,IAAI,sCAAJ,C;QACI,iBAAc,OAAO,
CAAP,IAAd,UAA6B,IAA7B,U;UACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;;QAEI,sCAAa,OAAO,CAAP,IAAb,
C;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAK,WAAI,IAAJ,C;;;MAEb,OAAO,I;K;kGAGX,yB;MAAA,qD;MA
AA,gE;MAAA,gD;MAAA,uC;QAMI,IAAI,mBAAJ,C;UACI,OAAO,W;QACX,eAAe,+BAAa,cAAb,C;QACf,OA
AO,QAAS,cAAhB,C;UACI,IAAI,CAAC,UAAU,QAAS,WAAnB,CAAL,C;YACI,QAAS,O;YACT,mBAAmB,iBA
AO,QAAS,YAAhB,I;YACnB,IAAI,iBAAgB,CAApB,C;cAAuB,OAAO,W;YACI,kBAA3B,eAAa,YAAb,C;YACH
,OAAgB,kBAAhB,C;cACI,sBAAa,eAAb,C;YAFR,OH51BD,W;;;QGk2BP,OAAO,iB;O;KApBX,C;0FAuBA,yB;
MAAA,+D;MAAA,uC;QAOiB,Q;QADb,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,CA
AC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KAZX,C;IAoBA,+B;MAII,I
AAI,wCAAsB,kBAAQ,CAAlC,C;QAAqC,OAAO,mB;MAC5C,WAAW,0B;MACN,WAAL,IAAK,C;MACL,OAA
O,I;K;IAGX,uC;MAOI,aAAU,2BAAV,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;Q
ACf,sBAAK,CAAL,EAAU,SAAK,aAAI,CAAJ,EAAO,sBAAK,CAAL,CAAP,CAAf,C;;K;oFAIR,yB;MAAA,oD;
MJr4BA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,c
AAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MI83Bf,sC;QAMI,IAAI,iBA
AO,CAAX,C;UAAc,oBJp4Bd,eAAW,iBIo4BsB,QJp4BtB,CAAX,CIo4Bc,C;;O;KANlB,C;wGASA,yB;MAAA,oD;
MJ33BA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,c
AAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,C;MIo3Bf,sC;QAMI,IAAI,iBAAO
,CAAX,C;UAAc,oBJ13Bd,eAAW,2BI03BgC,QJ13BhC,CAAX,CI03Bc,C;;O;KANlB,C;IASA,sC;MAMI,sBAAS,c
AAT,C;K;IAGJ,6B;MASgB,Q;MAHZ,IAAI,oCAAJ,C;QACI,IAAI,kBAAQ,CAAZ,C;UAAe,OAAY,SAAL,SAAK
,C;QAEwB,kBAA3C,sBC7+BsD,sBD6+BtD,uB;QAAmD,mB;QAA3D,OAAoE,OHp7BjE,WGo7BiE,C;;MAEjD,k
BAAhB,0B;MAAwB,oB;MAA/B,OHt7BO,W;K;wFGy7BX,yB;MAAA,wD;MJ96BA,sC;MAAA,oC;MAAA,uBA

Oe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA
2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MIu6Bf,sC;QAQI,OAAO,sBJ/6BP,eAAW,iBI+6BiB,QJ/6BjB,CA
AX,CI+6BO,C;O;KARX,C;4GAWA,yB;MAAA,wD;MJt6BA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe
,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3
B,C;W;S;OA+EI,C;MI+5Bf,sC;QAMI,OAAO,sBJr6BP,eAAW,2BIq6B2B,QJr6B3B,CAAX,CIq6BO,C;O;KANX,
C;IASA,uC;MAMI,OAAO,wBAAW,cAAX,C;K;IAGX,6C;MASe,Q;MAHX,IAAI,oCAAJ,C;QACG,IAAI,kBAAQ
,CAAZ,C;UAAe,OAAY,SAAL,SAAK,C;QAEe,kBAAlC,sBCxhCuD,sBDwhCvD,uB;QAA0C,iC;QAAlD,OAAyE,
OH/9BrE,WG+9BqE,C;;MAErD,kBAAhB,0B;MAAwB,mC;MAA/B,OHj+BO,W;K;IGo+BX,qC;MAMoB,UACL,
M;MAHX,aAAa,oBAAa,cAAb,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OA
AO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OAAO,M;K;IAGX,kC;MAMoB,UACL,M;MAHX,aAAa,cAAU,cA
AV,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP,Y
AAkB,O;;MACtB,OAAO,M;K;IAGX,kC;MAMoB,UACL,M;MAHX,aAAa,iBAAU,cAAV,C;MACb,YAAY,C;M
ACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,oC;QACZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OAAO,
M;K;IAGX,oC;MAMoB,UACL,M;MAHX,aAAa,iBAAY,cAAZ,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAgB,c
AAhB,C;QAAgB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OAAO,M;K;IAGX,mC;MAMoB,U
ACL,M;MAHX,aAAa,iBAAW,cAAX,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QA
CZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OAAO,M;K;IAGX,iC;MAMoB,UACL,M;MAHX,aAAa,eA
AS,cAAT,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,cAAP,EAAO,sB
AAP,YAAkB,O;;MACtB,OAAO,M;K;IAGX,kC;MAMoB,UACL,M;MAHX,aAAa,iBAAU,cAAV,C;MACb,YAA
Y,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,
OAAO,M;K;IAGX,mC;MAMoB,UACL,M;MAHX,aAAa,eAAW,cAAX,C;MACb,YAAY,C;MACI,2B;MAAhB,O
AAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;MACtB,OAAO,M;K;0FAGX,yB;M
AAA,kF;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,uC;QAWI,eAAwD,cAAzC,YAAY,mCAAwB,EAAxB,CAAZ,
CAAyC,EAAc,EAAd,C;QACjD,kBAAY,mBAAoB,QAApB,C;QAyEH,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;
UAAgB,yB;UACZ,WA1E8C,SA0E/B,CAAU,OAAV,C;UflkBnB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C
;;QewfA,OA4EO,W;O;KAxFX,C;+FAeA,yB;MAAA,kF;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,yC;QAWI,eA
AwD,cAAzC,YAAY,mCAAwB,EAAxB,CAAZ,CAAyC,EAAc,EAAd,C;QACjD,kBAAc,mBAAoB,QAApB,C;QA
2BL,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAY,aA5BoC,WA4BhC,CAAY,OAAZ,CAA
J,EAA0B,OAA1B,C;;QA5BhB,OA8BO,W;O;KA1CX,C;+FAeA,yB;MAAA,kF;MAAA,0D;MAAA,yD;MAAA,uE
;MAAA,yD;QAUI,eAAwD,cAAzC,YAAY,mCAAwB,EAAxB,CAAZ,CAAyC,EAAc,EAAd,C;QACjD,kBAAc,m
BAAoB,QAApB,C;QA6BL,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAY,aA9BoC,WA8B
hC,CAAY,OAAZ,CAAJ,EA9BiD,cA8BvB,CAAe,OAAf,CAA1B,C;;QA9BhB,OAgCO,W;O;KA3CX,C;mGAcA,+
C;MAUoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,WAAY,aAAI,YAAY,OAAZ,CAAJ,EA
A0B,OAA1B,C;;MAEhB,OAAO,W;K;mGAGX,+D;MAUoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,y
B;QACZ,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,eAAe,OAAf,CAA1B,C;;MAEhB,OAAO,W;K;8FAGX,6C;
MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,WAAe,UAAU,OAAV,C;QflkBnB,wBAAI,
IAAK,MAAT,EAAgB,IAAK,OAArB,C;;MeokBA,OAAO,W;K;kGAGX,yB;MAAA,kF;MAAA,0D;MAAA,yD;M
AAA,uE;MAAA,2C;QAYI,aAAa,mBAA6D,cAAzC,YAAY,mCAAwB,EAAxB,CAAZ,CAAyC,EAAc,EAAd,CAA
7D,C;QAcG,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAbO,MAcP,aAAI,OAAJ,EAde,aAcF,CAAc,
OAAd,CAAb,C;;QAdhB,OAAuB,M;O;KAb3B,C;sGAgBA,iD;MAUoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;
QAAgB,yB;QACZ,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;IAGX,gD;MAIiB,Q;
MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAY,WAAI,IAAJ,C;;MAEhB,OAAO,W;K;IAGX,gC;MAI
I,OAAO,0BAAa,eAAW,YAAY,mCAAwB,EAAxB,CAAZ,CAAX,CAAb,C;K;IAGX,6B;MAKqB,IAAN,I;MADX,
IAAI,oCAAJ,C;QACW,QAAM,cAAN,C;eACH,C;YAAK,kB;YAAL,K;eACA,C;YAAK,cAAW,8BAAJ,GAAAkB,s
BAAI,CAAJ,CAAlB,GAA8B,oBAAW,OAAhD,C;YAAL,K;;YACa,uBAAL,SAAK,C;YAHV,K;;QAAP,W;;MAM
J,OAA4B,qBAAhB,gBAAL,SAAK,CAAgB,C;K;IAGhC,oC;MAII,IAAI,oCAAJ,C;QACI,OAAY,gBAAL,SAAK,
C;MAChB,OAAO,0BAAa,gBAAb,C;K;IAGX,oC;MAII,OAAO,iBAAU,SAAV,C;K;IAGX,4B;MAOqB,IAAN,I;M
ADX,IAAI,oCAAJ,C;QACW,QAAM,cAAN,C;eACH,C;YAAK,iB;YAAL,K;eACA,C;YAAK,aAAU,8BAAJ,GAA

kB,sBAAK,CAAL,CAAlB,GAA+B,oBAAW,OAAhD,C;YAAL,K;;YACQ,iCAAa,qBAAiB,YAAY,cAAZ,CAAjB,
CAAb,C;YAHL,K;;QAAP,W;;MAMJ,OAAwC,oBAAjC,0BAAa,sBAAb,CAAiC,C;K;sFAG5C,yB;MAAA,+D;M
AwFA,gD;MAxFA,uC;QAMW,kBAAU,gB;QAsFD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,
WAvF6B,SAuFlB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAxFhB,OA0FO,W;O;KAhGX,C;uF
ASA,yB;MAAA,+D;MA0FA,gD;MA1FA,uC;QAUW,kBAAU,gB;QAwFD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB
,C;UAAgB,yB;UACZ,WAzF6B,SAyFlB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA1FhB,OA4F
O,W;O;KAtGX,C;oGAaA,yB;MAAA,+D;MA8BA,wE;MAAA,gD;MA9BA,uC;QAYW,kBAAiB,gB;QA6BR,gB;
QADhB,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WA9BoC,SA8BzB,CAAU,oBAAmB,
cAAnB,EAAmB,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA/BhB,OAiCO,W;
O;KA7CX,C;oGAeA,yB;MAAA,+D;MAiCA,wE;MAAA,gD;MAjCA,uC;QAYW,kBAAiB,gB;QAgCR,gB;QADh
B,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAjCoC,SAiCzB,CAAU,oBAAmB,cAAnB
,EAAmB,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAlChB,OAoCO,W;O;KA
hDX,C;wGAeA,yB;MAAA,wE;MAAA,gD;MAAA,oD;QAWoB,UAC4B,M;QAF5C,YAAY,C;QACI,2B;QAAhB,
OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,oBAAmB,cAAnB,EAAmB,sBAAnB,UAAV,EAAuC,OAA
vC,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAfX,C;yGAkBA,yB;MAAA,wE;MAAA,gD;
MAAA,oD;QAWoB,UAC4B,M;QAF5C,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAA
W,UAAU,oBAAmB,cAAnB,EAAmB,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;
;QAEhB,OAAO,W;O;KAfX,C;0FAkBA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB
,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KA
RX,C;2FAWA,yB;MAAA,gD;MAAA,oD;QAQoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,
WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAZX,C;uFAeA,yB;MA
AA,wE;MAyBA,+D;MAzBA,yC;QASW,kBAAU,oB;QAyBD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,y
B;UACZ,UA1BiD,WA0BvC,CAAY,OAAZ,C;UfrnCP,U;UADP,YeunCe,WfvnCH,WeunCwB,GfvnCxB,C;UACL,
IAAI,aAAJ,C;YACH,aeqnCuC,gB;YAA5B,WfpnCX,aeonCgC,GfpnChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;Uei
nCA,iB;UACA,IAAK,WAAI,OAAJ,C;;QA5BT,OA8BO,W;O;KAvCX,C;uFAYA,yB;MAAA,wE;MA8BA,+D;MA
9BA,yD;QAUW,kBAAU,oB;QA8BD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UA/BiD,WA+
BvC,CAAY,OAAZ,C;UfvoCP,U;UADP,YeyoCe,WfzoCH,WeyoCwB,GfzoCxB,C;UACL,IAAI,aAAJ,C;YACH,ae
uoCuC,gB;YAA5B,WftoCX,aesoCgC,GftoChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UemoCA,iB;UACA,IAAK,W
AjCyD,cAiCrD,CAAe,OAAf,CAAJ,C;;QAjCT,OAmCO,W;O;KA7CX,C;0FAaA,yB;MAAA,+D;MAAA,sD;QASo
B,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;UfrnCP,U;UADP,YeunCe
,WfvnCH,WeunCwB,GfvnCxB,C;UACL,IAAI,aAAJ,C;YACH,aeqnCuC,gB;YAA5B,WfpnCX,aeonCgC,GfpnChC
,EAAS,MAAT,C;YACA,e;;YAEA,c;;UeinCA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAdX,C;2F
AiBA,yB;MAAA,+D;MAAA,sE;QAUoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YA
AY,OAAZ,C;UfvoCP,U;UADP,YeyoCe,WfzoCH,WeyoCwB,GfzoCxB,C;UACL,IAAI,aAAJ,C;YACH,aeuoCuC,g
B;YAA5B,WftoCX,aesoCgC,GftoChC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UemoCA,iB;UACA,IAAK,WAAI,eA
Ae,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAfX,C;4FAkBA,yB;MAAA,kC;MAAA,4C;MAAA,wE;QAQW,sC;QAA
A,8C;O;MARX,oDASQ,Y;QAA6C,OAAA,oBAAgB,W;O;MATrE,iDAUQ,mB;QAAoC,gCAAY,OAAZ,C;O;MA
V5C,gF;MAAA,yC;QAQI,2D;O;KARJ,C;8EAcA,yB;MAAA,kF;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,m
CAAwB,EAAxB,CAAb,C;QAuEA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAY,WAxEwC,SAw
EpC,CAAU,IAAV,CAAJ,C;;QAxEhB,OAyEO,W;O;KAhFX,C;4FAUA,yB;MAAA,kF;MAAA,gE;MA+BA,wE;M
A/BA,uC;QAOW,kBAAa,eAAa,mCAAwB,EAAxB,CAAb,C;QAgCP,gB;QADb,YAAY,C;QACC,2B;QAAb,OAA
a,cAAb,C;UAAa,sB;UACT,WAAY,WAjC+C,SAiC3C,CAAU,oBAAmB,cAAnB,EAAmB,sBAAnB,UAAV,EAAu
C,IAAvC,CAAJ,C;;QAjChB,OAkCO,W;O;KAzCX,C;0GAUA,yB;MAAA,+D;MAwSA,wE;MAxSA,uC;QAOW,k
BAAoB,gB;QAwSd,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA9RsB,U;UAAA,cAVQ,S
AUR,CA8RT,oBAAmB,cAAnB,EAAmB,sBAAnB,UA9RS,EA8RoB,IA9RpB,W;YAA6C,6B;;;QAVhF,OAWO,W
;O;KAlBX,C;8GAUA,yB;MA8RA,wE;MA9RA,oD;QAqSiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,
C;UAAa,sB;UA9RsB,U;UAAA,wBA8RT,oBAAmB,cAAnB,EAAmB,sBAAnB,UA9RS,EA8RoB,IA9RpB,W;YA
A6C,6B;;;QAChF,OAAO,W;O;KARX,C;+FAWA,yB;MAAA,wE;MAAA,oD;QAQiB,UACoC,M;QAFjD,YAAY,

C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAY,WAAI,UAAU,oBAAmB,cAAnB,EAAmB,sBAAnB,
UAAV,EAAuC,IAAvC,CAAJ,C;;QAChB,OAAO,W;O;KAVX,C;4FAaA,yB;MAAA,+D;MAAA,uC;QAOW,kBA
Aa,gB;QA4PJ,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UApPK,U;UAAA,cARe,SAQf,CAoPQ,OApP
R,W;YAAsC,6B;;;QAR3D,OASO,W;O;KAhBX,C;gGAUA,yB;MAAA,oD;QAyPoB,Q;QAAA,2B;QAAhB,OAAg
B,cAAhB,C;UAAgB,yB;UApPK,U;UAAA,wBAoPQ,OApPR,W;YAAsC,6B;;;QAC3D,OAAO,W;O;KANX,C;kF
ASA,6C;MAKiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAY,WAAI,UAAU,IAAV,CAAJ,C;;M
AChB,OAAO,W;K;IAQiB,4C;MAAA,mB;QAAE,gC;O;K;IAL9B,gC;MAKI,OAAO,qBAAiB,6BAAjB,C;K;IAGX
,+B;MASI,OAA2B,SAAf,eAAL,SAAK,CAAe,C;K;4FAG/B,yB;MAAA,2D;MAAA,+D;MAAA,sC;QAYc,Q;QAF
V,UAAU,c;QACV,WAAW,gB;QACD,2B;QAAV,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,SAAS,CAAT,C;U
ACV,IAAI,GAAI,WAAI,GAAJ,CAAR,C;YACI,IAAK,WAAI,CAAJ,C;;QAEb,OAAO,I;O;KAjBX,C;IAoBA,uC;
MAQI,UAAe,eAAL,SAAK,C;MACX,YAAJ,GAAI,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,sC;MAMI,UAAe,
eAAL,SAAK,C;MACX,YAAJ,GAAI,EAAU,KAAV,C;MACJ,OAAO,G;K;IAGX,mC;MAMiB,IAAN,I;MACH,kB
ADS,SACT,c;QAAoB,4BAAc,SAAd,C;;QACZ,iCAAa,sBAAb,C;MAFZ,W;K;IAMJ,mC;MAUI,UAAe,eAAL,SA
AK,C;MACX,OAAJ,GAAI,EAAO,KAAP,C;MACJ,OAAO,G;K;8EAGX,yB;MAAA,gD;MAAA,uC;QAWoB,Q;Q
ADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,OAAO,I;QAC5B,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,
IAAI,CAAC,UAAU,OAAV,CAAL,C;YAAyB,OAAO,K;;QACtD,OAAO,I;O;KAZX,C;IAeA,2B;MAMI,IAAI,oC
AAJ,C;QAAwB,OAAO,CAAC,mB;MAChC,OAAO,oBAAW,U;K;+EAGtB,yB;MAAA,gD;MAAA,uC;QAOoB,Q;
QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,OAAO,K;QAC5B,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAA
M,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,I;;QACrD,OAAO,K;O;KARX,C;IAWA,6B;MAMoB,Q;MAFhB
,IAAI,oCAAJ,C;QAAwB,OAAO,c;MAC/B,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,o
BAAmB,qBAAnB,EAAmB,KAAnB,E;;MACtB,OAAO,K;K;mFAGX,qB;MAKI,OAAO,c;K;mFAGX,yB;MAAA,
gD;MAAA,wE;MAAA,uC;QAMoB,Q;QAFhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,OAAO,C;QAC5C,YAAY,C;
QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,oBAAmB,qBA
AnB,EAAmB,KAAnB,E;;QAC9C,OAAO,K;O;KAPX,C;gFAUA,yC;MAUoB,Q;MADhB,kBAAkB,O;MACF,2B;
MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;
8FAGX,yB;MAAA,wE;MAAA,gD;QAYoB,UAAiD,M;QAFjE,YAAY,C;QACZ,kBAAkB,O;QACF,2B;QAAhB,O
AAgB,cAAhB,C;UAAgB,yB;UAAM,cAAc,UAAU,oBAAmB,cAAnB,EAAmB,sBAAnB,UAAV,EAAuC,WAAvC
,EAAoD,OAApD,C;;QACpC,OAAO,W;O;KAbX,C;0FAgBA,yC;MASI,kBAAkB,O;MAClB,IAAI,CAAC,mBAA
L,C;QACI,eAAe,+BAAa,cAAb,C;QACf,OAAO,QAAS,cAAhB,C;UACI,cAAc,UAAU,QAAS,WAAnB,EAA+B,W
AA/B,C;;;MAGtB,OAAO,W;K;wGAGX,yC;MAUI,kBAAkB,O;MAClB,IAAI,CAAC,mBAAL,C;QACI,eAAe,+B
AAa,cAAb,C;QACf,OAAO,QAAS,cAAhB,C;UACI,YAAY,QAAS,gB;UACrB,cAAc,UAAU,KAAV,EAAiB,QAA
S,WAA1B,EAAsC,WAAtC,C;;;MAGtB,OAAO,W;K;sFAGX,6B;MAKoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB
,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;K;oGAG1B,yB;MAAA,wE;MAAA,oC;QAOiB,UAAgC,M;QAD7C,YA
AY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAAM,OAAO,oBAAmB,cAAnB,EAAmB,sBAAnB,UAAP,EA
AoC,IAApC,C;;O;KAPvB,C;IAUA,0B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,MAAM,6B
;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MF5wDG,MAAO,K
E4wDE,GF5wDF,EE4wDO,CF5wDP,C;;ME8wDd,OAAO,G;K;IAGX,2B;MAWI,eAAe,oB;MACf,IAAI,CAAC,Q
AAS,UAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,Q
AAS,O;QACjB,MF5yDG,MAAO,KE4yDE,GF5yDF,EE4yDO,CF5yDP,C;;ME8yDd,OAAO,G;K;IAGX,2B;MASI,
eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QA
AS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,
G;K;kFAGX,yB;MAAA,sE;MAAA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MAAM,6B;
QAC/B,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAd,C;UAAyB,OAAO,O;QAChC,eAAe,SAAS,OAAT,C;;
UAEX,QAAQ,QAAS,O;UACjB,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;
YACV,WAAW,C;;;QAED,QAAT,QAAS,W;QAClB,OAAO,O;O;KAxBX,C;8FA2BA,+B;MAOI,eAAe,oB;MACf,
IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChChC,cAAc,QAAS,O;MACvB,IAAI,CAAC,QAAS,UAAd,C;Q
AAyB,OAAO,O;MAChChC,eAAe,SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CAAT,C;QACR,
IAAI,2BAAW,CAAX,KAAJ,C;UACI,UAAU,C;UACV,WAAW,C;;;MAED,QAAT,QAAS,W;MAClB,OAAO,O;K

;mFAGX,yB;MAAA,sE;MFn3DA,iB;MEm3DA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,
MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,O
AAlB,C;UACR,WF73DG,MAAO,KE63DO,QF73DP,EE63DiB,CF73DjB,C;;QE+3Dd,OAAO,Q;O;KApBX,C;mF
AuBA,yB;MAAA,sE;MFr5DA,iB;MEq5DA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MA
AM,6B;QAC/B,eAAe,SAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAl
B,C;UACR,WF/5DG,MAAO,KE+5DO,QF/5DP,EE+5DiB,CF/5DjB,C;;QEi6Dd,OAAO,Q;O;KApBX,C;mFAuBA,
yB;MAAA,sE;MAAA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe
,SAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAlB,C;UACR,IAAI,2B
AAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;+FAuBA,yB;MFx7DA,iB;MEw7DA,sC
;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAlB,C;Q
ACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAlB,C;UACR,WFh8DG,MAAO,KEg8DO,QFh8DP
,EEg8DiB,CFh8DjB,C;;QEk8Dd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MFx9DA,iB;MEw9DA,sC;QAWI,eAAe,oB;
QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAlB,C;QACf,OAAO,QAA
S,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAlB,C;UACR,WFh+DG,MAAO,KEg+DO,QFh+DP,EEg+DiB,CFh+
DjB,C;;QEk+Dd,OAAO,Q;O;KAlBX,C;+FAqBA,+B;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAy
B,OAAO,I;MAChC,eAAe,SAAS,QAAS,OAAlB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,
OAAlB,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;0FAGX,yB;MAAA,sE;
MAAA,kD;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS
,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAlB,C;UACR,IAAI,UAAW,SAAQ,Q
AAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;sGAuBA,2C;
MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,eAAe,SAAS,QAAS,OAAlB,C;M
ACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAlB,C;QACR,IAAI,UAAW,SAAQ,QAAR,EAAkB
,CAAlB,CAAX,GAAkC,CAAtC,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;IAGX,gC;MAOI,eAAe,oB;MACf,IA
AI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,
QAAQ,QAAS,O;QACjB,MFviEG,MAAO,KEuiEE,GFviEF,EEuiEO,CFviEP,C;;MEyiEd,OAAO,G;K;IAGX,iC;M
AOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO,
QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MFnkEG,MAAO,KEmkEE,GFnkEF,EEmkEO,CFnkEP,C;;MEq
kEd,OAAO,G;K;IAGX,iC;MAKI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,UAA
U,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UA
Aa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,0C;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,MA
AM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAA
W,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MA
KI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO,Q
AAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAj
C,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,0B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;Q
AAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,M
F16DG,MAAO,KE06DE,GF16DF,EE06DO,CF16DP,C;;ME46Dd,OAAO,G;K;IAGX,2B;MAWI,eAAe,oB;MACf,
IAAI,CAAC,QAAS,UAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;Q
ACI,QAAQ,QAAS,O;QACjB,MF18DG,MAAO,KE08DE,GF18DF,EE08DO,CF18DP,C;;ME48Dd,OAAO,G;K;IA
GX,2B;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MAC
nB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;M
AEvB,OAAO,G;K;kFAGX,yB;MAAA,sE;MAAA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAy
B,MAAM,6B;QAC/B,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAd,C;UAAyB,OAAO,O;QAChC,eAAe,SA
AS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YA
CI,UAAU,C;YACV,WAAW,C;;;QAED,QAAT,QAAS,W;QAClB,OAAO,O;O;KAxBX,C;8FA2BA,+B;MAOI,eA
Ae,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,cAAc,QAAS,O;MACvB,IAAI,CAAC,QAA
S,UAAd,C;QAAyB,OAAO,O;MAChC,eAAe,SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CA
AT,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,UAAU,C;UACV,WAAW,C;;;MAED,QAAT,QAAS,W;MACl

B,OAAO,O;K;mFAGX,yB;MAAA,sE;MFjhEA,iB;MEihEA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,
C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAA
S,QAAS,OAAlB,C;UACR,WF3hEG,MAAO,KE2hEO,QF3hEP,EE2hEiB,CF3hEjB,C;;QE6hEd,OAAO,Q;O;KAp
BX,C;mFAuBA,yB;MAAA,sE;MFnjEA,iB;MEmjEA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UA
AyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QA
AS,OAAlB,C;UACR,WF7jEG,MAAO,KE6jEO,QF7jEP,EE6jEiB,CF7jEjB,C;;QE+jEd,OAAO,Q;O;KApBX,C;mF
AuBA,yB;MAAA,sE;MAAA,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MAAM,6B;QAC/
B,eAAe,SAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAlB,C;UACR,I
AAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;+FAuBA,yB;MFtlEA,iB;MEslE
A,sC;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAlB
,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAlB,C;UACR,WF9lEG,MAAO,KE8lEO,QF9l
EP,EE8lEiB,CF9lEjB,C;;QEgmEd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MFtnEA,iB;MEsnEA,sC;QAWI,eAAe,oB;
QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAlB,C;QACf,OAAO,QAA
S,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAlB,C;UACR,WF9nEG,MAAO,KE8nEO,QF9nEP,EE8nEiB,CF9nEj
B,C;;QEgoEd,OAAO,Q;O;KAlBX,C;+FAqBA,+B;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,
OAAO,I;MAChC,eAAe,SAAS,QAAS,OAAlB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OA
AlB,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;0FAGX,yB;MAAA,sE;MA
AA,kD;QAWI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OA
AlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAlB,C;UACR,IAAI,UAAW,SAAQ,QAAR
,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;sGAuBA,2C;MAS
I,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,eAAe,SAAS,QAAS,OAAlB,C;MACf,
OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAlB,C;QACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CA
AlB,CAAX,GAAkC,CAAtC,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;IAGX,gC;MAOI,eAAe,oB;MACf,IAAI,C
AAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAA
Q,QAAS,O;QACjB,MFrsEG,MAAO,KEqsEE,GFrsEF,EEqsEO,CFrsEP,C;;MEusEd,OAAO,G;K;IAGX,iC;MAOI,
eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO,QAA
S,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MFjuEG,MAAO,KEiuEE,GFjuEF,EEiuEO,CFjuEP,C;;MEmuEd,OA
AO,G;K;IAGX,iC;MAKI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS
,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAA
M,C;;MAEvB,OAAO,G;K;IAGX,0C;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,MAAM,6B;M
AC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,G
AAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,gD;MAKI,eAAe,o
B;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAh
B,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC
,MAAM,C;;MAE9C,OAAO,G;K;IAGX,4B;MAMI,IAAI,oCAAJ,C;QAAwB,OAAO,mB;MAC/B,OAAO,CAAC,o
BAAW,U;K;iFAGvB,yB;MAAA,gD;MAAA,uC;QAOoB,Q;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,OAAO,
I;QAC5B,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,K;;Q
ACrD,OAAO,I;O;KARX,C;oFAWA,6B;MAKmC,Q;MAAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,
OAAO,OAAP,C;;MAArC,gB;K;kGAGJ,yB;MAAA,6B;MAAA,sC;MA3wBA,wE;MA2wBA,2BAQiB,yB;QAnxBj
B,wE;eAmxBiB,0B;UAAA,4B;YAAE,aAAe,c;YA5wBjB,gB;YADb,YAAY,C;YACC,2B;YAAb,OAAa,cAAb,C;c
AAa,sB;cAAM,OAAO,oBAAmB,cAAnB,EAAAmB,sBAAnB,UAAP,EAAoC,IAApC,C;;YA4wBmB,W;W;S;OAAz
B,C;MARjB,oC;QApwBiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAAM,OAAO,oBA
AmB,cAAnB,EAAmB,sBAAnB,UAAP,EAAoC,IAApC,C;;QA4wBnB,gB;O;KARJ,C;oFAWA,yB;MAAA,4F;MA
AA,uC;QAaI,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MAAM,mCAA8B,oCAA9B,C;QAC
/B,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,WAAV,EAAuB,QAAS,OAAhC,C;;
QAElB,OAAO,W;O;KAnBX,C;kGAsBA,yB;MAAA,4F;MAAA,wE;MAAA,uC;QAkBmD,Q;QAL/C,eAAe,SAAK
,W;QACpB,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MAAM,mCAA8B,oCAA9B,C;QAC/B,YAAY,C;QACZ,kBAA
qB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAmB,oBAAnB,QAAV,

EAAuC,WAAvC,EAAoD,QAAS,OAA7D,C;;QAElB,OAAO,W;O;KApBX,C;8GAuBA,yB;MAAA,wE;MAAA,uC
;QAkBmD,Q;QAL/C,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAd,C;UAAyB,OAAO,I;QAChC,YAAY,C;
QACZ,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAmB,oB
AAnB,QAAV,EAAuC,WAAvC,EAAoD,QAAS,OAA7D,C;;QAElB,OAAO,W;O;KApBX,C;gGAuBA,gC;MAcI,e
AAe,SAAK,W;MACpB,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,kBAAqB,QAAS,O;MAC9B,OA
AO,QAAS,UAAhB,C;QACI,cAAc,UAAU,WAAV,EAAuB,QAAS,OAAhC,C;;MAElB,OAAO,W;K;8FAGX,yB;
MAAA,4F;MAAA,uC;QAaI,eAAe,+BAAa,cAAb,C;QACf,IAAI,CAAC,QAAS,cAAd,C;UACI,MAAM,mCAA8B,
8BAA9B,C;QACV,kBAAqB,QAAS,W;QAC9B,OAAO,QAAS,cAAhB,C;UACI,cAAc,UAAU,QAAS,WAAnB,EA
A+B,WAA/B,C;;QAElB,OAAO,W;O;KApBX,C;4GAuBA,yB;MAAA,4F;MAAA,uC;QAaI,eAAe,+BAAa,cAAb,C
;QACf,IAAI,CAAC,QAAS,cAAd,C;UACI,MAAM,mCAA8B,8BAA9B,C;QACV,kBAAqB,QAAS,W;QAC9B,OA
AO,QAAS,cAAhB,C;UACI,YAAY,QAAS,gB;UACrB,cAAc,UAAU,KAAV,EAAiB,QAAS,WAA1B,EAAsC,WA
AtC,C;;QAElB,OAAO,W;O;KArBX,C;wHAwBA,gC;MAaI,eAAe,+BAAa,cAAb,C;MACf,IAAI,CAAC,QAAS,cA
Ad,C;QACI,OAAO,I;MACX,kBAAqB,QAAS,W;MAC9B,OAAO,QAAS,cAAhB,C;QACI,YAAY,QAAS,gB;QAC
rB,cAAc,UAAU,KAAV,EAAiB,QAAS,WAA1B,EAAsC,WAAtC,C;;MAElB,OAAO,W;K;0GAGX,gC;MAcI,eAA
e,+BAAa,cAAb,C;MACf,IAAI,CAAC,QAAS,cAAd,C;QACI,OAAO,I;MACX,kBAAqB,QAAS,W;MAC9B,OAA
O,QAAS,cAAhB,C;QACI,cAAc,UAAU,QAAS,WAAnB,EAA+B,WAA/B,C;;MAElB,OAAO,W;K;8FAGX,yB;M
AAA,kF;MAAA,gD;MAAA,gE;MAAA,gD;QAiBoB,Q;QAJhB,oBAAoB,mCAAwB,CAAxB,C;QACpB,IAAI,kB
AAiB,CAArB,C;UAAwB,OAAO,OAAO,OAAP,C;QACc,kBAAhC,eAAa,gBAAgB,CAAhB,IAAb,C;QAAwC,8B;
QAArD,aH7sFO,W;QG8sFP,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAc,UAAU,
WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArBX,C;4GAwBA,yB;MAAA,
kF;MAAA,gD;MAAA,gE;MAAA,gD;QAmBoB,UACY,M;QAN5B,oBAAoB,mCAAwB,CAAxB,C;QACpB,IAAI,
kBAAiB,CAArB,C;UAAwB,OAAO,OAAO,OAAP,C;QACc,kBAAhC,eAAa,gBAAgB,CAAhB,IAAb,C;QAAwC,
8B;QAArD,aHtuFO,W;QGuuFP,YAAY,C;QACZ,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;
UACZ,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAnB,EAAgC,OAAhC,C;UACd,MAAO,WAAI,WAAJ,
C;;QAEX,OAAO,M;O;KAvBX,C;kGA0BA,yB;MAAA,qD;MAAA,kF;MAAA,gE;MAAA,uC;QAcI,eAAe,SAAK,
W;QACpB,IAAI,CAAC,QAAS,UAAd,C;UAAyB,OAAO,W;QAChC,sBAAqB,QAAS,OAA9B,C;QACuD,kBAA1
C,eAAa,mCAAwB,EAAxB,CAAb,C;QAAkD,sBAAI,aAAJ,C;QAA/D,aHjwFO,W;QGkwFP,OAAO,QAAS,UAAh
B,C;UACI,gBAAc,UAAU,aAAV,EAAuB,QAAS,OAAhC,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;K
AtBX,C;gHAyBA,yB;MAAA,qD;MAAA,kF;MAAA,gE;MAAA,uC;QAoBgC,Q;QAN5B,eAAe,SAAK,W;QACpB
,IAAI,CAAC,QAAS,UAAd,C;UAAyB,OAAO,W;QAChC,sBAAqB,QAAS,OAA9B,C;QACuD,kBAA1C,eAAa,m
CAAwB,EAAxB,CAAb,C;QAAkD,sBAAI,aAAJ,C;QAA/D,aH1xFO,W;QG2xFP,YAAY,C;QACZ,OAAO,QAAS,
UAAhB,C;UACI,gBAAc,WAAU,YAAV,EAAU,oBAAV,SAAmB,aAAnB,EAAgC,QAAS,OAAzC,C;UACd,MAA
O,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;gFA0BA,yB;MArGA,kF;MAAA,gD;MAAA,gE;MAqGA,gD;Q
AcW,sB;;UAlGS,Q;UAJhB,oBAAoB,mCAAwB,CAAxB,C;UACpB,IAAI,kBAAiB,CAArB,C;YAAwB,qBAAO,O
AqGZ,OArGY,C;YAAP,uB;;UACqB,kBAAhC,eAAa,gBAAgB,CAAhB,IAAb,C;UAAwC,sBAoGlC,OApGkC,C;
UAArD,aH7sFO,W;UG8sFP,kBAmGmB,O;UAlGH,2B;UAAhB,OAAgB,cAAhB,C;YAAgB,yB;YACZ,cAiGwB,S
AjGV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QA8FP,yB;O;KAd
J,C;8FAiBA,yB;MA9FA,kF;MAAA,gD;MAAA,gE;MA8FA,gD;QAeW,6B;;UA1FS,gB;UALhB,oBAAoB,mCAA
wB,CAAxB,C;UACpB,IAAI,kBAAiB,CAArB,C;YAAwB,4BAAO,OA8FL,OA9FK,C;YAAP,8B;;UACqB,kBAAh
C,eAAa,gBAAgB,CAAhB,IAAb,C;UAAwC,sBA6F3B,OA7F2B,C;UAArD,aHtuFO,W;UGuuFP,YAAY,C;UACZ,
kBA2F0B,O;UA1FV,2B;UAAhB,OAAgB,cAAhB,C;YAAgB,yB;YACZ,cAyF+B,SAzFjB,EAAU,cAAV,EAAU,s
BAAV,WAAmB,WAAnB,EAAgC,OAAhC,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAsFP,gC;O;
KAfJ,C;kFAkBA,+B;MAOoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YA
AO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;8FAGX,+B;MAOoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,O
AAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAUoB,Q;MADh
B,UAAoB,C;MACJ,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G
;K;mFAGX,+B;MAUoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,S
AAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,yB;MAAA,SASoB,gB;MATpB,sC;QAUoB,Q;QADhB,Y;QAC

gB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAbX,C
;mFAgBA,yB;MlBjlFA,6B;MkBilFA,sC;QAWoB,Q;QADhB,UlBjlFmC,ckBilFnB,ClBjlFmB,C;QkBklFnB,2B;QA
AhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MlB/5FiD,ckB+5FjD,GlB/5F2D,KAAK,GkB+5FzD,SAAS,OAAT,Cl
B/5FoE,KAAX,IAAf,C;;QkBi6FrD,OAAO,G;O;KAdX,C;mFAiBA,yB;MD/lFA,+B;MC+lFA,sC;QAWoB,Q;QAD
hB,UD9lFqC,eAAW,oBC8lF/B,CD9lF+B,CAAX,C;QC+lFrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,
MD76FmD,eC66FnD,GD76F8D,KAAK,KC66F5D,SAAS,OAAT,CD76FuE,KAAX,CAAhB,C;;QC+6FvD,OAAO,
G;O;KAdX,C;IAiBA,qC;MAIoB,UAMT,M;MANS,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,eAA
J,C;UACI,MAAM,gCAAyB,2BAAwB,SAAxB,MAAzB,C;;;MAId,OAAO,mE;K;IAGX,qC;MAIoB,UAMT,M;MA
NS,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,eAAJ,C;UACI,MAAM,gCAAyB,2BAAwB,SAAxB,
MAAzB,C;;;MAId,OAAO,+D;K;IAGX,kC;MAWI,OAAO,oBAAS,IAAT,EAAe,IAAf,EAAsC,IAAtC,C;K;IAGX,
+C;MAgBI,OAAO,sBAAS,IAAT,EAAe,IAAf,EAAsC,IAAtC,EAAwD,SAAxD,C;K;IAGX,mC;MAII,aAAa,iBAA
a,mCAAwB,EAAxB,CAAb,C;MACb,kBAAc,KAAd,C;MA7uEgB,Q;MAAA,OA8uET,SA9uES,W;MAAhB,OAA
gB,cAAhB,C;QAAgB,2B;QAAU,oB;QA8uEK,IAAI,CAAC,SAAD,IAAY,OA9uEX,SA8uEW,UAAhB,C;UAAiC,
YAAU,I;UAA3C,mBAAiD,K;;UAAjD,mBAA8D,I;;QA9uEvE,qB;UA8uED,MA9uEqC,WAAI,SAAJ,C;;MA8uE1
D,OAAqB,M;K;IAGzB,sC;MAII,IAAI,QrBypJG,YAAQ,CqBzpJf,C;QAAwB,OAAY,SAAL,SAAK,C;MA3xE7B,
kBAAY,gB;MA4BH,Q;MAAA,OAgwET,SAhwES,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAg
wEF,mBAhwEa,OAgwEb,CAhwEF,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAgwE3D,OA/vEO,W;K;IAkwEX,sC;
MAII,YAAqB,6BAAT,QAAS,C;MACrB,IAAI,KAAM,UAAV,C;QACI,OAAY,SAAL,SAAK,C;MAryET,kBAAY
,gB;MA4BH,Q;MAAA,OA0wET,SA1wES,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CA0wEF,qB
A1wEa,OA0wEb,CA1wEF,C;UAAyB,WAAY,WAAI,OAAJ,C;;MA0wE3D,OAzwEO,W;K;IA4wEX,sC;MAII,YA
AqB,UAAT,QAAS,C;MACrB,IAAI,KAAM,UAAV,C;QACI,OAAY,SAAL,SAAK,C;MA/yET,kBAAY,gB;MA4B
H,Q;MAAA,OAoxET,SApxES,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAoxEF,qBApxEa,OAo
xEb,CApxEF,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAoxE3D,OAnxEO,W;K;8FAsxEX,yB;MAAA,8C;MAAA,qC;
QAKI,OAAO,iBAAM,OAAN,C;O;KALX,C;0FAQA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAUoB,Q;QAFhB,Y
AAY,gB;QACZ,aAAa,gB;QACG,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,
C;YACI,KAAM,WAAI,OAAJ,C;;YAEN,MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;
O;KAjBX,C;IAoBA,kC;MAII,IAAI,oCAAJ,C;QAAwB,OAAY,OAAL,SAAK,EAAK,OAAL,C;MACpC,aAAa,gB;
MACN,OAAP,MAAO,EAAO,SAAP,C;MACP,MAAO,WAAI,OAAJ,C;MACP,OAAO,M;K;IAGX,oC;MAII,aAA
a,iBAAa,iBAAO,CAAP,IAAb,C;MACb,MAAO,gBAAO,SAAP,C;MACP,MAAO,WAAI,OAAJ,C;MACP,OAAO,
M;K;IAGX,qC;MAII,IAAI,oCAAJ,C;QAAwB,OAAY,OAAL,SAAK,EAAK,QAAL,C;MACpC,aAAa,gB;MACN,
OAAP,MAAO,EAAO,SAAP,C;MACA,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;IAGX,qC;MAII,aAA
a,iBAAa,SAAK,KAAL,GAAY,QAAS,OAArB,IAAb,C;MACb,MAAO,gBAAO,SAAP,C;MACA,SAAP,MAAO,E
AAO,QAAP,C;MACP,OAAO,M;K;IAGX,qC;MAII,IAAI,oCAAJ,C;QAAwB,OAAY,OAAL,SAAK,EAAK,QAAL
,C;MACpC,aAAa,gB;MACN,OAAP,MAAO,EAAO,SAAP,C;MACA,OAAP,MAAO,EAAO,QAAP,C;MACP,OA
AO,M;K;IAGX,qC;MAII,IAAI,mCAAJ,C;QACI,aAAa,iBAAa,SAAK,KAAL,GAAY,QAAS,KAArB,IAAb,C;QA
Cb,MAAO,gBAAO,SAAP,C;QACP,MAAO,gBAAO,QAAP,C;QACP,OAAO,M;;QAEP,eAAa,iBAAa,SAAb,C;Q
ACN,OAAP,QAAO,EAAO,QAAP,C;QACP,OAAO,Q;;K;IAIf,qC;MAII,aAAa,gB;MACN,OAAP,MAAO,EAAO,
SAAP,C;MACA,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;IAGX,qC;MAII,aAAa,iBAAa,SAAK,KAAL
,GAAY,EAAZ,IAAb,C;MACb,MAAO,gBAAO,SAAP,C;MACA,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,
M;K;4FAGX,yB;MAAA,4C;MAAA,qC;QAKI,OAAO,gBAAK,OAAL,C;O;KALX,C;8FAQA,yB;MAAA,4C;MA
AA,qC;QAKI,OAAO,gBAAK,OAAL,C;O;KALX,C;IAQA,yD;MAgB+C,oB;QAAA,OAAY,C;MAAG,8B;QAAA,
iBAA0B,K;MAOzE,Q;MANX,oBAAoB,IAApB,EAA0B,IAA1B,C;MACA,IAAI,0CAAwB,8BAA5B,C;QACI,eA
Ae,SAAK,K;QACpB,qBAAqB,YAAW,IAAX,SAAsB,YAAW,IAAX,UAAmB,CAAvB,GAA0B,CAA1B,GAAiC,
CAAnD,K;QACrB,aAAa,iBAAmB,cAAnB,C;QACb,gBAAY,CAAZ,C;QACA,Y;UAAO,c;UAAP,MAAgB,CAAT
,mBAAiB,QAAxB,E;YAAA,K;UACI,iBAAsB,eAAL,IAAK,EAAa,WAAW,OAAX,IAAb,C;UACtB,IAAI,aAAa,I
AAb,IAAqB,CAAC,cAA1B,C;YAA0C,K;UhB3mGlD,WAAW,iBgB4mGa,UhB5mGb,C;UaCX,mBAAc,CAAd,Y
G2mGwB,UH3mGxB,Y;YbA6B,egB2mGS,sBH1mG3B,OG0mGgC,GAAK,OAAL,IAAL,ChB3mGT,C;;UgB2mG
rB,MAAO,WhB1mGR,IgB0mGQ,C;UACP,oBAAS,IAAT,I;;QAEJ,OAAO,M;;MAEX,eAAa,gB;MACiE,kBAA9E,

iBAAiB,oBAAjB,EAA6B,IAA7B,EAAmC,IAAnC,EAAyC,cAAzC,EAAuE,KAAvE,C;ME3uGA,OAAgB,qBAAh
B,C;QAAgB,gC;QF4uGL,mBE5uGqB,OF4uGrB,C;;MAEX,OAAO,Q;K;IAGX,sE;MAkBkD,oB;QAAA,OAAY,C;
MAAG,8B;QAAA,iBAA0B,K;MACvF,oBAAoB,IAApB,EAA0B,IAA1B,C;MACA,IAAI,0CAAwB,8BAA5B,C;Q
ACI,eAAe,SAAK,K;QACpB,qBAAqB,YAAW,IAAX,SAAsB,YAAW,IAAX,UAAmB,CAAvB,GAA0B,CAA1B,
GAAiC,CAAnD,K;QACrB,aAAa,iBAAa,cAAb,C;QACb,eAAa,kBAAc,SAAd,C;QACb,YAAY,C;QACZ,OAAgB,
CAAT,qBAAiB,QAAxB,C;UACI,iBAAsB,eAAL,IAAK,EAAa,WAAW,KAAX,IAAb,C;UACtB,IAAI,CAAC,cAA
D,IAAmB,aAAa,IAApC,C;YAA0C,K;UAC1C,QAAO,cAAK,KAAL,EAAY,QAAQ,UAAR,IAAZ,C;UACP,MAA
O,WAAI,UAAU,QAAV,CAAJ,C;UACP,gBAAS,IAAT,I;;QAEJ,OAAO,M;;MAEX,eAAa,gB;MACgE,kBAA7E,i
BAAiB,oBAAjB,EAA6B,IAA7B,EAAmC,IAAnC,EAAyC,cAAzC,EAAuE,IAAvE,C;MErxGA,OAAgB,qBAAhB,
C;QAAgB,gC;QFsxGL,mBAAI,UEtxGiB,OFsxGjB,CAAJ,C;;MAEX,OAAO,Q;K;IAGX,kC;MAqBoB,gB;MAHh
B,gBAXW,KAWW,O;MACtB,WAAW,iBFzkGJ,MAAO,KEykGgB,mCAAwB,EAAxB,CFzkGhB,EEykG6C,SFzk
G7C,CEykGH,C;MACX,QAAQ,C;MACQ,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT
,C;UAAoB,K;QACpB,IAAK,WAhBqB,GAgBP,OAhBO,EAAnB,KAgBqB,CAAM,UAAN,EAAM,kBAAN,SAhB
F,CAgBrB,C;;MAhBT,OAkBO,I;K;+EAfX,yB;MAAA,kF;MAAA,gE;MFtkGA,iB;MEskGA,8C;QAWoB,UAEsB,
M;QALtC,gBAAgB,KAAM,O;QACtB,WAAW,eFzkGJ,MAAO,KEykGgB,mCAAwB,EAAxB,CFzkGhB,EEykG6
C,SFzkG7C,CEykGH,C;QACX,QAAQ,C;QACQ,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,
SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,OAAV,EAAmB,MAAM,UAAN,EAAM,kBAAN,SAAnB,CAA
J,C;;QAET,OAAO,I;O;KAfX,C;IAkBA,kC;MAkBI,YAAY,oB;MACZ,aAZW,KAYQ,W;MACnB,WAAW,iBFtm
GJ,MAAO,KEsmGgB,mCAAwB,EAAxB,CFtmGhB,EEsmGmD,wBAbtD,KAasD,EAAwB,EAAxB,CFtmGnD,CE
smGH,C;MACX,OAAO,KAAM,UAAN,IAAmB,MAAO,UAAjC,C;QACI,IAAK,WAfqB,GAeP,KAAM,OAfC,EA
eO,MAAO,OAfd,CAerB,C;;MAfT,OAiBO,I;K;+EAdX,yB;MAAA,kF;MAAA,gE;MFlmGA,iB;MEkmGA,8C;QA
QI,YAAY,oB;QACZ,aAAa,KAAM,W;QACnB,WAAW,eFtmGJ,MAAO,KEsmGgB,mCAAwB,EAAxB,CFtmGhB
,EEsmGmD,wBAAN,KAAM,EAAwB,EAAxB,CFtmGnD,CEsmGH,C;QACX,OAAO,KAAM,UAAN,IAAmB,MA
AO,UAAjC,C;UACI,IAAK,WAAI,UAAU,KAAM,OAAhB,EAAwB,MAAO,OAA/B,CAAJ,C;;QAET,OAAO,I;O;
KAdX,C;IAiBA,gC;MASW,sB;;QAaP,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,qBAAO,W;UAAP,u
B;;QACzB,ahB9yGoD,gB;QgB+yGpD,cAAc,QAAS,O;QACvB,OAAO,QAAS,UAAhB,C;UACI,WAAW,QAAS,O
;UACpB,MAAO,WAnBkB,GAmBJ,OAnBI,EAmBK,IAnBL,CAmBlB,C;UACP,UAAU,I;;QAEd,qBAAO,M;;;MA
tBP,yB;K;8FAGJ,yB;MAAA,qD;MhBxyGA,+D;MgBwyGA,uC;QAUI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAA
d,C;UAAyB,OAAO,W;QAChC,ahB9yGoD,gB;QgB+yGpD,cAAc,QAAS,O;QACvB,OAAO,QAAS,UAAhB,C;UA
CI,WAAW,QAAS,O;UACpB,MAAO,WAAI,UAAU,OAAV,EAAmB,IAAnB,CAAJ,C;UACP,UAAU,I;;QAEd,OA
AO,M;O;KAnBX,C;IAsBA,8F;MAQ6D,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UA
AwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MAGtN,Q;MAFhB
,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,iCA
AU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KAA1B,C;UACW,gBA
AP,MAAO,EAAc,OAAd,EAAuB,SAAvB,C;;UACJ,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QAAQ,KAA1B,C;QAAi
C,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,4F;MAQwC,yB;QAA
A,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QA
AA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MACjN,OAAO,oBAAO,sBAAP,EAAwB,SAAxB,EAAmC,MAAnC,
EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAiF,W;K;4FAG5F,qB;MAKI,OAAO,S;K;I
ASS,8C;MAAA,mB;QAAE,OAAAA,eAAK,W;O;K;IAN3B,iC;MAMI,oCAAgB,8BAAhB,C;K;IAGJ,+B;MAOoB,Q
;MAFhB,UAAkB,G;MAClB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QAC
P,oBAAmB,qBAAnB,EAAmB,KAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAA
M,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB,G;MAClB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QA
AgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,
wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB,G;MAClB,YAAiB,C;MACD,2B;
MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAnB,E;;MAEJ,
OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB,G;
MAClB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAn

B,EAAmB,KAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;M
AOoB,Q;MAFhB,UAAkB,G;MAClB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAA
O,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAA
gC,MAAM,K;K;IAGjD,+B;MAOoB,Q;MAFhB,UAAkB,G;MAClB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAh
B,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAnB,E;;MAEJ,OAAW,UAAS,CAAb,G
AAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,2B;MAMoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAA
gB,cAAhB,C;QAAgB,yB;QACZ,YAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAMoB,Q;MADhB,UAAe,C;MACC,2
B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAMoB,Q;MADhB,
UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,OAAP,I;;MAEJ,OAAO,G;K;IAGX,2
B;MAMoB,Q;MADhB,Y;MACgB,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,cAAO,OAAP,C;;MAEJ,OA
AO,G;K;IAGX,2B;MAMoB,Q;MADhB,UAAiB,G;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OA
AO,O;;MAEX,OAAO,G;K;IAGX,2B;MAMoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAgB,cAAhB,C;QA
AgB,yB;QACZ,OAAO,O;;MAEX,OAAO,G;K;IGl+GX,uC;MAOI,OAAO,SAAM,CAAN,EAAS,SAAM,CAAN,E
AAS,CAAT,EAAY,UAAZ,CAAT,EAAkC,UAAlC,C;K;IAGX,oC;MAOI,OAAW,UAAW,SAAQ,CAAR,EAAW,C
AAX,CAAX,IAA4B,CAAhC,GAAmC,CAAnC,GAA0C,C;K;IAmDrD,wC;MAQc,Q;MADV,UAAU,C;MACV,wB
AAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CA
AjC,C;UAAoC,MAAM,C;;MAC3D,OAAO,G;K;IA+GX,uC;MAOI,OAAO,SAAM,CAAN,EAAS,SAAM,CAAN,E
AAS,CAAT,EAAY,UAAZ,CAAT,EAAkC,UAAlC,C;K;IAGX,oC;MAOI,OAAW,UAAW,SAAQ,CAAR,EAAW,C
AAX,CAAX,IAA4B,CAAhC,GAAmC,CAAnC,GAA0C,C;K;IAmDrD,wC;MAQc,Q;MADV,UAAU,C;MACV,wB
AAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CA
AjC,C;UAAoC,MAAM,C;;MAC3D,OAAO,G;K;oGCnXX,yB;MAAA,iE;MAAA,uC;QAS8C,IAAnC,I;QAAA,+B;
;UAYS,U;UAAA,SnBgVoE,iBAAQ,W;UmBhV5F,OAAgB,gBAAhB,C;YAAgB,2B;YACZ,aAbwB,SAaX,CAAU,
OAAV,C;YACb,IAAI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;;UAGR,8BAAO,I;;;QAlBA,kC;QAAA,iB;UAAmC,M
AAM,gCAAuB,4DAAvB,C;;QAAhD,OAAO,I;O;KATX,C;gHAYA,gC;MASoB,Q;MAAA,OAAA,SnBgVoE,QA
AQ,W;MmBhV5F,OAAgB,cAAhB,C;QAAgB,yB;QACZ,aAAa,UAAU,OAAV,C;QACb,IAAI,cAAJ,C;UACI,OA
AO,M;;;MAGf,OAAO,I;K;IAGX,6B;MAII,IAAI,mBAAQ,CAAZ,C;QACI,OAAO,W;MACX,eAAe,iBAAQ,W;M
ACvB,IAAI,CAAC,QAAS,UAAd,C;QACI,OAAO,W;MACX,YAAY,QAAS,O;MACrB,IAAI,CAAC,QAAS,UAA
d,C;QACI,OAAO,OnBgQiD,SmBhQ1C,KnBgQ+C,IAAL,EmBhQ1C,KnBgQoD,MAAV,CmBhQjD,C;;MACX,aA
Aa,iBAAsB,cAAtB,C;MACb,MAAO,WnB8PqD,SmB9PjD,KnB8PsD,IAAL,EmB9PjD,KnB8P2D,MAAV,CmB9P
rD,C;;QAEwB,kBAAhB,QAAS,O;QAApB,MAAO,WnB4PiD,SAAK,eAAL,EAAU,iBAAV,CmB5PjD,C;;MACO,
QAAT,QAAS,W;MAClB,OAAO,M;K;uFAGX,yB;MAAA,+D;MAsBA,gD;MAtBA,uC;QAMW,kBAAU,gB;QAo
BD,Q;QAAA,OnByRoE,iBAAQ,W;QmBzR5F,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WArB6B,SAqBlB,CAAU,O
AAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAtBhB,OAwBO,W;O;KA9BX,C;uFASA,yB;MAAA,+D;MAwB
A,gD;MAxBA,uC;QAUW,kBAAU,gB;QAsBD,Q;QAAA,OnB0QoE,iBAAQ,W;QmB1Q5F,OAAgB,cAAhB,C;UA
AgB,yB;UACZ,WAvB6B,SAuBlB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAxBhB,OA0BO,W;
O;KApCX,C;2FAaA,yB;MAAA,gD;MAAA,oD;QAIoB,Q;QAAA,OAAA,SnByRoE,QAAQ,W;QmBzR5F,OAAgB
,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,
W;O;KARX,C;2FAWA,yB;MAAA,gD;MAAA,oD;QAQoB,Q;QAAA,OAAA,SnB0QoE,QAAQ,W;QmB1Q5F,OA
AgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OA
AO,W;O;KAZX,C;8EAeA,yB;MAAA,gE;MAAA,uC;QAOW,kBAAM,eAAa,cAAb,C;QA2BA,Q;QAAA,OnBiOu
E,iBAAQ,W;QmBjO5F,OAAa,cAAb,C;UAAa,sB;UACT,WAAY,WA5BiB,SA4Bb,CAAU,IAAV,CAAJ,C;;QA5B
hB,OA6BO,W;O;KApCX,C;4FAUA,yB;MAAA,+D;MAAA,uC;QAOW,kBAAa,gB;QAgFJ,Q;QAAA,OnBkKoE,i
BAAQ,W;QmBlK5F,OAAgB,cAAhB,C;UAAgB,yB;UAxEK,U;UAAA,cARe,SAQf,CAwEQ,OAxER,W;YAAsC,6
B;;;QAR3D,OASO,W;O;KAhBX,C;gGAUA,yB;MAAA,oD;QA6EoB,Q;QAAA,OnBkKoE,iBAAQ,W;QmBlK5F,
OAAgB,cAAhB,C;UAAgB,yB;UAxEK,U;UAAA,wBAwEQ,OAxER,W;YAAsC,6B;;;QAC3D,OAAO,W;O;KAN
X,C;kFASA,6C;MAKiB,Q;MAAA,OAAA,SnBiOuE,QAAQ,W;MmBjO5F,OAAa,cAAb,C;QAAa,sB;QACT,WAA
Y,WAAI,UAAU,IAAV,CAAJ,C;;MAChB,OAAO,W;K;8EAGX,gC;MAWoB,Q;MADhB,IAAI,mBAAJ,C;QAAe,
OAAO,I;MACN,OAAA,SnBiNoE,QAAQ,W;MmBjN5F,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UA

AU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;IAGX,2B;MAMI,OAAO,CAAC,mB;K;+EAGZ,gC;
MAOoB,Q;MADhB,IAAI,mBAAJ,C;QAAe,OAAO,K;MACN,OAAA,SnB6LoE,QAAQ,W;MmB7L5F,OAAgB,cA
AhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;mFAGX,qB;M
AKI,OAAO,c;K;mFAGX,gC;MAMoB,Q;MAFhB,IAAI,mBAAJ,C;QAAe,OAAO,C;MACtB,YAAY,C;MACI,OA
AA,SnB2KoE,QAAQ,W;MmB3K5F,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAw
B,qB;;MAC9C,OAAO,K;K;sFAGX,6B;MAKoB,Q;MAAA,OAAA,SnBkKoE,QAAQ,W;MmBlK5F,OAAgB,cAAh
B,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;K;kFAG1B,yB;MJ+qDA,sE;MI/qDA,sC;QAYmB,kBAAR,iB;QAAQ,g
B;;UJ8qDf,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAd,C;YAAyB,MAAM,6B;UAC/B,cAAc,QAAS,O;UACvB,IA
AI,CAAC,QAAS,UAAd,C;YAAyB,eAAO,O;YAAP,iB;;UACzB,eIlrDqB,QJkrDN,CAAS,OAAT,C;;YAEX,QAA
Q,QAAS,O;YACjB,QIrrDiB,QJqrDT,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAAU,C;cAC
V,WAAW,C;;;UAED,QAAT,QAAS,W;UAClB,eAAO,O;;;QI3rDP,mB;O;KAZJ,C;8FAeA,+B;MAQmB,kBAAR,i
B;MAAQ,sB;;QJ0rDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,qBAAO,I;UAAP,uB;;QACzB,cAAc,
QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAd,C;UAAyB,qBAAO,O;UAAP,uB;;QACzB,eI9rD2B,QJ8rDZ,CAAS,
OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QIjsDuB,QJisDf,CAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,
C;YACI,UAAU,C;YACV,WAAW,C;;;QAED,QAAT,QAAS,W;QAClB,qBAAO,O;;;MIvsDP,yB;K;mFAGJ,yB;M
JusDA,sE;MFn3DA,iB;MM4KA,sC;QJotDI,eIvsDO,iBJusDQ,W;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MA
AM,6B;QAC/B,eIzsDqB,QJysDN,CAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI3sDiB,QJ2s
DT,CAAS,QAAS,OAAlB,C;UACR,WF73DG,MAAO,KE63DO,QF73DP,EE63DiB,CF73DjB,C;;QMiLd,OJ8sDO,
Q;O;KI3tDX,C;mFAgBA,yB;MJ8sDA,sE;MFr5DA,iB;MMuMA,sC;QJ2tDI,eI9sDO,iBJ8sDQ,W;QACf,IAAI,CAA
C,QAAS,UAAd,C;UAAyB,MAAM,6B;QAC/B,eIhtDqB,QJgtDN,CAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,U
AAhB,C;UACI,QIltDiB,QJktDT,CAAS,QAAS,OAAlB,C;UACR,WF/5DG,MAAO,KE+5DO,QF/5DP,EE+5DiB,C
F/5DjB,C;;QM4Md,OJqtDO,Q;O;KIluDX,C;mFAgBA,yB;MJqtDA,sE;MIrtDA,sC;QJguDI,eIrtDO,iBJqtDQ,W;QA
Cf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MAAM,6B;QAC/B,eIvtDqB,QJutDN,CAAS,QAAS,OAAlB,C;QACf,O
AAO,QAAS,UAAhB,C;UACI,QIztDiB,QJytDT,CAAS,QAAS,OAAlB,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;Y
ACI,WAAW,C;;;QI3tDnB,OJ8tDO,Q;O;KIzuDX,C;+FAcA,yB;MN1NA,iB;MM0NA,sC;QAWmB,kBAAR,iB;QA
AQ,sB;;UJ8tDf,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAd,C;YAAyB,qBAAO,I;YAAP,uB;;UACzB,eIhuD2B,QJ
guDZ,CAAS,QAAS,OAAlB,C;UACf,OAAO,QAAS,UAAhB,C;YACI,QIluDuB,QJkuDf,CAAS,QAAS,OAAlB,C;
YACR,WFh8DG,MAAO,KEg8DO,QFh8DP,EEg8DiB,CFh8DjB,C;;UEk8Dd,qBAAO,Q;;;QIruDP,yB;O;KAXJ,C;
+FAcA,yB;MNnPA,iB;MMmPA,sC;QAWmB,kBAAR,iB;QAAQ,sB;;UJquDf,eAAe,sB;UACf,IAAI,CAAC,QAAS
,UAAd,C;YAAyB,qBAAO,I;YAAP,uB;;UACzB,eIvuD2B,QJuuDZ,CAAS,QAAS,OAAlB,C;UACf,OAAO,QAAS,
UAAhB,C;YACI,QIzuDuB,QJyuDf,CAAS,QAAS,OAAlB,C;YACR,WFh+DG,MAAO,KEg+DO,QFh+DP,EEg+D
iB,CFh+DjB,C;;UEk+Dd,qBAAO,Q;;;QI5uDP,yB;O;KAXJ,C;+FAcA,+B;MASmB,kBAAR,iB;MAAQ,sB;;QJ4uD
f,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,qBAAO,I;UAAP,uB;;QACzB,eI9uD2B,QJ8uDZ,CAAS,Q
AAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIhvDuB,QJgvDf,CAAS,QAAS,OAAlB,C;UACR,IAAI,2
BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,qBAAO,Q;;;MIrvDP,yB;K;0FAGJ,yB;MJqvDA,sE;MIrvDA,k
D;QJgwDI,eIrvDO,iBJqvDQ,W;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MAAM,6B;QAC/B,eIvvDqC,QJuvD
tB,CAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIzvDiC,QJyvDzB,CAAS,QAAS,OAAlB,C;U
ACR,II1vDqB,UJ0vDN,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QI3vDnB,OJ
8vDO,Q;O;KIzwDX,C;sGAcA,2C;MASmB,kBAAR,iB;MAAQ,0B;;QJ8vDf,eAAe,sB;QACf,IAAI,CAAC,QAAS,
UAAd,C;UAAyB,yBAAO,I;UAAP,2B;;QACzB,eIhwD2C,QJgwD5B,CAAS,QAAS,OAAlB,C;QACf,OAAO,QAA
S,UAAhB,C;UACI,QIlwDuC,QJkwD/B,CAAS,QAAS,OAAlB,C;UACR,IInwD2B,UJmwDZ,SAAQ,QAAR,EAAk
B,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,yBAAO,Q;;;MIvwDP,6B;K;sFAGJ,yB;MAAA,k
D;MAAA,wC;QAUI,OAAe,QAAR,iBAAQ,EAAQ,UAAR,C;O;KAVnB,C;kGAaA,yB;MAAA,8D;MAAA,wC;QA
MI,OAAe,cAAR,iBAAQ,EAAc,UAAd,C;O;KANnB,C;kFASA,yB;MJi4DA,sE;MIj4DA,sC;QAYmB,kBAAR,iB;Q
AAQ,gB;;UJg4Df,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAd,C;YAAyB,MAAM,6B;UAC/B,cAAc,QAAS,O;UA
CvB,IAAI,CAAC,QAAS,UAAd,C;YAAyB,eAAO,O;YAAP,iB;;UACzB,eIp4DqB,QJo4DN,CAAS,OAAT,C;;YAE
X,QAAQ,QAAS,O;YACjB,QIv4DiB,QJu4DT,CAAS,CAAT,C;YACR,IAAI,2BAAW,CAAX,KAAJ,C;cACI,UAA
U,C;cACV,WAAW,C;;;UAED,QAAT,QAAS,W;UAClB,eAAO,O;;;QI74DP,mB;O;KAZJ,C;8FAeA,+B;MAQmB,

kBAAR,iB;MAAQ,sB;;QJ44Df,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,qBAAO,I;UAAP,uB;;QAC
zB,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAd,C;UAAyB,qBAAO,O;UAAP,uB;;QACzB,eIh5D2B,QJg5D
Z,CAAS,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QIn5DuB,QJm5Df,CAAS,CAAT,C;UACR,IAAI,2BAAW,CA
AX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAED,QAAT,QAAS,W;QAClB,qBAAO,O;;;MIz5DP,yB;K;mF
AGJ,yB;MJy5DA,sE;MFjhEA,iB;MMwHA,sC;QJs6DI,eIz5DO,iBJy5DQ,W;QACf,IAAI,CAAC,QAAS,UAAd,C;
UAAyB,MAAM,6B;QAC/B,eI35DqB,QJ25DN,CAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,Q
I75DiB,QJ65DT,CAAS,QAAS,OAAlB,C;UACR,WF3hEG,MAAO,KE2hEO,QF3hEP,EE2hEiB,CF3hEjB,C;;QM6
Hd,OJg6DO,Q;O;KI76DX,C;mFAgBA,yB;MJg6DA,sE;MFnjEA,iB;MMmJA,sC;QJ66DI,eIh6DO,iBJg6DQ,W;QA
Cf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MAAM,6B;QAC/B,eIl6DqB,QJk6DN,CAAS,QAAS,OAAlB,C;QACf,O
AAO,QAAS,UAAhB,C;UACI,QIp6DiB,QJo6DT,CAAS,QAAS,OAAlB,C;UACR,WF7jEG,MAAO,KE6jEO,QF7j
EP,EE6jEiB,CF7jEjB,C;;QMwJd,OJu6DO,Q;O;KIp7DX,C;mFAgBA,yB;MJu6DA,sE;MIv6DA,sC;QJk7DI,eIv6D
O,iBJu6DQ,W;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MAAM,6B;QAC/B,eIz6DqB,QJy6DN,CAAS,QAAS,
OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI36DiB,QJ26DT,CAAS,QAAS,OAAlB,C;UACR,IAAI,2BAA
W,CAAX,KAAJ,C;YACI,WAAW,C;;;QI76DnB,OJg7DO,Q;O;KI37DX,C;+FAcA,yB;MNtKA,iB;MMsKA,sC;QA
WmB,kBAAR,iB;QAAQ,sB;;UJg7Df,eAAe,sB;UACf,IAAI,CAAC,QAAS,UAAd,C;YAAyB,qBAAO,I;YAAP,uB;
;UACzB,eIl7D2B,QJk7DZ,CAAS,QAAS,OAAlB,C;UACf,OAAO,QAAS,UAAhB,C;YACI,QIp7DuB,QJo7Df,CA
AS,QAAS,OAAlB,C;YACR,WF9lEG,MAAO,KE8lEO,QF9lEP,EE8lEiB,CF9lEjB,C;;UEgmEd,qBAAO,Q;;;QIv7
DP,yB;O;KAXJ,C;+FAcA,yB;MN/LA,iB;MM+LA,sC;QAWmB,kBAAR,iB;QAAQ,sB;;UJu7Df,eAAe,sB;UACf,I
AAI,CAAC,QAAS,UAAd,C;YAAyB,qBAAO,I;YAAP,uB;;UACzB,eIz7D2B,QJy7DZ,CAAS,QAAS,OAAlB,C;U
ACf,OAAO,QAAS,UAAhB,C;YACI,QI37DuB,QJ27Df,CAAS,QAAS,OAAlB,C;YACR,WF9nEG,MAAO,KE8nE
O,QF9nEP,EE8nEiB,CF9nEjB,C;;UEgoEd,qBAAO,Q;;;QI97DP,yB;O;KAXJ,C;+FAcA,+B;MASmB,kBAAR,iB;
MAAQ,sB;;QJ87Df,eAAe,sB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,qBAAO,I;UAAP,uB;;QACzB,eIh8D2B
,QJg8DZ,CAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QIl8DuB,QJk8Df,CAAS,QAAS,OAAlB
,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,qBAAO,Q;;;MIv8DP,yB;K;0FAGJ,yB;MJu8
DA,sE;MIv8DA,kD;QJk9DI,eIv8DO,iBJu8DQ,W;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MAAM,6B;QAC/
B,eIz8DqC,QJy8DtB,CAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QI38DiC,QJ28DzB,CAAS,
QAAS,OAAlB,C;UACR,II58DqB,UJ48DN,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAA
W,C;;;QI78DnB,OJg9DO,Q;O;KI39DX,C;sGAcA,2C;MASmB,kBAAR,iB;MAAQ,0B;;QJg9Df,eAAe,sB;QACf,I
AAI,CAAC,QAAS,UAAd,C;UAAyB,yBAAO,I;UAAP,2B;;QACzB,eIl9D2C,QJk9D5B,CAAS,QAAS,OAAlB,C;Q
ACf,OAAO,QAAS,UAAhB,C;UACI,QIp9DuC,QJo9D/B,CAAS,QAAS,OAAlB,C;UACR,IIr9D2B,UJq9DZ,SAA
Q,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,yBAAO,Q;;;MIz9DP,6B;K;sFAG
J,yB;MAAA,kD;MAAA,wC;QAUI,OAAe,QAAR,iBAAQ,EAAQ,UAAR,C;O;KAVnB,C;kGAaA,yB;MAAA,8D;
MAAA,wC;QAMI,OAAe,cAAR,iBAAQ,EAAc,UAAd,C;O;KANnB,C;IASA,4B;MAMI,OAAO,mB;K;iFAGX,gC;
MAOoB,Q;MADhB,IAAI,mBAAJ,C;QAAe,OAAO,I;MACN,OAAA,SnBjLoE,QAAQ,W;MmBiL5F,OAAgB,cAA
hB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;oFAGX,6B;MA
KmC,Q;MAAA,OnB1LqD,iBAAQ,W;MmB0L7E,OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;MAAr
C,gB;K;kGAGJ,yB;MAAA,6B;MAAA,sC;MJ4wCA,wE;MI5wCA,2BAQiB,yB;QJowCjB,wE;eIpwCiB,0B;UAAA
,4B;YAAU,kBAAR,iB;YAAQ,aAAe,c;YJ2wCzB,gB;YADb,YAAY,C;YACC,6B;YAAb,OAAa,cAAb,C;cAAa,sB;
cAAM,OAAO,oBAAmB,cAAnB,EAAmB,sBAAnB,UAAP,EAAoC,IAApC,C;;YI3wC2B,W;W;S;OAAjC,C;MAR
jB,oC;QJmxCiB,gB;QADb,YAAY,C;QACC,OI3wCE,iBJ2wCF,W;QAAb,OAAa,cAAb,C;UAAa,sB;UAAM,OAA
O,oBAAmB,cAAnB,EAAmB,sBAAnB,UAAP,EAAoC,IAApC,C;;QI3wCnB,gB;O;KARJ,C;4FAWA,qB;MAKI,O
AAO,iB;K;IAGX,iC;MAII,OAAe,aAAR,iBAAQ,C;K;IChkBnB,kC;MAEI,gBCmE2D,8BAAY,c;MDlEvE,IAAI,S
AAU,OAAV,GAAmB,CAAvB,C;QACW,Q;QAAA,IAAI,cAAQ,GAAZ,C;UAAA,OAAsB,S;;uBAAe,qBAAU,CA
AV,C;UAAA,YAAe,SE0Oc,WF1OM,CE0ON,CAxCf,c;UFlMnD,OG8MoD,2BAAL,GAAkB,K;;QH9MxE,W;;MA
EJ,OAAuB,oBAAhB,wBAAgB,C;K;IzBD3B,6B;MAOI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAd,e
AAvB,C;MACV,OAAO,SAAK,M;K;IAGhB,6B;MAOI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAd,e
AAvB,C;MACV,OAAO,SAAK,M;K;IAGhB,6B;MAOI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAd,e
AAvB,C;MACV,OAAO,SAAK,M;K;IAGhB,mC;MAKI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,SAAK,M;K;IAGz

C,mC;MAKI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,SAAK,M;K;IAGzC,mC;MAKI,OAAW,mBAAJ,GAAe,IAAf,
GAAyB,SAAK,M;K;IAGzC,4B;MASI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAd,eAAvB,C;MACV,
OAAO,SAAK,K;K;IAGhB,4B;MASI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAd,eAAvB,C;MACV,O
AAO,SAAK,K;K;IAGhB,4B;MASI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAd,eAAvB,C;MACV,OA
AO,SAAK,K;K;IAGhB,kC;MAOI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,SAAK,K;K;IAGzC,kC;MAOI,OAAW,m
BAAJ,GAAe,IAAf,GAAyB,SAAK,K;K;IAGzC,kC;MAOI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,SAAK,K;K;gFA
GzC,yB;MAAA,mC;MAAA,2C;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;gFAWA,yB;MAAA,mC;
MAAA,2C;MAAA,4B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;gFAWA,yB;MAAA,mC;MAAA,2C;MAAA,4
B;QAQI,OAAO,kBAAO,cAAP,C;O;KARX,C;IAWA,sC;;QAQQ,OAAc,QAAP,MAAO,EAAQ,SAAR,C;;QAChB,
+C;UACE,MAAM,2BAAuB,CAAE,QAAzB,C;;UAHV,O;;K;IAOJ,sC;;QAQQ,OAAc,SAAP,MAAO,EAAS,SAAT
,C;;QAChB,+C;UACE,MAAM,2BAAuB,CAAE,QAAzB,C;;UAHV,O;;K;IAOJ,sC;;QAQQ,OAAiD,OAA1C,MAA
O,iBAAQ,e6BzKgB,I7ByKxB,EAAoB,CAAA,c6BzKI,I7ByKJ,IAAY,CAAZ,IAApB,CAAmC,C;;QACnD,+C;UA
CE,MAAM,2BAAuB,CAAE,QAAzB,C;;UAHV,O;;K;4FAOJ,yB;MAAA,mC;MAAA,uD;MAAA,4B;QAOI,OAA
O,wBAAa,cAAb,C;O;KAPX,C;4FAUA,yB;MAAA,mC;MAAA,uD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;
KAPX,C;4FAUA,yB;MAAA,mC;MAAA,uD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;IAUA,4C;M
AMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAc,QAAP,MAAO,EAAQ,SAAR,C;K;IAGlB,4C;MAMI,IAAI,m
BAAJ,C;QACI,OAAO,I;MACX,OAAc,SAAP,MAAO,EAAS,SAAT,C;K;IAGlB,4C;MAMI,IAAI,mBAAJ,C;QAC
I,OAAO,I;MACX,OAAiD,OAA1C,MAAO,iBAAQ,e6B3OoB,I7B2O5B,EAAoB,CAAA,c6B3OQ,I7B2OR,IAAY,
CAAZ,IAApB,CAAmC,C;K;mFAGrD,8B;MAQI,OAAO,mBAAmB,2BAAS,OAAT,C;K;oFAG9B,8B;MAQI,OA
AO,mBAAmB,2BAAS,OAAT,C;K;oFAG9B,8B;MAQI,OAAO,mBAAmB,2BAAS,OAAT,C;K;IAG9B,uC;MAKI,
OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAKI,OAAO,2BAAe,oBAAN,KAAM,CAAf,C;K;IAGX,uC;MAKI,OAAO,
2BAAe,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,KAAf,C;K;IAG
X,uC;MAOI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,oBAAN,KAAM,CAAf,C;K;IAGX,uC;M
AOI,OAAO,2BAAe,KAAf,C;K;oFAGX,yB;MAAA,6C;MAAA,8B;MAAA,+C;MAAA,mC;QAKY,Q;QAAR,OA
AkC,SAA1B,gEAA0B,EAAS,KAAT,C;O;KALtC,C;oFAQA,yB;MAAA,6C;MAAA,8B;MAAA,+C;MAAA,mC;Q
AKY,Q;QAAR,OAAmC,SAA3B,gEAA2B,EAAS,KAAT,C;O;KALvC,C;IAQA,uC;MiB3SW,SjBkTM,mBAAN,K
AAM,C;MAAb,OAA0C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG5E,uC;MiBrTW,SjB4TM,kBA
AN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG7E,uC;MiB/TW,SjBsUM,
oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG7E,uC;MiBzUW,SjB
gVM,qBAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG9E,uC;MAKI,
OAAO,2BAAe,KAAf,C;K;IAGX,uC;MiB3VW,SjBkWM,mBAAN,KAAM,C;MAAb,OAA0C,UAAJ,GAAgB,2BA
AS,EAAT,CAAhB,GAAkC,K;K;IAG5E,uC;MiBrWW,SjB4WM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAg
B,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG7E,uC;MiB/WW,SjBsXM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,
GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG7E,uC;MiBzXW,SjBgYM,qBAAN,KAAM,C;MAAb,OAA4C,
UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG9E,uC;MAKI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MA
OI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAKI,OAAO,2BAAe,oBAAN,KAAM,CAAf,C;K;IAGX,uC;MiB7ZW,Sj
BkaM,kBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG7E,uC;MiBra
W,SjB0aM,mBAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG9E,uC;
MAOI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,o
BAAN,KAAM,CAAf,C;K;IAGX,uC;MiB3cW,SjBkdM,kBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAA
S,EAAT,CAAhB,GAAkC,K;K;IAG7E,uC;MiBrdW,SjB4dM,mBAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2
BAAS,EAAT,CAAhB,GAAkC,K;K;oFAG9E,yB;MAAA,6C;MAAA,8B;MAAA,+C;MAAA,mC;QAKY,Q;QAAR
,OAAmC,SAA3B,gEAA2B,EAAS,KAAT,C;O;KALvC,C;IAQA,uC;MiBveW,SjB4eM,iBAAN,KAAM,C;MAAb,
OAA0C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG5E,uC;MiB/eW,SjBofM,oBAAN,KAAM,C;MA
Ab,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG7E,uC;MiBvfW,SjB4fM,qBAAN,KAAM,C;
MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG9E,uC;MAOI,OAAO,2BAAS,KAAM,
WAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAS,KAAM,WAAf,C;K;IAGX,uC;MiBnhBW,SjB0hBM,iBAAN,KAAM,
C;MAAb,OAA0C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG5E,uC;MiB7hBW,SjBoiBM,oBAAN,

KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG7E,uC;MiBviBW,SjB8iBM,q
BAAN,KAAM,C;MAAb,OAA4C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;oFAG9E,yB;MAAA,6C;M
AAA,8B;MAAA,+C;MAAA,mC;QAKY,Q;QAAR,OAAkC,SAA1B,gEAA0B,EAAS,KAAT,C;O;KALtC,C;IAQA,
uC;MAKI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAKI,OAAO,2BAAe,oBAAN,KAAM,CAAf,C;K;IAGX,uC;MiB
zkBW,SjB8kBM,oBAAN,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;IAG7E,
uC;MAOI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAA
e,KAAf,C;K;IAGX,uC;MAOI,OAAO,2BAAe,oBAAN,KAAM,CAAf,C;K;IAGX,uC;MiBznBW,SjBgoBM,oBAA
N,KAAM,C;MAAb,OAA2C,UAAJ,GAAgB,2BAAS,EAAT,CAAhB,GAAkC,K;K;oFAG7E,yB;MAAA,6C;MAA
A,8B;MAAA,+C;MAAA,mC;QAKY,Q;QAAR,OAAkC,SAA1B,gEAA0B,EAAS,KAAT,C;O;KALtC,C;oFAQA,y
B;MAAA,6C;MAAA,8B;MAAA,+C;MAAA,mC;QAKY,Q;QAAR,OAAmC,SAA3B,gEAA2B,EAAS,KAAT,C;O;
KALvC,C;IAQA,+B;MAOI,OAAO,sCAAe,yBAAgB,SAAhB,EAAyB,EAAzB,EAAkC,EAAlC,C;K;IAG1B,iC;M
AOI,OAAO,uCAAgB,yBAAgB,SAAhB,EAAyB,oBAAH,EAAG,CAAzB,M;K;IAG3B,iC;MAOI,OAAO,sCAAe,y
BAAqB,SAArB,EAAiC,EAAjC,EAA0C,EAA1C,C;K;IAG1B,iC;MAOI,OAAO,sCAAe,yBAAqB,SAArB,EAAiC,
EAAjC,EAA0C,EAA1C,C;K;IAG1B,iC;MAOI,OAAO,uCAAgB,yBAAgB,SAAhB,EAAsB,EAAtB,EAA0B,EAA1
B,C;K;IAG3B,iC;MAOI,OAAO,sCAAe,yBAAgB,SAAhB,EAAsB,EAAtB,EAA0B,EAA1B,C;K;IAG1B,iC;MAOI
,OAAO,uCAAgB,yBAAgB,SAAhB,EAAyB,oBAAH,EAAG,CAAzB,M;K;IAG3B,iC;MAOI,OAAO,sCAAe,yBA
AqB,SAArB,EAA8B,EAA9B,EAAkC,EAAlC,C;K;IAG1B,iC;MAOI,OAAO,sCAAe,yBAAqB,SAArB,EAA8B,EA
A9B,EAAkC,EAAlC,C;K;IAG1B,iC;MAOI,OAAO,uCAAgB,yBAAqB,oBAAL,SAAK,CAArB,EAA+B,EAA/B,
M;K;IAG3B,iC;MAOI,OAAO,uCAAgB,yBAAgB,SAAhB,EAAsB,EAAtB,M;K;IAG3B,kC;MAOI,OAAO,uCAAg
B,yBAAqB,oBAAL,SAAK,CAArB,EAA+B,EAA/B,M;K;IAG3B,kC;MAOI,OAAO,uCAAgB,yBAAqB,oBAAL,S
AAK,CAArB,EAA+B,EAA/B,M;K;IAG3B,kC;MAOI,OAAO,sCAAe,yBAAgB,SAAhB,EAAyB,EAAzB,EAAkC,
EAAlC,C;K;IAG1B,kC;MAOI,OAAO,uCAAgB,yBAAgB,SAAhB,EAAyB,oBAAH,EAAG,CAAzB,M;K;IAG3B,k
C;MAOI,OAAO,sCAAe,yBAAqB,SAArB,EAAiC,EAAjC,EAA0C,EAA1C,C;K;IAG1B,kC;MAOI,OAAO,sCAAe,
yBAAqB,SAArB,EAAiC,EAAjC,EAA0C,EAA1C,C;K;IAG1B,+B;MAII,OAAO,sCAAe,yBAAgB,cAAhB,EAAsB
,eAAtB,EAA6B,CAAC,cAAD,IAA7B,C;K;IAG1B,gC;MAII,OAAO,uCAAgB,yBAAgB,cAAhB,EAAsB,eAAtB,E
AA8B,cAAD,aAA7B,C;K;IAG3B,gC;MAII,OAAO,uCAAgB,yBAAgB,cAAhB,EAAsB,eAAtB,EAA6B,CAAC,cA
AD,IAA7B,C;K;IAG3B,+B;MAII,oBAAoB,OAAO,CAA3B,EAA8B,IAA9B,C;MACA,OAAO,sCAAe,yBAAgB,e
AAhB,EAAuB,cAAvB,EAAiC,SAAK,KAAL,GAAY,CAAhB,GAAmB,IAAnB,GAA6B,CAAC,IAAD,IAA1D,C;
K;IAG1B,iC;MAII,oBAAoB,kBAAO,CAA3B,EAA8B,IAA9B,C;MACA,OAAO,uCAAgB,yBAAgB,eAAhB,EAA
uB,cAAvB,EAAiC,SAAK,KAAL,cAAY,CAAhB,GAAmB,IAAnB,GAA8B,IAAD,aAA1D,C;K;IAG3B,iC;MAII,o
BAAoB,OAAO,CAA3B,EAA8B,IAA9B,C;MACA,OAAO,uCAAgB,yBAAgB,eAAhB,EAAuB,cAAvB,EAAiC,S
AAK,KAAL,GAAY,CAAhB,GAAmB,IAAnB,GAA6B,CAAC,IAAD,IAA1D,C;K;IAG3B,sC;MACI,OAAmB,IAA
R,8BAAgC,GAApC,GAAiE,OAAL,SAAK,CAAjE,GAA+E,I;K;IAG1F,wC;MACI,OAAW,mEAAJ,GAAmE,OAA
L,SAAK,SAAnE,GAAiF,I;K;IAG5F,wC;MA1OY,Q;MA2OR,OA3OkC,YAA1B,qBA2OW,aAAA,sCAAe,UAAf,E
AA0B,sCAAe,UAAzC,CA3OX,kCAA0B,EA2OvB,SA3OuB,CA2O3B,GAAqE,OAAL,SAAK,CAArE,GAAmF,I;
K;IAG9F,wC;MACI,OAAmB,UAAA,sCAAe,UAAf,EAA2B,sCAAe,UAA1C,CAAR,4BAAJ,GAA+E,OAAR,YA
AL,SAAK,CAAQ,CAA/E,GAA6F,I;K;IAGxG,wC;MACI,OAAmB,UAAA,sCAAe,UAAf,EAA0B,sCAAe,UAAzC
,CAAR,4BAAJ,GAA6E,OAAR,YAAL,SAAK,CAAQ,CAA7E,GAA2F,I;K;IAGtG,qC;MACI,OAAW,iFAAJ,GAA
4D,SAAK,QAAjE,GAA8E,I;K;IAGzF,uC;MACI,OAAmB,UAAc,WAAd,EAAwC,UAAxC,CAAR,4BAAJ,GAAq
E,YAAL,SAAK,CAArE,GAAkF,I;K;IAG7F,uC;MACI,OAAmB,UAAc,WAAd,EAAuC,UAAvC,CAAR,4BAAJ,G
AAmE,YAAL,SAAK,CAAnE,GAAgF,I;K;IAG3F,sC;MACI,OAAmB,UAAe,mCAAf,EAA0C,mCAA1C,CAAR,4
BAAJ,GAAuE,uBAAL,SAAK,CAAvE,GAAqF,I;K;IAGhG,wC;MACI,OAAmB,UAAe,mCAAf,EAAyC,mCAAzC
,CAAR,4BAAJ,GAAqE,uBAAL,SAAK,CAArE,GAAmF,I;K;IAG9F,uC;MACI,OAAmB,MAAR,8BAAiC,KAArC
,GAAmE,QAAL,SAAK,CAAnE,GAAkF,I;K;IAG7F,yC;MACI,OAAW,uEAAJ,GAAqE,QAAL,SAAK,SAArE,GA
AoF,I;K;IAG/F,yC;MACI,OAAmB,UAAA,uCAAgB,UAAhB,EAA4B,uCAAgB,UAA5C,CAAR,4BAAJ,GAAiF,Q
AAR,YAAL,SAAK,CAAQ,CAAjF,GAAgG,I;K;IAG3G,yC;MACI,OAAmB,UAAA,uCAAgB,UAAhB,EAA2B,uC
AAgB,UAA3C,CAAR,4BAAJ,GAA+E,QAAR,YAAL,SAAK,CAAQ,CAA/E,GAA8F,I;K;IAGzG,8B;MAMI,OAA
O,wBAAY,EAAa,GAAH,CAAG,IAAzB,C;K;IAGX,gC;MAMI,OAAO,kBAAY,oBAAH,EAAG,CAAc,8BAAH,C

AAG,EAA1B,C;K;IAGX,gC;MAMI,OAAO,aAAK,SAAL,EAAoB,EAAa,GAAH,CAAG,IAAjC,C;K;IAGX,gC;MAMI,OAAO,aAAK,SAAL,EAAoB,EAAa,GAAH,CAAG,IAAjC,C;K;IAGX,gC;MAMI,IAAI,MAAM,CAAV,C;QAAoB,OAAO,iCAAU,M;MACrC,OAAO,yBAAiB,OAAR,EAAQ,GAAH,CAAG,CAAjB,C;K;IAGX,gC;MAMI,IAAI,MAAM,WAAV,C;QAAyB,OAAO,gCAAS,M;MACzC,OAAO,wBAAS,EAAQ,GAAH,CAAG,IAAjB,C;K;IAGX,gC;MAMI,OAAO,kBAAY,oBAAH,EAAG,CAAc,8BAAH,CAAG,EAA1B,C;K;IAGX,gC;MAMI,IAAI,MAAM,WAAV,C;QAAyB,OAAO,gCAAS,M;MACzC,OAAO,aAAK,SAAL,EAAiB,EAAQ,GAAH,CAAG,IAAzB,C;K;IAGX,gC;MAMI,IAAI,MAAM,WAAV,C;QAAyB,OAAO,gCAAS,M;MACzC,OAAO,aAAK,SAAL,EAAiB,EAAQ,GAAH,CAAG,IAAzB,C;K;IAGX,gC;MAMI,IAAI,iDAAJ,C;QAA0B,OAAO,iCAAU,M;MAC3C,OAAY,oBAAL,SAAK,CAAL,SAAkB,EAAQ,8BAAH,CAAG,EAA1B,C;K;IAGX,gC;MAMI,IAAI,iDAAJ,C;QAA0B,OAAO,iCAAU,M;MAC3C,OAAO,kBAAS,EAAQ,8BAAH,CAAG,EAAjB,C;K;IAGX,iC;MAMI,IAAI,iDAAJ,C;QAA0B,OAAO,iCAAU,M;MAC3C,OAAY,oBAAL,SAAK,CAAL,SAAkB,EAAQ,8BAAH,CAAG,EAA1B,C;K;IAGX,iC;MAMI,IAAI,iDAAJ,C;QAA0B,OAAO,iCAAU,M;MAC3C,OAAY,oBAAL,SAAK,CAAL,SAAkB,EAAQ,8BAAH,CAAG,EAA1B,C;K;IAGX,iC;MAMI,OAAO,wBAAY,EAAa,GAAH,CAAG,IAAzB,C;K;IAGX,iC;MAMI,OAAO,kBAAY,oBAAH,EAAG,CAAc,8BAAH,CAAG,EAA1B,C;K;IAGX,iC;MAMI,OAAO,aAAK,SAAL,EAAoB,EAAa,GAAH,CAAG,IAAjC,C;K;IAGX,iC;MAMI,OAAO,aAAK,SAAL,EAAoB,EAAa,GAAH,CAAG,IAAjC,C;K;IAGX,gD;MAQI,OAAW,4BAAO,YAAP,KAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,kD;MAQI,OAAW,YAAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,kD;MAQI,OAAW,YAAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,kD;MAQI,OAAW,YAAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,kD;MAQI,OAAW,0BAAO,YAAP,KAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,kD;MAQI,OAAW,YAAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,kD;MAQI,OAAW,YAAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,+C;MAQI,OAAW,4BAAO,YAAP,KAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAQI,OAAW,YAAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAQI,OAAW,YAAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAQI,OAAW,YAAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAQI,OAAW,0BAAO,YAAP,KAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAQI,OAAW,YAAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAQI,OAAW,YAAO,YAAX,GAAyB,YAAzB,GAA2C,S;K;IAGtD,yD;MAQI,IAAI,iBAAiB,IAAjB,IAAyB,iBAAiB,IAA9C,C;QACI,IAAI,+BAAe,YAAf,KAAJ,C;UAAiC,MAAM,gCAAyB,6DAAiD,YAAjD,wCAAoF,YAApF,OAAzB,C;QACvC,IAAI,4BAAO,YAAP,KAAJ,C;UAAyB,OAAO,Y;QAChC,IAAI,4BAAO,YAAP,KAAJ,C;UAAyB,OAAO,Y;;QAGhC,IAAI,iBAAiB,IAAjB,IAAyB,4BAAO,YAAP,KAA7B,C;UAAkD,OAAO,Y;QACzD,IAAI,iBAAiB,IAAjB,IAAyB,4BAAO,YAAP,KAA7B,C;UAAkD,OAAO,Y;;MAE7D,OAAO,S;K;IAGX,2D;MAQI,IAAI,eAAe,YAAnB,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MAChC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MAChC,OAAO,S;K;IAGX,2D;MAQI,IAAI,eAAe,YAAnB,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MAChC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MAChC,OAAO,S;K;IAGX,2D;MAQI,IAAI,eAAe,YAAnB,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MAChC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MAChC,OAAO,S;K;IAGX,2D;MAQI,IAAI,6BAAe,YAAf,KAAJ,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,yCAAoF,YAApF,iBAAzB,C;MACvC,IAAI,0BAAO,YAAP,KAAJ,C;QAAyB,OAAO,Y;MAChC,IAAI,0BAAO,YAAP,KAAJ,C;QAAyB,OAAO,Y;MAChC,OAAO,S;K;IAGX,2D;MAQI,IAAI,eAAe,YAAnB,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MAChC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MAChC,OAAO,S;K;IAGX,2D;MAQI,IAAI,eAAe,YAAnB,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MAChC,IAAI,YAAO,YAAX,C;QAAyB,OAAO,Y;MAChC,OAAO,S;K;IAGX,sC;MAUW,Q;MADP,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,4CAAyC,KAAzC,MAAzB,C;MACvB,IAAA,KAAM,0BAAiB,SAAjB,EAAuB,KAAM,MAA7B,CAAN,IAA6C,CAAC,KAAM,0BAAiB,KAAM,MAAvB,EAA8B,SAA9B,CAApD,C;QAAiG,OAAN,KAAM,M;WAEjG,IAAA,KAAM,0BAAiB,KAAM,aAAvB,EAA6B,SAArC,CAAN,IAAoD,CAAC,KAAM,0BAAiB,SAAjB,EAAuB,KAAM,aAA7B,CAA3D,C;QAA+G,OAAN,KAAM,a;;QAAC,gB;MAA5H,W;K;IASJ,sC;MAYW,Q;MAJP,IAAI,8CAAJ,C;QACI,OAAY,WAAL,SAAK,EAAY,KAAZ,C;;MAEhB,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,4CAAyC,KAAzC,MAAzB,C;MAEvB,IAAA,4BAAO,gCAAP,C;QAAiC,gCAAN,KAAM,MAAN,C;WAC3B,IAAA,4BAAO,gCAAP,C;QAAiC,gCAAN,KAAM,aAAN,C;QAAwB,gB;MAAvD,W;K;IAGJ,sC;MAaW,Q;MAJP,IAAI,8CAAJ,C;QACI,OAAY,WAAL,SAAK,EAAY,KAAZ,C;;MAEhB,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,4CAAyC,KAAzC,MAAzB,C;MAEvB,IAAA,4BAAO,gCAAP,C;QAAiC,gCAAN,KAAM,MAAN,C;WAC3B,IAAA,4BAAO,gCAAP,C;QAAiC,gCAAN,KAAM,aAAN,C;QAAwB,gB;MAAvD,W;K;IAGJ,sC;MAYW,Q;MAJP,IAAI,8CAAJ,C;QACI,OAAY,WAAL,SAAK,EAAY,KAAZ,C;;MAEhB,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,4CAAyC,KAAzC,MAAzB,C;MAEvB,IAAA,4BAAO,gCAAP,C;QAAiC,gCAAN,KAAM,MAAN,C;WAC3B,IAAA,4BAAO,gCAAP,C;QAAiC,gCAAN,KAAM,aAAN,C;QAAwB,gB;MAAvD,W;K;IAGJ,wC;MAOI,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,4CAAyC,KAAzC,MAAzB,C;MACvB,OAAW,OAAO,KAAM,MAAjB,GAAwB,KAAM,MAA9B,GAA4C,OAAO,KAAM,aAAjB,GAA+B,KAAM,aAArC,GAAmD,S;K;IAGtG,wC;MAOI,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,4CAAyC,KAAzC,MAAzB,C;MACvB,OAAW,4BAAO,KAAM,MAAb,KAAJ,GAAwB,KAAM,MAA9B,GAA4C,4BAAO,KAAM,aAAb,KAAJ,GAA+B,KAAM,aAArC,GAAmD,S;K;IAGtG,yB;MAMI,OAAO,2BAAY,0BAAW,WAAW,CAAC,GAAZ,CAAX,C;K;IAGvB,kC;MAQI,IAAI,CAAC,qBAAL,C;QAAqB,OAAO,S;MAE5B,OAAO,K;K;IAGX,oC;MAMI,IAAI,CAAC,qBAAL,C;QAAqB,OAAO,S;MAE5B,OAAO,K;K;IAGX,uC;MAMI,IAAI,CAAC,qBAAL,C;QAAqB,OAAO,S;MAE5B,OAAO,K;K;IAGX,uB;MASI,IAAI,cAAc,SAAd,IAAsB,mBAAmB,eAAzC,C;QAAqD,OAAO,S;;QAChD,MAAM,gCAAyB,mBAAY,GAAZ,YAAzB,C;K;IAGX,6B;MAMI,OAAO,8BAAe,+BAAf,6BAAyC,CAA7C,GAAyD,IAAzD,GAAmE,SAAK,iBAAN,GAAsB,aAA9F,E;K;IAGtD,6B;MASI,IAAI,cAAc,SAAlB,C;QAA4B,OAAO,S;;QACnC,OAAO,8BAAe,+BAAf,E;QAA2C,CAAvD,GAAmE,IAAnE,GAA6E,SAAK,iBAAN,GAAsB,aAAtB,EAAqC,iBAAjI,C;K;IAGJ,+B;MAMI,OAAO,kBAAmB,+BAAnB,C;K;IAGX,yC;MASW,OAAA,yBAAsB,+BAAtB,C;K;IAGX,kD;MAaI,OAAO,wBAAiB,0BAAjB,EAA6C,0BAA7C,EAAyE,C;K;IAGpF,qC;MAWI,OAAO,2BAAyC,+BAAzC,C;K;IAGX,+C;MASW,OAAA,2BAAwB,+BAAxB,C;K;IAGX,kD;MAYI,OAAO,wBAA0B,yBAA1B,EAAsC,0BAAtC,EAA0D,C;K;IAGrE,sC;MAWI,OAAO,cAAO,yBAAP,IAAiB,C;K;IAG5B,wC;MAWI,OAAO,cAAM,yBAAN,EAAe,C;K;IAG1B,wC;MAWI,OAAO,CAAC,cAAO,yBAAP,IAAiB,C;K;IAG7B,+C;MAUW,qBAAI,SAAJ,C;K;IAGX,sD;MAUW,+BAAI,SAAJ,C;K;IAGX,oD;MAWI,OAAO,cAAO,yBAAP,IAAiB,C;K;IAG5B,2D;MAWI,OAAO,cAAM,yBAAN,EAAe,C;K;IAG1B,yC;MAMI,IAAI,CAAC,qBAAL,C;QAAqB,OAAO,S;MAE5B,OAAO,K;K;IAGX,2C;MAMI,IAAI,CAAC,qBAAL,C;QAAqB,OAAO,S;MAE5B,OAAO,K;K;IAGX,0C;MAMI,IAAI,CAAC,qBAAL,C;QAAqB,OAAO,S;MAE5B,OAAO,K;K;IAGX,yC;MAMI,IAAI,CAAC,qBAAL,C;QAAqB,OAAO,S;MAE5B,OAAO,K;K;IAGX,uC;MAMI,IAAI,CAAC,qBAAL,C;QAAqB,OAAO,S;MAE5B,OAAO,K;K;IAGX,yB;MAMI,OAAO,sBAAY,SAAZ,EAAkB,CAAlB,EAAqB,OAAO,IAA5B,E;K;IAGX,+C;MAQI,OAAO,+BAAe,SAAf,EAAqB,IAArB,E;K;IAGX,2D;MAQI,OAAO,yBAAqB,SAArB,EAA2B,CAA3B,C;K;IAGX,uC;MAQI,OAAO,kBAAc,SAAd,EAAoB,CAApB,EAAuB,IAAvB,C;K;IAGX,uC;MAQI,OAAO,kBAAc,SAAd,EAAoB,CAApB,EAAuB,IAAvB,C;K;IAGX,uC;MAQI,OAAO,kBAAc,SAAd,EAAoB,CAApB,EAAuB,IAAvB,C;K;IAGX,2C;MAQI,OAAO,kBAAc,SAAd,EAAoB,CAApB,EAAuB,IAAvB,C;K;IAGX,uC;MAQI,OAAO,kBAAc,SAAd,EAAoB,CAApB,EAAuB,IAAvB,C;K

C,OAAN,KAAM,a;;QAC3B,gB;MAHZ,W;K;IAOJ,sC;MAYW,Q;MAJP,IAAI,8CAAJ,C;QACI,OAAY,WAAL,SA
AK,EAAc,KAAd,C;;MAEhB,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,4CAAyC,KAAzC,MAAzB,C;M
AEvB,gBAAO,KAAM,MAAb,C;QAA4B,OAAN,KAAM,M;WAC5B,gBAAO,KAAM,aAAb,C;QAAmC,OAAN,
KAAM,a;;QAC3B,gB;MAHZ,W;K;IAOJ,sC;MAYW,Q;MAJP,IAAI,8CAAJ,C;QACI,OAAY,WAAL,SAAK,EAA
e,KAAf,C;;MAEhB,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,4CAAyC,KAAzC,MAAzB,C;MAEvB,8B
AAO,KAAM,MAAb,M;QAA4B,OAAN,KAAM,M;WAC5B,8BAAO,KAAM,aAAb,M;QAAmC,OAAN,KAAM,a;
;QAC3B,gB;MAHZ,W;K;IYh/CJ,oD;MAMuF,wC;K;IANvF,8CAOI,Y;MAAuC,8B;K;IAP3C,gF;IkBQA,yC;MAM
I,OAAO,sBAAQ,OAAR,KAAoB,C;K;IAWG,2C;MAAA,qB;QAAE,MAAM,8BAA0B,+CAA4C,aAA5C,MAA1B,
C;O;K;IAR1C,uC;MAQI,OAAO,8BAAgB,KAAhB,EAAuB,yBAAvB,C;K;IAGX,4D;MAcqB,Q;MANjB,IAAI,QA
AQ,CAAZ,C;QACI,OAAO,aAAa,KAAb,C;MACX,eAAe,oB;MACf,YAAY,C;MACZ,OAAO,QAAS,UAAhB,C;Q
ACI,cAAc,QAAS,O;QACvB,IAAI,WAAS,YAAT,EAAS,oBAAT,OAAJ,C;UACI,OAAO,O;;MAEf,OAAO,aAAa,
KAAb,C;K;IAGX,8C;MAcqB,Q;MANjB,IAAI,QAAQ,CAAZ,C;QACI,OAAO,I;MACX,eAAe,oB;MACf,YAAY,
C;MACZ,OAAO,QAAS,UAAhB,C;QACI,cAAc,QAAS,O;QACvB,IAAI,WAAS,YAAT,EAAS,oBAAT,OAAJ,C;
UACI,OAAO,O;;MAEf,OAAO,I;K;8EAGX,gC;MASW,sB;;QA4FS,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UA
AgB,yB;UAAM,IA5FH,SA4FO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MA
7FP,yB;K;uFAGJ,gC;MAmOoB,Q;MADhB,WAAe,I;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,I
A3Nc,SA2NV,CAAU,OAAV,CAAJ,C;UACI,OAAO,O;;;MA5Nf,OA+NO,I;K;IA5NX,6B;MAQI,eAAe,oB;MACf,
IAAI,CAAC,QAAS,UAAd,C;QACI,MAAM,2BAAuB,oBAAvB,C;MACV,OAAO,QAAS,O;K;iFAGpB,yB;MAA
A,iE;MAAA,uC;QAOoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAA
J,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,sDAAvB,C;O;KARV,C;kGAWA,yB;MAAA,iE;MAAA,uC;QA
W8C,IAAnC,I;QAAA,+B;;UAcS,U;UAAA,6B;UAAhB,OAAgB,gBAAhB,C;YAAgB,2B;YACZ,aAfwB,SAeX,CA
AU,OAAV,C;YACb,IAAI,cAAJ,C;cACI,8BAAO,M;cAAP,gC;;;UAGR,8BAAO,I;;;QApBA,kC;QAAA,iB;UAAm
C,MAAM,gCAAuB,iEAAvB,C;;QAAhD,OAAO,I;O;KAXX,C;8GAcA,gC;MAWoB,Q;MAAA,2B;MAAhB,OAA
gB,cAAhB,C;QAAgB,yB;QACZ,aAAa,UAAU,OAAV,C;QACb,IAAI,cAAJ,C;UACI,OAAO,M;;;MAGf,OAAO,I;
K;IAGX,mC;MAMI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QACI,OAAO,I;MACX,OAAO,QAAS,O;K;6F
AGpB,gC;MAMoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;
UAAwB,OAAO,O;;MACrD,OAAO,I;K;IAGX,wC;MAOiB,Q;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C
;QAAa,sB;QACT,mBAAmB,KAAnB,C;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UACI,OAAO,K;QACX,qB;;MAEJ,
OAAO,E;K;+FAGX,yB;MAAA,wE;MAAA,uC;QAOiB,Q;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UA
Aa,sB;UACT,mBAAmB,KAAnB,C;UACA,IAAI,UAAU,IAAV,CAAJ,C;YACI,OAAO,K;UACX,qB;;QAEJ,OAA
O,E;O;KAbX,C;6FAgBA,yB;MAAA,wE;MAAA,uC;QAQiB,Q;QAFb,gBAAgB,E;QAChB,YAAY,C;QACC,2B;Q
AAb,OAAa,cAAb,C;UAAa,sB;UACT,mBAAmB,KAAnB,C;UACA,IAAI,UAAU,IAAV,CAAJ,C;YACI,YAAY,K;
UAChB,qB;;QAEJ,OAAO,S;O;KAdX,C;IAiBA,4B;MAUI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QACI,
MAAM,2BAAuB,oBAAvB,C;MACV,WAAW,QAAS,O;MACpB,OAAO,QAAS,UAAhB,C;QACI,OAAO,QAAS,
O;MACpB,OAAO,I;K;+EAGX,yB;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAYoB,UAQT,M;QAVP,WAAe
,I;QACf,YAAY,K;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,
OAAO,O;YACP,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAY,MAAM,gCAAuB,sDAAvB,C;QAElB,OAAO,2
E;O;KApBX,C;IAuBA,4C;MAQiB,Q;MAFb,gBAAgB,E;MAChB,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;Q
AAa,sB;QACT,mBAAmB,KAAnB,C;QACA,IAAI,gBAAW,IAAX,CAAJ,C;UACI,YAAY,K;QAChB,qB;;MAEJ,
OAAO,S;K;IAGX,kC;MAQI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QACI,OAAO,I;MACX,WAAW,QAA
S,O;MACpB,OAAO,QAAS,UAAhB,C;QACI,OAAO,QAAS,O;MACpB,OAAO,I;K;2FAGX,gC;MASoB,Q;MADh
B,WAAe,I;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,OAAO
,O;;;MAGf,OAAO,I;K;IAGX,8B;MAMI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QACI,MAAM,2BAAuB,o
BAAvB,C;MACV,aAAa,QAAS,O;MACtB,IAAI,QAAS,UAAb,C;QACI,MAAM,gCAAyB,qCAAzB,C;MACV,OA
AO,M;K;mFAGX,yB;MAAA,kF;MAAA,iE;MAAA,gB;MAAA,8B;MAAA,uC;QAQoB,UAST,M;QAXP,aAAAiB,I;
QACjB,YAAY,K;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,I
AAI,KAAJ,C;cAAW,MAAM,8BAAyB,mDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KA
AL,C;UAAY,MAAM,gCAAuB,sDAAvB,C;QAElB,OAAO,6E;O;KAjBX,C;IAoBA,oC;MAMI,eAAe,oB;MACf,I

AAI,CAAC,QAAS,UAAd,C;QACI,OAAO,I;MACX,aAAa,QAAS,O;MACtB,IAAI,QAAS,UAAb,C;QACI,OAAO,
I;MACX,OAAO,M;K;+FAGX,gC;MAQoB,Q;MAFhB,aAAiB,I;MACjB,YAAY,K;MACI,2B;MAAhB,OAAgB,cA
AhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UAClB,SAAS,O;
UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;IAGX,8B;MAWW,Q;Mh
BjXP,IAAI,EgBgXI,KAAK,ChBhXT,CAAJ,C;QACI,cgB+Wc,sD;QhB9Wd,MAAM,gCAAyB,OAAQ,WAAjC,C;;
MgBgXN,UAAK,CAAL,C;QAAU,gB;WACV,+C;QAAiC,OAAL,SAAK,cAAK,CAAL,C;;QACzB,wBAAa,SAAb
,EAAmB,CAAnB,C;MAHZ,W;K;IAOJ,2C;MAQI,OAAO,sBAAkB,SAAlB,EAAwB,SAAxB,C;K;IAGX,wC;MAQ
I,OAAO,sBAAkB,SAAlB,EAAwB,IAAxB,EAA8B,SAA9B,C;K;IAcqE,iD;MAAA,qB;QAAE,yBAAU,EAAG,MA
Ab,EAAoB,EAAG,MAAvB,C;O;K;IAAkC,oC;MAAE,OAAA,EAAG,M;K;IAXzH,+C;MAWI,OAAO,yBAAqB,s
BAAkB,qBAAiB,SAAjB,CAAlB,EAA0C,IAA1C,EAAgD,+BAAhD,CAArB,EAAyG,sBAAzG,C;K;oGAGX,yB;
MAk1BA,wE;MAl1BA,oD;QA21BiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAh1BT,I
AAI,UAg1BkB,oBAAmB,cAAnB,EAAmB,sBAAnB,UAh1BlB,EAg1B+C,IAh1B/C,CAAJ,C;YAA2C,sBAg1BQ,I
Ah1BR,C;;QAE/C,OAAO,W;O;KAbX,C;sGAgBA,yB;MAAA,8C;MAAA,0C;MAAA,8B;MASkB,qD;QAAA,qB;
UAAE,c;S;O;MATpB,sC;QASW,Q;QAAP,OAAO,uCAAO,iCAAP,gC;O;KATX,C;0GAYA,4C;MAQoB,Q;MAA
A,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,YAAJ,C;UAAkB,WAAY,WAAI,OAAJ,C;;MACpD,
OAAO,W;K;IAGX,2C;MAQI,OAAO,sBAAkB,SAAlB,EAAwB,KAAxB,EAA+B,SAA/B,C;K;IAYU,kC;MAAE,i
B;K;IATvB,oC;MASW,Q;MAAP,OAAO,4CAAU,oBAAV,kC;K;IAGX,mD;MAQoB,Q;MAAA,2B;MAAhB,OAA
gB,cAAhB,C;QAAgB,yB;QAAM,IAAI,eAAJ,C;UAAqB,WAAY,WAAI,OAAJ,C;;MACvD,OAAO,W;K;4FAGX,
6C;MAQoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;
UAAyB,WAAY,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;sFAGX,6C;MAQoB,Q;MAAA,2B;MAAhB,OAAgB,cAA
hB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;I
AGX,8B;MAWW,Q;MhB1gBP,IAAI,EgBygBI,KAAK,ChBzgBT,CAAJ,C;QACI,cgBwgBc,sD;QhBvgBd,MAAM,
gCAAyB,OAAQ,WAAjC,C;;MgBygBN,UAAK,CAAL,C;QAAU,sB;WACV,+C;QAAiC,OAAL,SAAK,cAAK,CA
AL,C;;QACzB,wBAAa,SAAb,EAAmB,CAAnB,C;MAHZ,W;K;IAOJ,2C;MAQI,OAAO,sBAAkB,SAAlB,EAAwB
,SAAxB,C;K;IAWA,2C;MAAA,8B;K;8CACH,Y;MACI,iBAA6B,iBAAZ,gBAAY,C;MAClB,QAAX,UAAW,C;M
ACX,OAAO,UAAW,W;K;;IAZ9B,6B;MAQI,0C;K;sFASJ,yB;MAAA,sD;MdlfA,sC;MAAA,oC;MAAA,uBAOe,y
B;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,S
A4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;Mc2ef,sC;QAUI,OAAO,sBdrfP,eAAW,iBcqfiB,QdrfjB,CAAX,CcqfO,C
;O;KAVX,C;0GAaA,yB;MAAA,sD;Md5eA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;UAAA,uB;Y
AAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;W;S;OA+EI,
C;Mcqef,sC;QAQI,OAAO,sBd7eP,eAAW,2Bc6e2B,Qd7e3B,CAAX,Cc6eO,C;O;KARX,C;IAWA,uC;MAQI,OAA
O,wBAAW,cAAX,C;K;IAWA,uE;MAAA,sC;MAAA,4C;K;kDACH,Y;MACI,iBAAiC,iBAAhB,oBAAgB,C;MAC
tB,WAAX,UAAW,EAAS,uBAAT,C;MACX,OAAO,UAAW,W;K;;IAZ9B,6C;MAQI,0D;K;wFASJ,yB;MAAA,wE
;MAAA,uC;QAaW,kBAAY,oB;QAiFH,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAlFsC,SA
kFvB,CAAU,OAAV,C;UzBhEnB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;QyBlBA,OAoFO,W;O;KAjG
X,C;6FAgBA,yB;MAAA,wE;MAAA,yC;QAaW,kBAAc,oB;QA8BL,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UA
AgB,yB;UACZ,WAAY,aA/B4B,WA+BxB,CAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;QA/BhB,OAiCO,W;O;KA9
CX,C;6FAgBA,yB;MAAA,wE;MAAA,yD;QAYW,kBAAc,oB;QAiCL,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;U
AAgB,yB;UACZ,WAAY,aAlC4B,WAkCxB,CAAY,OAAZ,CAAJ,EAlCyC,cAkCf,CAAe,OAAf,CAA1B,C;;QAlC
hB,OAoCO,W;O;KAhDX,C;iGAeA,+C;MAYoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,W
AAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,OAA1B,C;;MAEhB,OAAO,W;K;iGAGX,+D;MAYoB,Q;MAAA,2B;M
AAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,WAAY,aAAI,YAAY,OAAZ,CAAJ,EAA0B,eAAe,OAAf,CAA1B,C;
;MAEhB,OAAO,W;K;4FAGX,6C;MAWoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,WAAe,
UAAU,OAAV,C;QzBhEnB,wBAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;MyBkEA,OAAO,W;K;gGAGX,yB;
MAAA,wE;MAAA,2C;QAcI,aAAa,oB;QAgBG,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAfO,MAg
BP,aAAI,OAAJ,EAhBe,aAgBF,CAAc,OAAd,CAAb,C;;QAhBhB,OAAuB,M;O;KAf3B,C;oGAkBA,iD;MAYoB,Q
;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;M
AEhB,OAAO,W;K;IAGX,gD;MAMiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAY,WAAI,IAA

J,C;;MAEhB,OAAO,W;K;IAGX,gC;MAMI,OAAO,0BAAa,cAAb,C;K;IAGX,8B;MAMI,OAA4B,qBAAhB,iBAA
L,SAAK,CAAgB,C;K;IAGhC,qC;MAMI,OAAO,0BAAa,gBAAb,C;K;IAGX,4B;MAQI,OAAwC,oBAAjC,0BAAa
,sBAAb,CAAiC,C;K;IAG5C,0C;MAYI,OAAO,uBAAmB,SAAnB,EAAyB,SAAzB,6BAAoC,qB;;OAApC,E;K;IA
GX,0C;MAQI,OAAO,uBAAmB,SAAnB,EAAyB,SAAzB,6BAAoC,qB;;OAApC,E;K;IAGX,iD;MAaI,OAAO,kBA
Ae,SAAf,EAAqB,SAArB,6BAAgC,qB;;OAAhC,E;K;IAGX,iD;MAaI,OAAO,kBAAe,SAAf,EAAqB,SAArB,6BA
AgC,qB;;OAAhC,E;K;sGAGX,yB;MAAA,wE;MAAA,gD;MAAA,oD;QAaoB,UAC4B,M;QAF5C,YAAY,C;QACI
,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,oBAAmB,cAAnB,EAAmB,sBAAnB,UAAV,
EAAuC,OAAvC,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAjBX,C;uGAoBA,yB;MAAA,
wE;MAAA,gD;MAAA,oD;QAaoB,UAC4B,M;QAF5C,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,y
B;UACZ,WAAW,UAAU,oBAAmB,cAAnB,EAAmB,sBAAnB,UAAV,EAAuC,OAAvC,C;UACC,OAAZ,WAAY,
EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAjBX,C;yFAoBA,yB;MAAA,gD;MAAA,oD;QAUoB,Q;QAAA,2B;QAAh
B,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEh
B,OAAO,W;O;KAdX,C;yFAiBA,yB;MAAA,gD;MAAA,oD;QAMoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;U
AAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAVX,
C;qFAaA,yB;MAAA,wE;MA6BA,+D;MA7BA,yC;QAWW,kBAAU,oB;QA6BD,Q;QAAA,2B;QAAhB,OAAgB,c
AAhB,C;UAAgB,yB;UACZ,UA9BiD,WA8BvC,CAAY,OAAZ,C;UzB9nBP,U;UADP,YyBgoBe,WzBhoBH,WyB
goBwB,GzBhoBxB,C;UACL,IAAI,aAAJ,C;YACH,ayB8nBuC,gB;YAA5B,WzB7nBX,ayB6nBgC,GzB7nBhC,EA
AS,MAAT,C;YACA,e;;YAEA,c;;UyB0nBA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAhCT,OAkCO,W;O;KA7CX,C;q
FAcA,yB;MAAA,wE;MAkCA,+D;MAlCA,yD;QAYW,kBAAU,oB;QAkCD,Q;QAAA,2B;QAAhB,OAAgB,cAAh
B,C;UAAgB,yB;UACZ,UAnCiD,WAmCvC,CAAY,OAAZ,C;UzBlpBP,U;UADP,YyBopBe,WzBppBH,WyBopB
wB,GzBppBxB,C;UACL,IAAI,aAAJ,C;YACH,ayBkpBuC,gB;YAA5B,WzBjpBX,ayBipBgC,GzBjpBhC,EAAS,M
AAT,C;YACA,e;;YAEA,c;;UyB8oBA,iB;UACA,IAAK,WArCyD,cAqCrD,CAAe,OAAf,CAAJ,C;;QArCT,OAuC
O,W;O;KAnDX,C;yFAeA,yB;MAAA,+D;MAAA,sD;QAWoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,
yB;UACZ,UAAU,YAAY,OAAZ,C;UzB9nBP,U;UADP,YyBgoBe,WzBhoBH,WyBgoBwB,GzBhoBxB,C;UACL,I
AAI,aAAJ,C;YACH,ayB8nBuC,gB;YAA5B,WzB7nBX,ayB6nBgC,GzB7nBhC,EAAS,MAAT,C;YACA,e;;YAEA,
c;;UyB0nBA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAhBX,C;yFAmBA,yB;MAAA,+D;MAAA,s
E;QAYoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;UzBlpBP,U;UA
DP,YyBopBe,WzBppBH,WyBopBwB,GzBppBxB,C;UACL,IAAI,aAAJ,C;YACH,ayBkpBuC,gB;YAA5B,WzBjp
BX,ayBipBgC,GzBjpBhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UyB8oBA,iB;UACA,IAAK,WAAI,eAAe,OAAf,C
AAJ,C;;QAET,OAAO,W;O;KAjBX,C;0FAoBA,yB;MAAA,kC;MAAA,4C;MAAA,wE;QAUW,sC;QAAA,8C;O;
MAVX,oDAWQ,Y;QAA6C,OAAA,oBAAgB,W;O;MAXrE,iDAYQ,mB;QAAoC,gCAAY,OAAZ,C;O;MAZ5C,gF
;MAAA,yC;QAUI,2D;O;KAVJ,C;IAgBA,sC;MASI,OAAO,yBAAqB,SAArB,EAA2B,SAA3B,C;K;IAGX,4C;MA
SI,OAAO,gCAA4B,SAA5B,EAAkC,SAAlC,C;K;IAGX,mD;MASI,OAAoD,gBAA7C,gCAA4B,SAA5B,EAAkC,S
AAlC,CAA6C,C;K;4GAGxD,yB;MA2NA,wE;MA3NA,oD;QAoOiB,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa
,cAAb,C;UAAa,sB;UA3NsB,U;UAAAA,wBA2NT,oBAAmB,cAAnB,EAAmB,sBAAnB,UA3NS,EA2NoB,IA3NpB,
W;YAA6C,6B;;;QAChF,OAAO,W;O;KAVX,C;8FAaA,yB;MAAA,wE;MAAA,oD;QAUiB,UACoC,M;QAFjD,YA
AY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAY,WAAI,UAAU,oBAAmB,cAAnB,EAAmB,sBA
AnB,UAAV,EAAuC,IAAvC,CAAJ,C;;QAChB,OAAO,W;O;KAZX,C;IAeA,4C;MASI,OAA6C,gBAAtC,yBAAqB
,SAArB,EAA2B,SAA3B,CAAsC,C;K;8FAGjD,yB;MAAA,oD;QAgLoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;
UAAgB,yB;UAzKK,U;UAAA,wBAAyKQ,OAzKR,W;YAAsC,6B;;;QAC3D,OAAO,W;O;KARX,C;iFAWA,6C;MA
OiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAY,WAAI,UAAU,IAAV,CAAJ,C;;MAChB,OAA
O,W;K;IAGX,gC;MAOI,OAAO,qBAAiB,SAAjB,C;K;IAcgB,6B;MAAE,S;K;IAX7B,+B;MAWI,OAAY,aAAL,SA
AK,EAAW,eAAX,C;K;IAGhB,2C;MAYI,OAAO,qBAAiB,SAAjB,EAAuB,QAAvB,C;K;IAGX,mC;MASiB,Q;M
ADb,UAAU,sB;MACG,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,GAAI,WAAI,IAAJ,C;;MACvB,OAAO,G;K;
6EAGX,gC;MAYoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,C
AAL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;IAGX,2B;MAQI,OAAO,oBAAW,U;K;6EAGtB,gC;MAQoB,Q;M
AAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACr
D,OAAO,K;K;IAGX,6B;MAOoB,Q;MADhB,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAA

M,oBAAmB,qBAAnB,EAAmB,KAAnB,E;;MACtB,OAAO,K;K;iFAGX,yB;MAAA,wE;MAAA,uC;QAOoB,Q;Q
ADhB,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB
,oBAAmB,qBAAnB,EAAmB,KAAnB,E;;QAC9C,OAAO,K;O;KARX,C;8EAWA,yC;MAYoB,Q;MADhB,kBAAk
B,O;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACp
C,OAAO,W;K;4FAGX,yB;MAAA,wE;MAAA,gD;QAcoB,UAAiD,M;QAFjE,YAAY,C;QACZ,kBAAkB,O;QACF
,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,cAAc,UAAU,oBAAmB,cAAnB,EAAmB,sBAAnB,UAAV,E
AAuC,WAAvC,EAAoD,OAApD,C;;QACpC,OAAO,W;O;KAfX,C;qFAkBA,6B;MAMoB,Q;MAAA,2B;MAAhB,
OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;K;kGAG1B,yB;MAAA,wE;MAAA,oC;QASiB,UAAgC,
M;QAD7C,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAAM,OAAO,oBAAmB,cAAnB,EAAmB,sBA
AnB,UAAP,EAAoC,IAApC,C;;O;KATvB,C;IAYA,2B;MAaI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAA
yB,MAAM,6B;MAC/B,UAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZh
uCG,MAAO,KYguCE,GZhuCF,EYguCO,CZhuCP,C;;MYkuCd,OAAO,G;K;IAGX,2B;MAaI,eAAe,oB;MACf,IA
AI,CAAC,QAAS,UAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QAC
I,QAAQ,QAAS,O;QACjB,MZlwCG,MAAO,KYkwCE,GZlwCF,EYkwCO,CZlwCP,C;;MYowCd,OAAO,G;K;IAG
X,2B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACn
B,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MA
EvB,OAAO,G;K;iFAGX,yB;MAAA,sE;MAAA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,
MAAM,6B;QAC/B,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAd,C;UAAyB,OAAO,O;QAChC,eAAe,SAA
S,OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YAC
I,UAAU,C;YACV,WAAW,C;;;QAED,QAAT,QAAS,W;QAClB,OAAO,O;O;KA1BX,C;6FA6BA,+B;MASI,eAAe
,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,cAAc,QAAS,O;MACvB,IAAI,CAAC,QAAS,
UAAd,C;QAAyB,OAAO,O;MAChC,eAAe,SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CAA
T,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,UAAU,C;UACV,WAAW,C;;;MAED,QAAT,QAAS,W;MAClB,
OAAO,O;K;iFAGX,yB;MAAA,sE;MZ/0CA,iB;MY+0CA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;
UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,
QAAS,OAAlB,C;UACR,WZ31CG,MAAO,KY21CO,QZ31CP,EY21CiB,CZ31CjB,C;;QY61Cd,OAAO,Q;O;KAtB
X,C;iFAyBA,yB;MAAA,sE;MZn3CA,iB;MYm3CA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAA
yB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAA
S,OAAlB,C;UACR,WZ/3CG,MAAO,KY+3CO,QZ/3CP,EY+3CiB,CZ/3CjB,C;;QYi4Cd,OAAO,Q;O;KAtBX,C;iF
AyBA,yB;MAAA,sE;MAAA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MAAM,6B;QAC/B,
eAAe,SAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAlB,C;UACR,IAA
I,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAtBX,C;6FAyBA,yB;MZ15CA,iB;MY05CA
,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAlB,C;
QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAlB,C;UACR,WZp6CG,MAAO,KYo6CO,QZp6
CP,EYo6CiB,CZp6CjB,C;;QYs6Cd,OAAO,Q;O;KApBX,C;6FAuBA,yB;MZ57CA,iB;MY47CA,sC;QAaI,eAAe,o
B;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAlB,C;QACf,OAAO,QA
AS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAlB,C;UACR,WZt8CG,MAAO,KYs8CO,QZt8CP,EYs8CiB,CZt8C
jB,C;;QYw8Cd,OAAO,Q;O;KApBX,C;6FAuBA,+B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAy
B,OAAO,I;MAChC,eAAe,SAAS,QAAS,OAAlB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,
OAAlB,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;yFAGX,yB;MAAA,sE;
MAAA,kD;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,
OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAlB,C;UACR,IAAI,UAAW,SAAQ,QA
AR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAtBX,C;qGAyBA,2C;M
AWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,eAAe,SAAS,QAAS,OAAlB,C;MA
Cf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAlB,C;QACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,C
AAlB,CAAX,GAAkC,CAAtC,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;IAGX,iC;MASI,eAAe,oB;MACf,IAAI,
CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QA
AQ,QAAS,O;QACjB,MZrhDG,MAAO,KYqhDE,GZrhDF,EYqhDO,CZrhDP,C;;MYuhDd,OAAO,G;K;IAGX,iC;

MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO
,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZnjDG,MAAO,KYmjDE,GZnjDF,EYmjDO,CZnjDP,C;;MY
qjDd,OAAO,G;K;IAGX,iC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,UAA
U,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UA
Aa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,MA
AM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAA
W,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MA
OI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO,Q
AAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAj
C,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2B;MAaI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QA
AyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZ
l6CG,MAAO,KYk6CE,GZl6CF,EYk6CO,CZl6CP,C;;MYo6Cd,OAAO,G;K;IAGX,2B;MAaI,eAAe,oB;MACf,IAA
I,CAAC,QAAS,UAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,
QAAQ,QAAS,O;QACjB,MZp8CG,MAAO,KYo8CE,GZp8CF,EYo8CO,CZp8CP,C;;MYs8Cd,OAAO,G;K;IAGX,
2B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,
OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,C;UAAa,MAAM,C;;MAE
vB,OAAO,G;K;iFAGX,yB;MAAA,sE;MAAA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,M
AAM,6B;QAC/B,cAAc,QAAS,O;QACvB,IAAI,CAAC,QAAS,UAAd,C;UAAyB,OAAO,O;QAChC,eAAe,SAAS,
OAAT,C;;UAEX,QAAQ,QAAS,O;UACjB,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,
UAAU,C;YACV,WAAW,C;;;QAED,QAAT,QAAS,W;QAClB,OAAO,O;O;KA1BX,C;6FA6BA,+B;MASI,eAAe,o
B;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,cAAc,QAAS,O;MACvB,IAAI,CAAC,QAAS,U
AAd,C;QAAyB,OAAO,O;MAChC,eAAe,SAAS,OAAT,C;;QAEX,QAAQ,QAAS,O;QACjB,QAAQ,SAAS,CAAT,
C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,UAAU,C;UACV,WAAW,C;;;MAED,QAAT,QAAS,W;MAClB,O
AAO,O;K;iFAGX,yB;MAAA,sE;MZjhDA,iB;MYihDA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;U
AAyB,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,Q
AAS,OAAlB,C;UACR,WZ7hDG,MAAO,KY6hDO,QZ7hDP,EY6hDiB,CZ7hDjB,C;;QY+hDd,OAAO,Q;O;KAtB
X,C;iFAyBA,yB;MAAA,sE;MZrjDA,iB;MYqjDA,sC;QAeI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAy
B,MAAM,6B;QAC/B,eAAe,SAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,
OAAlB,C;UACR,WZjkDG,MAAO,KYikDO,QZjkDP,EYikDiB,CZjkDjB,C;;QYmkDd,OAAO,Q;O;KAtBX,C;iFA
yBA,yB;MAAA,sE;MAAA,sC;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MAAM,6B;QAC/B,e
AAe,SAAS,QAAS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAlB,C;UACR,IAAI
,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAtBX,C;6FAyBA,yB;MZ5lDA,iB;MY4lDA,s
C;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAlB,C;Q
ACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAlB,C;UACR,WZtmDG,MAAO,KYsmDO,QZtmD
P,EYsmDiB,CZtmDjB,C;;QYwmDd,OAAO,Q;O;KApBX,C;6FAuBA,yB;MZ9nDA,iB;MY8nDA,sC;QAaI,eAAe,o
B;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,OAAO,I;QAChC,eAAe,SAAS,QAAS,OAAlB,C;QACf,OAAO,QA
AS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAlB,C;UACR,WZxoDG,MAAO,KYwoDO,QZxoDP,EYwoDiB,CZ
xoDjB,C;;QY0oDd,OAAO,Q;O;KApBX,C;6FAuBA,+B;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;Q
AAyB,OAAO,I;MAChC,eAAe,SAAS,QAAS,OAAlB,C;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QA
AS,OAAlB,C;QACR,IAAI,2BAAW,CAAX,KAAJ,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;yFAGX,yB;MAAA,
sE;MAAA,kD;QAaI,eAAe,oB;QACf,IAAI,CAAC,QAAS,UAAd,C;UAAyB,MAAM,6B;QAC/B,eAAe,SAAS,QA
AS,OAAlB,C;QACf,OAAO,QAAS,UAAhB,C;UACI,QAAQ,SAAS,QAAS,OAAlB,C;UACR,IAAI,UAAW,SAAQ
,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAtBX,C;qGAyBA,2
C;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,eAAe,SAAS,QAAS,OAAlB,C
;MACf,OAAO,QAAS,UAAhB,C;QACI,QAAQ,SAAS,QAAS,OAAlB,C;QACR,IAAI,UAAW,SAAQ,QAAR,EAA
kB,CAAlB,CAAX,GAAkC,CAAtC,C;UACI,WAAW,C;;;MAGnB,OAAO,Q;K;IAGX,iC;MASI,eAAe,oB;MACf,I
AAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QAC
I,QAAQ,QAAS,O;QACjB,MZvtDG,MAAO,KYutDE,GZvtDF,EYutDO,CZvtDP,C;;MYytDd,OAAO,G;K;IAGX,i

C;MASI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OA
AO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,MZrvDG,MAAO,KYqvDE,GZrvDF,EYqvDO,CZrvDP,C;;
MYuvDd,OAAO,G;K;IAGX,iC;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,
UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,sBAAM,CAAN,KAAJ,
C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAWI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAy
B,MAAM,6B;MAC/B,UAAU,QAAS,O;MACnB,OAAO,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,
UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD
;MAOI,eAAe,oB;MACf,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,I;MAChC,UAAU,QAAS,O;MACnB,OAA
O,QAAS,UAAhB,C;QACI,QAAQ,QAAS,O;QACjB,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,C
AAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,4B;MAQI,OAAO,CAAC,oBAAW,U;K;+EAGvB,gC;M
AQoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OA
AO,K;;MACrD,OAAO,I;K;IAUI,uC;MAAA,qB;QACP,eAAO,EAAP,C;QAAA,OACA,E;O;K;IATR,sC;MAOI,O
AAO,kBAAI,qBAAJ,C;K;IAeW,8C;MAAA,iC;QACd,eAAO,KAAP,EAAc,OAAd,C;QAAA,OACA,O;O;K;IAXR,
6C;MASI,OAAO,wBAAW,4BAAX,C;K;kFAMX,yB;MAAA,4F;MAAA,uC;QAeI,eAAe,SAAK,W;QACpB,IAAI,
CAAC,QAAS,UAAd,C;UAAyB,MAAM,mCAA8B,kCAA9B,C;QAC/B,kBAAqB,QAAS,O;QAC9B,OAAO,QAA
S,UAAhB,C;UACI,cAAc,UAAU,WAAV,EAAuB,QAAS,OAAhC,C;;QAElB,OAAO,W;O;KArBX,C;gGAwBA,y
B;MAAA,4F;MAAA,wE;MAAA,uC;QAoBmD,Q;QAL/C,eAAe,SAAK,W;QACpB,IAAI,CAAC,QAAS,UAAd,C;
UAAyB,MAAM,mCAA8B,kCAA9B,C;QAC/B,YAAY,C;QACZ,kBAAqB,QAAS,O;QAC9B,OAAO,QAAS,UAA
hB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAmB,oBAAnB,QAAV,EAAuC,WAAvC,EAAoD,QAAS,OAA7D
,C;;QAElB,OAAO,W;O;KAtBX,C;4GAyBA,yB;MAAA,wE;MAAA,uC;QAoBmD,Q;QAL/C,eAAe,SAAK,W;QA
CpB,IAAI,CAAC,QAAS,UAAd,C;UAAyB,OAAO,I;QAChC,YAAY,C;QACZ,kBAAqB,QAAS,O;QAC9B,OAAO
,QAAS,UAAhB,C;UACI,cAAc,UAAU,oBAAmB,YAAnB,EAAmB,oBAAnB,QAAV,EAAuC,WAAvC,EAAoD,Q
AAS,OAA7D,C;;QAElB,OAAO,W;O;KAtBX,C;8FAyBA,gC;MAgBI,eAAe,SAAK,W;MACpB,IAAI,CAAC,QAA
S,UAAd,C;QAAyB,OAAO,I;MAChC,kBAAqB,QAAS,O;MAC9B,OAAO,QAAS,UAAhB,C;QACI,cAAc,UAAU,
WAAV,EAAuB,QAAS,OAAhC,C;;MAElB,OAAO,W;K;IAoBS,2I;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,8C;
MAAA,gD;MAAA,kD;MAAA,wB;MAAA,+B;MAAA,kC;K;;;;sDAAA,Y;;;;;cACZ,gB;8BAAA,iCAAM,0BAAN,
O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;;uCACkB,0B;cACF,wD;cAAhB,gB;;;cAAA,KAAgB,yBAAhB,C;gBAAA,g
B;;;;cAAgB,oC;cACZ,yBAAc,6BAAU,sBAAV,EAAuB,OAAvB,C;cACd,gB;8BAAA,iCAAM,sBAAN,O;kBAAA,
2C;uBAAA,yB;cAAA,Q;;cAFJ,gB;;;cAIJ,W;;;;;;;;;;;;;;K;IAPgB,wF;MAAA,yD;uBAAA,+H;YAAA,S;iBAAA,Q;;i
BAAA,uB;O;K;IAjBpB,sD;MAiBI,OAAO,SAAS,iDAAT,C;K;IA4BS,yJ;MAAA,wC;MAAA,6B;MAAA,yB;MAA
A,8C;MAAA,8D;MAAA,kD;MAAA,wB;MAAA,yB;MAAA,+B;MAAA,kC;K;;;;6DAAA,Y;;;;;kBAKmC,I;cAJ/C,
gB;8BAAA,iCAAM,0BAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;;;iCACY,C;uCACM,0B;cACF,+D;cAAhB,gB;;;
cAAA,KAAgB,yBAAhB,C;gBAAA,gB;;;;cAAgB,oC;cACZ,yBAAc,6BAAU,oBAAmB,uBAAnB,EAAmB,+BAA
nB,QAAV,EAAuC,sBAAvC,EAAoD,OAApD,C;cACd,gB;8BAAA,iCAAM,sBAAN,O;kBAAA,2C;uBAAA,yB;c
AAA,Q;;cAFJ,gB;;;cAIJ,W;;;;;;;;;;;;;;K;IARgB,sG;MAAA,yD;uBAAA,6I;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;I
AlBpB,6D;MAkBI,OAAO,SAAS,wDAAT,C;K;IA2BS,4H;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,oD;MAAA,
kD;MAAA,4B;MAAA,+B;MAAA,kC;K;;;;wDAAA,Y;;;;;oCACG,wC;cACf,IAAI,mBAAS,UAAb,C;yCACyB,mB
AAS,O;gBAC9B,gB;gCAAA,iCAAM,sBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBAFJ,gB;;;;;;;cAGI,gB;;;c
AAA,KAAO,mBAAS,UAAhB,C;gBAAA,gB;;;;cACI,yBAAc,6BAAU,sBAAV,EAAuB,mBAAS,OAAhC,C;cACd,
gB;8BAAA,iCAAM,sBAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cAFJ,gB;;;cAHJ,gB;;;cAQJ,W;;;;;;;;;;;;;;K;IAV
gB,yE;MAAA,yD;uBAAA,gH;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAhBpB,+C;MAgBI,OAAO,SAAS,0CAAT,C
;K;IA6BS,0I;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,kE;MAAA,kD;MAAA,4B;MAAA,+B;MAAA,yB;MAAA
,kC;K;;;;+DAAA,Y;;;;;cAOuC,Q;oCANpC,+C;cACf,IAAI,mBAAS,UAAb,C;yCACyB,mBAAS,O;gBAC9B,gB;gC
AAA,iCAAM,sBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBAFJ,gB;;;;;;;iCAGgB,C;cACZ,gB;;;cAAA,KAAO
,mBAAS,UAAhB,C;gBAAA,gB;;;;cACI,yBAAc,6BAAU,oBAAmB,uBAAnB,EAAmB,+BAAnB,QAAV,EAAuC,
sBAAvC,EAAoD,mBAAS,OAA7D,C;cACd,gB;8BAAA,iCAAM,sBAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cA
FJ,gB;;;cAJJ,gB;;;cASJ,W;;;;;;;;;;;;;;K;IAXgB,uF;MAAA,yD;uBAAA,8H;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IA
hBpB,sD;MAgBI,OAAO,SAAS,iDAAT,C;K;IAcX,+C;MAkBI,OAAO,yBAAY,OAAAZ,EAAAqB,SAArB,C;K;IAGX

,sD;MAmBI,OAAO,gCAAmB,OAAnB,EAA4B,SAA5B,C;K;gFAGX,+B;MASoB,Q;MADhB,UAAe,C;MACC,2B
;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;4FAGX,+B;M
ASoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;
MAEX,OAAO,G;K;iFAGX,+B;MAYoB,Q;MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,y
B;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;iFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAA
hB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;iFAGX,yB;MAAA,S
AWoB,gB;MAXpB,sC;QAYoB,Q;QADhB,Y;QACgB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAO,S
AAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAfX,C;iFAkBA,yB;M5B/jEA,6B;M4B+jEA,sC;QAaoB,Q;QADhB,U5
BjkEmC,c4BikEnB,C5BjkEmB,C;Q4BkkEnB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,M5B/4EiD,c4B+
4EjD,G5B/4E2D,KAAK,G4B+4EzD,SAAS,OAAT,C5B/4EoE,KAAX,IAAf,C;;Q4Bi5ErD,OAAO,G;O;KAhBX,C;
iFAmBA,yB;MX/kEA,+B;MW+kEA,sC;QAaoB,Q;QADhB,UXhlEqC,eAAW,oBWglE/B,CXhlE+B,CAAX,C;QW
ilErB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MX/5EmD,eW+5EnD,GX/5E8D,KAAK,KW+5E5D,SA
AS,OAAT,CX/5EuE,KAAX,CAAhB,C;;QWi6EvD,OAAO,G;O;KAhBX,C;IAyBe,oD;MAAA,qB;QAAE,e;UAAM
,MAAM,gCAAyB,2BAAwB,mBAAxB,MAAzB,C;;QAAZ,S;O;K;IANjB,qC;MAMI,OAAO,kBAAI,gCAAJ,C;K;I
AGX,oC;MAaI,OAAO,sBAAS,IAAT,EAAe,IAAf,EAAsC,IAAtC,C;K;IAGX,+C;MAkBI,OAAO,sBAAS,IAAT,E
AAe,IAAf,EAAsC,IAAtC,EAAwD,SAAxD,C;K;IASA,0D;MAAA,4B;MAAA,sC;K;IAG0B,+E;MAAA,qB;QAAE
,IAAI,CAAC,iBAAD,IAAY,WAAM,eAAN,CAAhB,C;UAAiC,oBAAU,I;UAA3C,OAAiD,K;;UAAjD,OAA8D,I;
O;K;6CAF7F,Y;MACI,kBAAc,KAAd,C;MACA,OAAkB,SAAX,eAAW,EAAO,kEAAP,CAA8E,W;K;;IAT5G,qC;
MAMI,kD;K;IAkBO,6D;MAAA,4B;MAAA,wC;K;IAE6B,iE;MAAA,qB;QAAE,gBAAM,gBAAN,K;O;K;+CADl
C,Y;MACI,OAAkB,YAAX,eAAW,EAAU,4DAAV,CAA6B,W;K;;IAZ3D,sC;MASI,IAAI,Q/B0qKG,YAAQ,C+B1
qKf,C;QAAwB,OAAO,S;MAC/B,qD;K;IAgBO,6D;MAAA,wC;MAAA,4B;K;IAMiC,8D;MAAA,qB;QAAE,OAA
M,aAAN,mB;O;K;+CALtC,Y;MACI,YAAqB,6BAAT,qBAAS,C;MACrB,IAAI,KAAM,UAAV,C;QACI,OAAO,e
AAW,W;;QAElB,OAAkB,YAAX,eAAW,EAAU,4CAAV,CAA0B,W;K;;IAf5D,sC;MASI,qD;K;IAoBO,6D;MAA
A,wC;MAAA,4B;K;IAMiC,8D;MAAA,qB;QAAE,OAAM,aAAN,mB;O;K;+CALtC,Y;MACI,YAAqB,UAAT,qB
AAS,C;MACrB,IAAI,KAAM,UAAV,C;QACI,OAAO,eAAW,W;;QAElB,OAAkB,YAAX,eAAW,EAAU,4CAAV,
CAA0B,W;K;;IAf5D,sC;MASI,qD;K;8FAWJ,yB;MAAA,4C;MAAA,qC;QAOI,OAAO,iBAAM,OAAN,C;O;KAP
X,C;wFAUA,yB;MAAA,+D;MAAA,6B;MAAA,uC;QAYoB,Q;QAFhB,YAAY,gB;QACZ,aAAa,gB;QACG,2B;Q
AAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,KAAM,WAAI,OAAJ,C;;YAE
N,MAAO,WAAI,OAAJ,C;;;QAGf,OAAO,cAAK,KAAL,EAAY,MAAZ,C;O;KAnBX,C;IAsBA,oC;MAMI,OAA6
C,UAAtC,YAAW,SAAX,EAAiB,YAAW,OAAX,EAAjB,EAAsC,C;K;IAGjD,qC;MASI,OAAY,OAAL,SAAK,EA
Ac,OAAT,QAAS,CAAd,C;K;IAGhB,qC;MASI,OAA+C,UAAxC,YAAW,SAAX,EAA0B,aAAT,QAAS,CAA1B,E
AAwC,C;K;IAGnD,sC;MASI,OAAkC,UAA3B,YAAW,SAAX,EAAiB,QAAjB,EAA2B,C;K;4FAGtC,yB;MAAA,
0C;MAAA,qC;QAOI,OAAO,gBAAK,OAAL,C;O;KAPX,C;IAUA,2D;MAgB+C,oB;QAAA,OAAY,C;MAAG,8B;
QAAA,iBAA0B,K;MACpF,OAAO,8BAAiB,IAAjB,EAAuB,IAAvB,EAA6B,cAA7B,EAA2D,KAA3D,C;K;IAGX,
sE;MAkBkD,oB;QAAA,OAAY,C;MAAG,8B;QAAA,iBAA0B,K;MACvF,OAAwE,OAAjE,8BAAiB,IAAjB,EAA
uB,IAAvB,EAA6B,cAA7B,EAA2D,IAA3D,CAAiE,EAAI,SAAJ,C;K;IAYpC,4B;MAAY,cAAM,EAAN,C;K;IATp
D,kC;MASI,OAAO,oBAAgB,SAAhB,EAAsB,KAAtB,EAA6B,EAA6B,UAA7B,C;K;IAGX,6C;MAUI,OAAO,oBAAgB,S
AAhB,EAAsB,KAAtB,EAA6B,SAA7B,C;K;IAcY,kC;MAAU,aAAK,CAAL,C;K;IAXjC,kC;MAWI,OAAO,yBAA
Y,kBAAZ,C;K;IAeiB,wH;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,gD;MAAA,kD;MAAA,4B;MAAA,2B;MAA
A,wB;MAAA,kC;K;;;;sDAAA,Y;;;;;oCACL,sC;cACf,IAAI,CAAC,mBAAS,UAAd,C;gBAAyB,M;;gBAAzB,gB;;;;
;;;mCACc,mBAAS,O;cACvB,gB;;;cAAA,KAAO,mBAAS,UAAhB,C;gBAAA,gB;;;;gCACe,mBAAS,O;cACpB,gB
;8BAAA,iCAAM,6BAAU,kBAAV,EAAmB,eAAnB,CAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cACA,qBAAU,e
;cAHd,gB;;;cAKJ,W;;;;;;;;;;;;;;K;IATwB,uE;MAAA,yD;uBAAA,4G;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAZ5B,
6C;MAYI,OAAO,SAAS,0CAAT,C;K;IAYX,8F;MAU6D,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MA
AI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,QAAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;
MAGtN,Q;MAFhB,MAAO,gBAAO,MAAP,C;MACP,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,y
B;QACZ,IAAI,iCAAU,CAAd,C;UAAiB,MAAO,gBAAO,SAAP,C;QACxB,IAAI,QAAQ,CAAR,IAAa,SAAS,KA
A1B,C;UACW,gBAAP,MAAO,EAAc,OAAd,EAAuB,SAAvB,C;;UACJ,K;;MAEX,IAAI,SAAS,CAAT,IAAc,QA

AQ,KAA1B,C;QAAiC,MAAO,gBAAO,SAAP,C;MACxC,MAAO,gBAAO,OAAP,C;MACP,OAAO,M;K;IAGX,4
F;MAUwC,yB;QAAA,YAA0B,I;MAAM,sB;QAAA,SAAuB,E;MAAI,uB;QAAA,UAAwB,E;MAAI,qB;QAAA,Q
AAa,E;MAAI,yB;QAAA,YAA0B,K;MAAO,yB;QAAA,YAAoC,I;MACjN,OAAO,oBAAO,sBAAP,EAAwB,SAA
xB,EAAmC,MAAnC,EAA2C,OAA3C,EAAoD,KAApD,EAA2D,SAA3D,EAAsE,SAAtE,CAAiF,W;K;IAOxE,8C;
MAAA,mB;QAAE,OAAA,eAAK,W;O;K;IAJ3B,kC;MAII,oCAAgB,8BAAhB,C;K;2FAGJ,qB;MAKI,OAAO,S;K;
IAGX,+B;MASoB,Q;MAFhB,UAAkB,G;MAClB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;
QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,I
AAvB,GAAgC,MAAM,K;K;IAGjD,+B;MASoB,Q;MAFhB,UAAkB,G;MAClB,YAAiB,C;MACD,2B;MAAhB,O
AAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAnB,E;;MAEJ,OAAW,UA
AS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MASoB,Q;MAFhB,UAAkB,G;MAClB,YA
AiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,
KAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MASoB,Q;M
AFhB,UAAkB,G;MAClB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,
oBAAmB,qBAAnB,EAAmB,KAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,
K;K;IAGjD,+B;MASoB,Q;MAFhB,UAAkB,G;MAClB,YAAiB,C;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB
,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAnB,E;;MAEJ,OAAW,UAAS,CAAb,GAAgB,wCA
AO,IAAvB,GAAgC,MAAM,K;K;IAGjD,+B;MASoB,Q;MAFhB,UAAkB,G;MAClB,YAAiB,C;MACD,2B;MAAh
B,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,O;QACP,oBAAmB,qBAAnB,EAAmB,KAAnB,E;;MAEJ,OAAW
,UAAS,CAAb,GAAgB,wCAAO,IAAvB,GAAgC,MAAM,K;K;IAGjD,2B;MAQoB,Q;MADhB,UAAe,C;MACC,2
B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,U
AAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAQ
oB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,OAAP,I;;MAEJ,OAA
O,G;K;IAGX,2B;MAQoB,Q;MADhB,Y;MACgB,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,cAAO,OAA
P,C;;MAEJ,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAiB,G;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB
,yB;QACZ,OAAO,O;;MAEX,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAgB,
cAAhB,C;QAAgB,yB;QACZ,OAAO,O;;MAEX,OAAO,G;K;IC//FX,qC;MAMI,aAAa,qBAAiB,YAAY,cAAZ,CA
AjB,C;MACb,kBAAc,KAAd,C;MX8zBgB,Q;MAAA,OW7zBT,SX6zBS,W;MAAhB,OAAgB,cAAhB,C;QAAgB,2
B;QAAU,oB;QW7zBK,IAAI,CAAC,SAAD,IAAY,OX6zBX,SW7zBW,UAAhB,C;UAAiC,YAAU,I;UAA3C,mBA
AiD,K;;UAAjD,mBAA8D,I;;QX6zBvE,qB;UW7zBD,MX6zBqC,WAAI,SAAJ,C;;MW7zB1D,OAAqB,M;K;IAGz
B,sC;MAMI,aAAa,qBAAiB,SAAjB,C;MACN,YAAP,MAAO,EAAU,QAAV,C;MACP,OAAO,M;K;IAGX,sC;MA
MI,YAAqB,6BAAT,QAAS,C;MACrB,IAAI,KAAM,UAAV,C;QACI,OAAY,QAAL,SAAK,C;MAChB,IAAI,yBA
AJ,C;QACgB,kBAAY,sB;QX2xBZ,Q;QAAA,OW3xBL,SX2xBK,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UA
AM,IAAI,CW3xBwB,qBX2xBb,OW3xBa,CX2xB5B,C;YAAyB,WAAY,WAAI,OAAJ,C;;QW3xBvD,OX4xBG,W
;;MW3xBP,aAAa,qBAAiB,SAAjB,C;MACb,MAAO,mBAAU,KAAV,C;MACP,OAAO,M;K;IAGX,uC;MAMI,aA
Aa,qBAAiB,SAAjB,C;MACN,YAAP,MAAO,EAAU,QAAV,C;MACP,OAAO,M;K;gGAGX,yB;MAAA,8C;MAA
A,qC;QAOI,OAAO,iBAAM,OAAN,C;O;KAPX,C;IAUA,qC;MAMI,aAAa,qBAAiB,YAAY,iBAAO,CAAP,IAAZ,
CAAjB,C;MACb,MAAO,gBAAO,SAAP,C;MACP,MAAO,WAAI,OAAJ,C;MACP,OAAO,M;K;IAGX,sC;MAOI,
aAAa,qBAAiB,YAAY,SAAK,KAAL,GAAY,QAAS,OAArB,IAAZ,CAAjB,C;MACb,MAAO,gBAAO,SAAP,C;M
ACA,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;IAGX,sC;MAMuD,UAAT,M;MAA1C,aAAa,qBAAiB,
YAAY,WAAS,4BAAT,QAAS,CAAT,YAA4C,cAAL,WAAvC,4BAA2D,SAAK,KAAL,GAAY,CAAZ,IAAvE,CA
AjB,C;MACb,MAAO,gBAAO,SAAP,C;MACA,OAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;IAGX,sC;M
AOI,aAAa,qBAAiB,YAAY,SAAK,KAAL,GAAY,CAAZ,IAAZ,CAAjB,C;MACb,MAAO,gBAAO,SAAP,C;MAC
A,SAAP,MAAO,EAAO,QAAP,C;MACP,OAAO,M;K;8FAGX,yB;MAAA,4C;MAAA,qC;QAOI,OAAO,gBAAK,
OAAL,C;O;KAPX,C;InBvHA,oD;MAMuF,wC;K;IANvF,8CAOI,Y;MAAuC,8B;K;IAP3C,gF;ICGA,oD;MAQuF,
wC;K;IARvF,8CASI,Y;MAAuC,8B;K;IAT3C,gF;gGmBYA,yB;MAAA,uD;MAAA,gC;MAAA,iD;QAOI,OAAW,
SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,qBAAI,KAAJ,CAAtC,GAAsD,uBAAa,KAAb,E;O;KAPjE,C;gGAU
A,yB;MAAA,+C;MAAA,mC;QAOI,OAAY,UAAL,SAAK,EAAU,KAAV,C;O;KAPhB,C;0EAUA,yB;MA6EA,6C;
MAAA,oC;MAAA,gC;MA7EA,uC;QAOW,sB;;UA0ES,Q;UAAA,0B;UAAhB,OAAgB,cAAhB,C;YAAgB,oC;YA

AM,IA1EH,SA0EO,CAAU,oBAAV,CAAJ,C;cAAwB,qBAAO,O;cAAP,uB;;;UAC9C,qBAAO,I;;;QA3EP,yB;O;K
APJ,C;kFAUA,yB;MAyJA,mD;MAAA,+C;MAAA,oC;MAzJA,uC;QAOW,qB;;UAwJO,Q;UAAA,OAAa,SAAR,s
BAAQ,CAAb,W;UAAd,OAAc,cAAd,C;YAAc,uB;YACV,cAAc,qBAAK,KAAL,C;YACd,IA1Jc,SA0JV,CAAU,o
BAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA5JP,wB;O;KAPJ,C;IAUA,6B;MAMI,ICiO
gD,qBAAU,CDjO1D,C;QACI,MAAM,2BAAuB,yBAAvB,C;MACV,OAAO,qBAAK,CAAL,C;K;4EAGX,yB;MA
AA,6C;MAAA,oC;MAAA,gC;MAAA,iE;MAAA,uC;QAKoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,o
C;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,6DAAvB,C;O;KANV,C;6
FASA,yB;MAAA,iE;MAYA,6C;MAAA,oC;MAAA,gC;MAZA,uC;QAS8C,IAAnC,I;QAAA,+B;;UAYS,U;UAAA
,4B;UAAhB,OAAgB,gBAAhB,C;YAAgB,sC;YACZ,aAbwB,SAaX,CAAU,oBAAV,C;YACb,IAAI,cAAJ,C;cACI,
8BAAO,M;cAAP,gC;;;UAGR,8BAAO,I;;;QAlBA,kC;QAAA,iB;UAAmC,MAAM,gCAAuB,sEAAvB,C;;QAAhD,
OAAO,I;O;KATX,C;yGAYA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QASoB,Q;QAAA,0B;QAAhB,O
AAgB,cAAhB,C;UAAgB,oC;UACZ,aAAa,UAAU,oBAAV,C;UACb,IAAI,cAAJ,C;YACI,OAAO,M;;;QAGf,OAA
O,I;O;KAfX,C;IAkBA,mC;MAII,OCiLgD,qBAAU,CDjLnD,GAAe,IAAf,GAAyB,qBAAK,CAAL,C;K;wFAGpC,y
B;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAIoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UA
AM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,OAAO,I;O;KALX,C;mFAQA,yB;MAAA,uD;MA
AA,gC;MAAA,iD;QAKI,OAAW,SAAS,CAAT,IAAc,SAAS,wBAA3B,GAAsC,qBAAI,KAAJ,CAAtC,GAAsD,uB
AAa,KAAb,E;O;KALjE,C;IAQA,uC;MAMI,OAAW,SAAS,CAAT,IAAc,SAAS,2BAA3B,GAAsC,qBAAI,KAAJ,
CAAtC,GAAsD,I;K;0FAGjE,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAIkB,gC;QAAA,6B;QAAA,mB;QAAA,kB;
QAAA,kB;QAAd,0D;UACI,IAAI,UAAU,iCAAK,KAAL,EAAV,CAAJ,C;YACI,OAAO,K;;;QAGf,OAAO,E;O;K
ATX,C;wFAYA,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA,uC;QAIkB,Q;QAAA,OAAQ,SAAR,sBAAQ,CAA
R,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,IAAI,UAAU,iCAAK,KAAL,EAAV,CAAJ,C;YACI,OAAO,K;;;QA
Gf,OAAO,E;O;KATX,C;IAYA,4B;MAQI,ICqHgD,qBAAU,CDrH1D,C;QACI,MAAM,2BAAuB,yBAAvB,C;MA
CV,OAAO,qBAAK,2BAAL,C;K;0EAGX,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA,iE;MAAA,uC;QAQkB,
Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,qBAAK,K
AAL,C;UACd,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,6DAAvB,C;O;KAZ
V,C;IAeA,kC;MAMI,OC2FgD,qBAAU,CD3FnD,GAAe,IAAf,GAAyB,qBAAK,mBAAS,CAAT,IAAL,C;K;sFAG
pC,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA,uC;QAMkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,CA
Ab,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,qBAAK,KAAL,C;UACd,IAAI,UAAU,oBAAV,CAAJ,C;YA
AwB,OAAO,O;;QAEnC,OAAO,I;O;KAVX,C;8EAaA,yB;MAAA,mC;MAAA,yC;MAAA,4B;QAQI,OAAO,kBA
AO,cAAP,C;O;KARX,C;IAWA,sC;MAOI,ICyDgD,qBAAU,CDzD1D,C;QACI,MAAM,2BAAuB,yBAAvB,C;MA
CV,OAAO,qBAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;0FAGX,yB;MAAA,mC;MAAA,qD;MAAA,4B;QAOI,O
AAO,wBAAa,cAAb,C;O;KAPX,C;IAUA,4C;MAMI,ICoCgD,qBAAU,CDpC1D,C;QACI,OAAO,I;MACX,OAAO,
qBAAI,MAAO,iBAAQ,gBAAR,CAAX,C;K;IAGX,8B;MAIiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAA
K,MAAM,2BAAuB,yBAAvB,C;aACX,C;UAAK,4BAAK,CAAL,C;UAAL,K;;UACQ,MAAM,gCAAyB,0CAAzB,
C;;MAHlB,W;K;8EAOJ,yB;MAAA,6C;MAAA,oC;MAAA,kF;MAAA,gC;MAAA,iE;MAAA,8B;MAAA,uC;QA
MoB,UAST,M;QAXP,aAAoB,I;QACpB,YAAY,K;QACI,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,IAAI
,UAAU,oBAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,wDAAzB,C;YACjB,SAAS,O;YACT,QA
AQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAY,MAAM,gCAAuB,6DAAvB,C;QAElB,OAAO,4E;O;KAfX,C;IAkB
A,oC;MAII,OAAW,qBAAU,CAAd,GAAiB,qBAAK,CAAL,CAAjB,GAA8B,I;K;0FAGzC,yB;MAAA,6C;MAAA,
oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAFhB,aAAoB,I;QACpB,YAAY,K;QACI,0B;QAAhB,OAAgB,cAAhB,C;
UAAgB,oC;UACZ,IAAI,UAAU,oBAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,OAAO,I;YAClB,SAAS,O;YACT,
QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAY,OAAO,I;QACnB,OAAO,M;O;KAdX,C;IAiBA,+B;MlB1RI,IAA
I,EkBkSI,KAAK,ClBlST,CAAJ,C;QACI,ckBiSc,wD;QlBhSd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MkBiSV,OAA
O,8BAAc,eAAF,CAAE,EAAa,gBAAb,CAAd,EAAoC,gBAApC,C,C;K;IAGX,+B;MlBtSI,IAAI,EkB8SI,KAAK,ClB
9ST,CAAJ,C;QACI,ckB6Sc,wD;QlB5Sd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MkB6SV,OLxF6E,oBKwF1D,eAA
F,CAAE,EAAa,gBAAb,CLxF0D,C;K;IK2FjF,kC;MlBlTI,IAAI,EkB0TI,KAAK,ClB1TT,CAAJ,C;QACI,ckByTc,w
D;QlBxTd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MkByTV,OAAO,mBAAkB,gBAAZ,mBAAS,CAAT,IAAY,EAA
c,CAAd,CAAlB,C;K;IAGX,mC;MlB9TI,IAAI,EkBsUI,KAAK,ClBtUT,CAAJ,C;QACI,ckBqUc,wD;QlBpUd,MA

AM,gCAAyB,OAAQ,WAAjC,C;;MkBqUV,OAAO,mBAAkB,gBAAZ,mBAAS,CAAT,IAAY,EAAc,CAAd,CAAl
B,C;K;2FAGX,yB;MAAA,uD;MAAA,oC;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CA
AC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OAAO,8BAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;QACf,O
AAO,E;O;KATX,C;4FAYA,yB;MAAA,uD;MAAA,oC;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UA
CI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OLpIoF,oBKoInE,CLpImE,EKoIhE,QAAQ,CAAR,
ILpIgE,C;;QKqI5F,OAAO,E;O;KATX,C;oFAYA,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAMuB,UAAL,MAAK,
EAAL,MAAK,EAAL,M;QAAK,mBAAL,SAAK,C;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,CAA
C,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OAAO,8BAAY,KAAZ,EAAmB,gBAAnB,C;QACf,OAAO,E;O;
KATX,C;oFAYA,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAMuB,UAAL,MAAK,EAAL,MAAK,EAAL,M;QAAK
,mBAAL,SAAK,C;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV
,CAAL,C;YACI,OL/JqE,oBK+JpD,KL/JoD,C;;QKgK7E,OAAO,E;O;KATX,C;8EAYA,yB;MAAA,yD;MAkFA,o
C;MAlFA,uC;QAMW,kBAAS,oB;QAkFM,Q;QAAA,uB;QAAtB,iBAAc,CAAd,wB;UACI,cAAc,qBAAI,KAAJ,C;
UACd,IApF6B,SAoFzB,CAAU,oBAAV,CAAJ,C;YAAwB,WAAY,gBAAO,OAAP,C;;QApFxC,OAsFO,W;O;KA
5FX,C;8EASA,yB;MAAA,yD;MAyEA,oC;MAzEA,uC;QAMW,kBAAS,oB;QAyEM,Q;QAAA,uB;QAAtB,iBAAc
,CAAd,wB;UACI,cAAc,qBAAI,KAAJ,C;UACd,IA3E6B,SA2EzB,CAAU,oBAAV,CAAJ,C;YAAwB,WAAY,gBA
AO,OAAP,C;;QA3ExC,OA6EO,WA7EqC,W;O;KANhD,C;4FASA,yB;MAAA,yD;MAsBA,gC;MAmtBA,6C;MA
AA,oC;MAzuBA,uC;QAQW,kBAAgB,oB;QAwuBV,gB;QADb,YAAY,C;QACC,0B;QAAb,OAAa,cAAb,C;UAA
a,iC;UAAM,eAAO,cAAP,EAAO,sBAAP,S;UAAA,cAAgB,iB;UAjtB/B,IAvBoC,SAuBhC,CAAU,OAAV,EAAiB,
OAAjB,CAAJ,C;YAA2C,2BAAO,kBAAP,C;;QAvB/C,OAyBO,W;O;KAjCX,C;4FAWA,yB;MAAA,yD;MAWA,g
C;MAmtBA,6C;MAAA,oC;MA9tBA,uC;QAQW,kBAAgB,oB;QA6tBV,gB;QADb,YAAY,C;QACC,0B;QAAb,O
AAa,cAAb,C;UAAa,iC;UAAM,eAAO,cAAP,EAAO,sBAAP,S;UAAA,cAAgB,iB;UAjtB/B,IAZoC,SAYhC,CAAU
,OAAV,EAAiB,OAAjB,CAAJ,C;YAA2C,2BAAO,kBAAP,C;;QAZ/C,OAcO,WAd4C,W;O;KARvD,C;gGAWA,y
B;MAAA,gC;MAmtBA,6C;MAAA,oC;MAntBA,oD;QA0tBiB,gB;QADb,YAAY,C;QACC,0B;QAAb,OAAa,cAA
b,C;UAAa,iC;UAAM,eAAO,cAAP,EAAO,sBAAP,S;UAAA,cAAgB,iB;UAjtB/B,IAAI,UAAU,OAAV,EAAiB,OA
AjB,CAAJ,C;YAA2C,2BAAO,kBAAP,C;;QAE/C,OAAO,W;O;KAXX,C;oFAcA,yB;MAAA,yD;MAkBA,6C;MA
AA,oC;MAAA,gC;MAlBA,uC;QAMW,kBAAY,oB;QAkBH,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC
;UAAM,IAAI,CAlBU,SAkBT,CAAU,oBAAV,CAAL,C;YAAyB,WAAY,gBAAO,OAAP,C;;QAlB3D,OAmBO,W
;O;KAzBX,C;oFASA,yB;MAAA,yD;MASA,6C;MAAA,oC;MAAA,gC;MATA,uC;QAMW,kBAAY,oB;QASH,Q;
QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,CATU,SAST,CAAU,oBAAV,CAAL,C;YAAyB,
WAAY,gBAAO,OAAP,C;;QAT3D,OAUO,WAVwC,W;O;KANnD,C;wFASA,yB;MAAA,6C;MAAA,oC;MAAA,
gC;MAAA,oD;QAMoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,CAAC,UAAU,oBAA
V,CAAL,C;YAAyB,WAAY,gBAAO,OAAP,C;;QAC3D,OAAO,W;O;KAPX,C;kFAUA,yB;MAAA,oC;MAAA,oD
;QAM0B,Q;QAAA,uB;QAAtB,iBAAc,CAAd,wB;UACI,cAAc,qBAAI,KAAJ,C;UACd,IAAI,UAAU,oBAAV,CA
AJ,C;YAAwB,WAAY,gBAAO,OAAP,C;;QAExC,OAAO,W;O;KAVX,C;IAaA,sC;MAII,IAAI,OAAQ,UAAZ,C;Q
AAuB,OAAO,E;MAC9B,OAAO,yBAAY,OAAZ,C;K;IAGX,sC;MAII,IAAI,OAAQ,UAAZ,C;QAAuB,OAAO,E;
MAC9B,OAAO,uBAAU,OAAV,C;K;IAGX,sC;MAOc,Q;MAHV,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;
MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,E;MACtB,aAAa,mBAAc,IAAd,C;MACH,yB;MAAV,OAAU,cAAV
,C;QAAU,mB;QACN,MAAO,gBAAO,qBAAI,CAAJ,CAAP,C;;MAEX,OAAO,M;K;4EAGX,yB;MAAA,8B;MAA
A,uC;MAAA,qC;QAKY,Q;QAAR,OAA8B,MAAtB,2DAAsB,EAAM,OAAN,CAAe,W;O;KALjD,C;IAQA,+B;Ml
B9fI,IAAI,EkBsgBI,KAAK,ClBtgBT,CAAJ,C;QACI,ckBqgBc,wD;QlBpgBd,MAAM,gCAAyB,OAAQ,WAAjC,C;
;MkBqgBV,OAAO,8BAAY,CAAZ,EAAiB,eAAF,CAAE,EAAa,gBAAb,CAAjB,C;K;IAGX,+B;MlB1gBI,IAAI,Ek
BkhBI,KAAK,ClBlhBT,CAAJ,C;QACI,ckBihBc,wD;QlBhhBd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MkBihBV,O
LzT4F,oBKyT3E,CLzT2E,EKyTtE,eAAF,CAAE,EAAa,gBAAb,CLzTsE,C;K;IK4ThG,kC;MlBthBI,IAAI,EkB8hB
I,KAAK,ClB9hBT,CAAJ,C;QACI,ckB6hBc,wD;QlB5hBd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MkB6hBV,aAAa,
gB;MACb,OAAO,8BAAY,SAAW,eAAF,CAAE,EAAa,MAAb,CAAX,IAAZ,EAA6C,MAA7C,C;K;IAGX,mC;Ml
BniBI,IAAI,EkB2iBI,KAAK,ClB3iBT,CAAJ,C;QACI,ckB0iBc,wD;QlBziBd,MAAM,gCAAyB,OAAQ,WAAjC,C;
;MkB0iBV,aAAa,gB;MACb,OLtV6E,oBKsV5D,SAAW,eAAF,CAAE,EAAa,MAAb,CAAX,ILtV4D,C;K;2FKyVj
F,yB;MAAA,uD;MAAA,oC;MAAA,uC;QAMI,iBAAc,wBAAd,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,iC

AAK,KAAL,EAAV,CAAL,C;YACI,OAAO,8BAAY,QAAQ,CAAR,IAAZ,EAAuB,gBAAvB,C;;;QAGf,OAAO,8B
AAY,CAAZ,EAAe,gBAAf,C;O;KAXX,C;4FAcA,yB;MAAA,uD;MAAA,oC;MAAA,uC;QAMI,iBAAc,wBAAd,
WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OL/WqE,oBK+WpD,QAA
Q,CAAR,IL/WoD,C;;;QKkX7E,OAAO,S;O;KAXX,C;oFAcA,yB;MAAA,oC;MAAA,uC;QAM0B,Q;QAAA,uB;Q
AAtB,iBAAc,CAAd,wB;UACI,IAAI,CAAC,UAAU,iCAAI,KAAJ,EAAV,CAAL,C;YACI,OAAO,8BAAY,CAAZ,
EAAe,KAAf,C;;QAEf,OAAO,8BAAY,CAAZ,EAAe,gBAAf,C;O;KAVX,C;oFAaA,yB;MAAA,oC;MAAA,uC;QA
M0B,Q;QAAA,uB;QAAtB,iBAAc,CAAd,wB;UACI,IAAI,CAAC,UAAU,iCAAI,KAAJ,EAAV,CAAL,C;YACI,O
LvYoF,oBKuYnE,CLvYmE,EKuYhE,KLvYgE,C;;QKyY5F,OAAO,S;O;KAVX,C;IAaA,gC;MAII,OAAO,qBAAc,
SAAd,CAAoB,U;K;kFAG/B,yB;MAAA,8B;MAAA,6C;MAAA,4B;QAKY,Q;QAAR,OAA8B,SAAtB,2DAAsB,C
AAW,W;O;KAL7C,C;oFAQA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MA4EA,6C;MAAA,oC;MAAA,gC;MA5EA
,uC;QAWI,eAAmC,cAApB,YAAY,gBAAZ,CAAoB,EAAc,EAAd,C;QAC5B,kBAAY,mBAAoB,QAApB,C;QAyE
H,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WA1E8C,SA0E/B,CAAU,oBAAV,C;U3B3EnB,w
BAAI,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;Q2BCA,OA4EO,W;O;KAxFX,C;wFAeA,yB;MAAA,0D;MAAA,y
D;MAAA,uE;MA6BA,6C;MAAA,oC;MAAA,gC;MA7BA,yC;QAWI,eAAmC,cAApB,YAAY,gBAAZ,CAAoB,E
AAc,EAAd,C;QAC5B,kBAAc,mBAAuB,QAAvB,C;QA2BL,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC
;UACZ,WAAY,aA5BuC,WA4BnC,CAAY,oBAAZ,CAAJ,EAA0B,oBAA1B,C;;QA5BhB,OA8BO,W;O;KA1CX,C
;wFAeA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MA8BA,6C;MAAA,oC;MAAA,gC;MA9BA,yD;QAUI,eAAmC,c
AApB,YAAY,gBAAZ,CAAoB,EAAc,EAAd,C;QAC5B,kBAAc,mBAAoB,QAApB,C;QA6BL,Q;QAAA,0B;QAA
hB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAAY,aA9BoC,WA8BhC,CAAY,oBAAZ,CAAJ,EA9BiD,cA8BvB,C
AAe,oBAAf,CAA1B,C;;QA9BhB,OAgCO,W;O;KA3CX,C;4FAcA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,
sD;QAUoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAAY,aAAI,YAAY,oBAAZ,CAAJ,EA
A0B,oBAA1B,C;;QAEhB,OAAO,W;O;KAbX,C;4FAgBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sE;QAU
oB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAAY,aAAI,YAAY,oBAAZ,CAAJ,EAA0B,eA
Ae,oBAAf,CAA1B,C;;QAEhB,OAAO,W;O;KAbX,C;wFAgBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oD;
QASoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAAe,UAAU,oBAAV,C;U3B3EnB,wBAAI
,IAAK,MAAT,EAAgB,IAAK,OAArB,C;;Q2B6EA,OAAO,W;O;KAZX,C;4FAeA,yB;MAAA,uD;MAAA,0D;MA
AA,yD;MAAA,uE;MAgBA,6C;MAAA,oC;MAAA,gC;MAhBA,2C;QAYI,aAAAa,mBAA6D,cAAtC,YAAmB,aAA
P,gBAAO,EAAa,GAAb,CAAnB,CAAsC,EAAc,EAAd,CAA7D,C;QAcG,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C
;UAAgB,oC;UAbO,MAcP,aAAI,oBAAJ,EAde,aAcF,CAAc,oBAAd,CAAb,C;;QAdhB,OAAuB,M;O;KAb3B,C;+F
AgBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,wD;QAUoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAA
gB,oC;UACZ,WAAY,aAAI,oBAAJ,EAAAa,cAAc,oBAAd,CAAb,C;;QAEhB,OAAO,W;O;KAbX,C;IAgBA,iD;MA
IiB,Q;MAAA,4B;MAAb,OAAa,cAAb,C;QAAa,iC;QACT,WAAY,WAAI,iBAAJ,C;;MAEhB,OAAO,W;K;IAGX,i
C;MAII,OAAO,2BAAa,eAAc,YAAmB,eAAP,gBAAO,EAAa,GAAb,CAAnB,CAAd,CAAb,C;K;IAGX,8B;MAIiB
,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,kB;UAAL,K;aACA,C;UAAK,cAAO,iCAAK,CAAL,EAAP,C
;UAAL,K;;UACa,wBAAL,SAAK,C;UAHV,K;;MAAP,W;K;IAOJ,qC;MAII,OAAO,2BAAa,iBAAgB,gBAAhB,C
AAb,C;K;IAGX,6B;MAMiB,IAAN,I;MAAA,QAAM,gBAAN,C;aACH,C;UAAK,iB;UAAL,K;aACA,C;UAAK,aA
AM,iCAAK,CAAL,EAAN,C;UAAL,K;;UACQ,kCAAa,qBAAoB,YAAmB,eAAP,gBAAO,EAAa,GAAb,CAAnB,
CAApB,CAAb,C;UAHL,K;;MAAP,W;K;gFAOJ,yB;MAAA,+D;MA0CA,6C;MAAA,oC;MAAA,gD;MAAA,gC;
MA1CA,uC;QAMW,kBAAU,gB;QAwCD,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAzC6B,
SAyClB,CAAU,oBAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA1ChB,OA4CO,W;O;KAlDX,C;8FASA,yB;
MAAA,+D;MAeA,6C;MAAA,oC;MAAA,gD;MAAA,gC;MAfA,uC;QAYW,kBAAAiB,gB;QAcR,gB;QADhB,YAA
Y,C;QACI,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAfoC,SAezB,EAAU,cAAV,EAAU,sBAAV,WAA
mB,oBAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAhBhBhB,OAkBO,W;O;KA9BX,C;kGAeA,yB;MAAA,6C
;MAAA,oC;MAAA,gD;MAAA,gC;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACI,0B;QAAhB,OAAgB,c
AAhB,C;UAAgB,oC;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,oBAAnB,C;UACC,OAAZ,WAAY,
EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAfX,C;oFAkBA,yB;MAAA,6C;MAAA,oC;MAAA,gD;MAAA,gC;MAA
A,oD;QAIoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,WAAW,UAAU,oBAAV,C;UACC,OA
AZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KARX,C;gFAWA,yB;MAAA,wE;MAyBA,6C;MAAA,oC;MA

AA,+D;MAAA,gC;MAzBA,yC;QASW,kBAAU,oB;QAyBD,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC
;UACZ,UA1BoD,WA0B1C,CAAY,oBAAZ,C;U3BljBP,U;UADP,Y2BojBe,W3BpjBH,W2BojBwB,G3BpjBxB,C;
UACL,IAAI,aAAJ,C;YACH,a2BkjBuC,gB;YAA5B,W3BjjBX,a2BijBgC,G3BjjBhC,EAAS,MAAT,C;YACA,e;;Y
AEA,c;;U2B8iBA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QA5BT,OA8BO,W;O;KAvCX,C;gFAYA,yB;MAAA,wE;M
A8BA,6C;MAAA,oC;MAAA,+D;MAAA,gC;MA9BA,yD;QAUW,kBAAU,oB;QA8BD,Q;QAAA,0B;QAAhB,OA
AgB,cAAhB,C;UAAgB,oC;UACZ,UA/BiD,WA+BvC,CAAY,oBAAZ,C;U3BpkBP,U;UADP,Y2BskBe,W3BtkBH
,W2BskBwB,G3BtkBxB,C;UACL,IAAI,aAAJ,C;YACH,a2BokBuC,gB;YAA5B,W3BnkBX,a2BmkBgC,G3BnkBh
C,EAAS,MAAT,C;YACA,e;;YAEA,c;;U2BgkBA,iB;UACA,IAAK,WAjCyD,cAiCrD,CAAe,oBAAf,CAAJ,C;;QA
jCT,OAmCO,W;O;KA7CX,C;oFAaA,yB;MAAA,6C;MAAA,oC;MAAA,+D;MAAA,gC;MAAA,sD;QASoB,Q;QA
AA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,UAAU,YAAY,oBAAZ,C;U3BljBP,U;UADP,Y2BojBe,W3
BpjBH,W2BojBwB,G3BpjBxB,C;UACL,IAAI,aAAJ,C;YACH,a2BkjBuC,gB;YAA5B,W3BjjBX,a2BijBgC,G3Bjj
BhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;U2B8iBA,iB;UACA,IAAK,WAAI,oBAAJ,C;;QAET,OAAO,W;O;KAd
X,C;oFAiBA,yB;MAAA,6C;MAAA,oC;MAAA,+D;MAAA,gC;MAAA,sE;QAUoB,Q;QAAA,0B;QAAhB,OAAg
B,cAAhB,C;UAAgB,oC;UACZ,UAAU,YAAY,oBAAZ,C;U3BpkBP,U;UADP,Y2BskBe,W3BtkBH,W2BskBwB,
G3BtkBxB,C;UACL,IAAI,aAAJ,C;YACH,a2BokBuC,gB;YAA5B,W3BnkBX,a2BmkBgC,G3BnkBhC,EAAS,MA
AT,C;YACA,e;;YAEA,c;;U2BgkBA,iB;UACA,IAAK,WAAI,eAAe,oBAAf,CAAJ,C;;QAET,OAAO,W;O;KAfX,C
;qFAkBA,yB;MAAA,6C;MAAA,oC;MAAA,kC;MAAA,4C;MAAA,wE;QAQW,sC;QAAA,8C;O;MARX,oDASQ,
Y;QAAgD,OAAgB,SAAhB,oBAAgB,C;O;MATxE,iDAUQ,mB;QAAuC,gCAAY,oBAAZ,C;O;MAV/C,gF;MAA
A,yC;QAQI,2D;O;KARJ,C;wEAcA,yB;MAAA,gE;MAyEA,6C;MAAA,oC;MAAA,gC;MAzEA,uC;QAOW,kBAA
M,eAAa,gBAAb,C;QAuEA,Q;QAAA,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UACT,WAAY,WAxEmB,SAwEf,CAA
U,iBAAV,CAAJ,C;;QAxEhB,OAyEO,W;O;KAhFX,C;sFAUA,yB;MAAA,gE;MA+BA,6C;MAAA,oC;MAAA,gC;
MA/BA,uC;QAOW,kBAAa,eAAa,gBAAb,C;QAgCP,gB;QADb,YAAY,C;QACC,0B;QAAb,OAAa,cAAb,C;UAA
a,iC;UACT,WAAY,WAjC0B,SAiCtB,EAAU,cAAV,EAAU,sBAAV,WAAmB,iBAAnB,CAAJ,C;;QAjChB,OAkC
O,W;O;KAzCX,C;mGAUA,yB;MAAA,+D;MAUA,gC;MAwLA,6C;MAAA,oC;MAlMA,uC;QAOW,kBAAoB,gB;
QAkMd,gB;QADb,YAAY,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UAxLsB,U;UAAA,cAVQ,SAUR,EAwL
T,cAxLS,EAwLT,sBAxLS,WAwLA,iBAxLA,W;YAA6C,6B;;;QAVhF,OAWO,W;O;KAlBX,C;uGAUA,yB;MAA
A,gC;MAwLA,6C;MAAA,oC;MAxLA,oD;QA+LiB,gB;QADb,YAAY,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,
iC;UAxLsB,U;UAAA,yBAwLT,cAxLS,EAwLT,sBAxLS,WAwLA,iBAxLA,W;YAA6C,6B;;;QAChF,OAAO,W;O
;KARX,C;0FAWA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oD;QAQiB,UACiB,M;QAF9B,YAAY,C;QACC
,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UACT,WAAY,WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,iBAAnB,CA
AJ,C;;QAChB,OAAO,W;O;KAVX,C;qFAaA,yB;MAAA,+D;MAUA,gC;MA+IA,6C;MAAA,oC;MAzJA,uC;QAO
W,kBAAa,gB;QAsJJ,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UA9IK,U;UAAA,cARe,SAQf,CA8IQ,
oBA9IR,W;YAAsC,6B;;;QAR3D,OASO,W;O;KAhBX,C;yFAUA,yB;MAAA,gC;MA+IA,6C;MAAA,oC;MA/IA,o
D;QAmJoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UA9IK,U;UAAA,wBA8IQ,oBA9IR,W;YAAsC,
6B;;;QAC3D,OAAO,W;O;KANX,C;4EASA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oD;QAKiB,Q;QAAA,
0B;QAAb,OAAa,cAAb,C;UAAa,iC;UACT,WAAY,WAAI,UAAU,iBAAV,CAAJ,C;;QAChB,OAAO,W;O;KAPX,
C;IAe4B,4C;MAAA,mB;QAAE,iC;O;K;IAL9B,iC;MAKI,OAAO,qBAAiB,6BAAjB,C;K;wEAGX,yB;MAAA,6C;
MAAA,oC;MAAA,gC;MAAA,uC;QAUoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,C
AAC,UAAU,oBAAV,CAAL,C;YAAyB,OAAO,K;;QACtD,OAAO,I;O;KAXX,C;IAcA,2B;MAMI,OAAO,EC1wB
yC,qBAAU,CD0wBnD,C;K;wEAGX,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;QAAA,0B;QA
AhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,I;;QACrD,OAAO,K;
O;KAPX,C;4EAUA,qB;MAKI,OAAO,gB;K;4EAGX,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAKoB,Q
;QADhB,YAAY,C;QACI,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAA
wB,qB;;QAC9C,OAAO,K;O;KANX,C;0EASA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,gD;QAUoB,Q;QAD
hB,kBAAkB,O;QACF,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,cAAc,UAAU,WAAV,EAAuB,oBAAv
B,C;;QACpC,OAAO,W;O;KAXX,C;wFAcA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,gD;QAYoB,UAA8B,
M;QAF9C,YAAY,C;QACZ,kBAAkB,O;QACF,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,cAAc,WAAU,
cAAV,EAAU,sBAAV,WAAmB,WAAnB,EAAgC,oBAAhC,C;;QACpC,OAAO,W;O;KAbX,C;mFAgBA,yB;MAA

A,uD;MAAA,oC;MAAA,gD;QAYoC,Q;QAHhC,YAAY,wB;QACZ,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;
UACI,cAAc,UAAU,kCAAI,YAAJ,EAAI,oBAAJ,SAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAdX,C;iGAi
BA,yB;MAAA,uD;MAAA,oC;MAAA,gD;QAUI,YAAY,wB;QACZ,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;
UACI,cAAc,UAAU,KAAV,EAAiB,iCAAI,KAAJ,EAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAh
BX,C;gFAmBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oC;QAIoB,Q;QAAA,0B;QAAhB,OAAgB,cAAhB,
C;UAAgB,oC;UAAM,OAAO,oBAAP,C;;O;KAJ1B,C;8FAOA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oC;
QAOiB,UAAa,M;QAD1B,YAAY,C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UAAM,QAAO,cAAP,EAAO,sB
AAP,WAAgB,iBAAhB,C;;O;KAPvB,C;IAUA,2B;MAWiB,Q;MAFb,ICp4BgD,qBAAU,CDo4B1D,C;QAAe,MAA
M,6B;MACrB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QA
CR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;4EAGX,yB;MAAA,sE;MAAA,uD;MAAA,oC
;MAAA,sC;QAWI,ICx5BgD,qBAAU,CDw5B1D,C;UAAe,MAAM,6B;QACrB,cAAc,qBAAK,CAAL,C;QACd,gB
AAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,a
AAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,qBAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAA
W,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KAxBX,C;wFA2BA,yB;MAAA,uD;
MAAA,oC;MAAA,sC;QAOI,IC/6BgD,qBAAU,CD+6B1D,C;UAAe,OAAO,I;QACtB,cAAc,qBAAK,CAAL,C;QA
Cd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,oBAAT,C;Q
ACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,qBAAK,CAAL,C;UACR,QAAQ,SAAS,cAAT,C;UACR,IAAI,2
BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KApBX,C;4EAuBA,yB;MAA
A,sE;MAAA,oC;MAAA,uD;Md/pCA,iB;Mc+pCA,sC;QAeiB,Q;QAFb,IC58BgD,qBAAU,CD48B1D,C;UAAe,MA
AM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iC
AAK,CAAL,EAAT,C;UACR,WdxqCG,MAAO,KcwqCO,QdxqCP,EcwqCiB,CdxqCjB,C;;Qc0qCd,OAAO,Q;O;K
AnBX,C;4EAsBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MdhsCA,iB;McgsCA,sC;QAeiB,Q;QAFb,ICl+BgD,qBA
AU,CDk+B1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAA
V,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,WdzsCG,MAAO,KcysCO,QdzsCP,EcysCiB,CdzsCjB,
C;;Qc2sCd,OAAO,Q;O;KAnBX,C;4EAsBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MAAA,sC;QAaiB,Q;QAFb,ICt
/BgD,qBAAU,CDs/B1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aA
AU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAA
W,C;;;QAGnB,OAAO,Q;O;KAnBX,C;wFAsBA,yB;MAAA,oC;MAAA,uD;MdjuCA,iB;MciuCA,sC;QAaiB,Q;QA
Fb,IC5gCgD,qBAAU,CD4gC1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAA
b,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,WdxuCG,MAAO,KcwuCO,QdxuCP,Ecw
uCiB,CdxuCjB,C;;Qc0uCd,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,oC;MAAA,uD;MdhwCA,iB;McgwCA,sC;
QAaiB,Q;QAFb,IChiCgD,qBAAU,CDgiC1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QA
CF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,WdvwCG,MAAO,KcuwCO,
QdvwCP,EcuwCiB,CdvwCjB,C;;QcywCd,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,oC;MAAA,uD;MAAA,sC;
QAWiB,Q;QAFb,ICljCgD,qBAAU,CDkjC1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;Q
ACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,2BAAW,CAAX,KA
AJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;oFAoBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MAAA,k
D;QAaiB,Q;QAFb,ICxkCgD,qBAAU,CDwkC1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT
,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,
QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;gGAsBA,y
B;MAAA,oC;MAAA,uD;MAAA,kD;QAWiB,Q;QAFb,IC5lCgD,qBAAU,CD4lC1D,C;UAAe,OAAO,I;QACtB,eA
Ae,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT
,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OA
AO,Q;O;KAjBX,C;IAoBA,iC;MAOiB,Q;MAFb,IC5mCgD,qBAAU,CD4mC1D,C;QAAe,OAAO,I;MACtB,UAAU
,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,MAAM,CA
AV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAWiB,Q;MAFb,IC9nCgD,qBAAU,CD8nC1D,C;QAAe
,MAAM,6B;MACrB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,
C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAA

O,G;K;IAGX,iD;MAOiB,Q;MAFb,IC5oCgD,qBAAU,CD4oC1D,C;QAAe,OAAO,I;MACtB,UAAU,qBAAK,CAA
L,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EA
Aa,cAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2B;MAWiB,Q;MAFb,IC9pC
gD,qBAAU,CD8pC1D,C;QAAe,MAAM,6B;MACrB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,i
B;QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;4EAGX
,yB;MAAA,sE;MAAA,uD;MAAA,oC;MAAA,sC;QAWI,IClrCgD,qBAAU,CDkrC1D,C;UAAe,MAAM,6B;QACr
B,cAAc,qBAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QA
C3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,qBAAK,CAAL,C;UACR,QAAQ
,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;
KAxBX,C;wFA2BA,yB;MAAA,uD;MAAA,oC;MAAA,sC;QAOI,ICzsCgD,qBAAU,CDysC1D,C;UAAe,OAAO,I;
QACtB,cAAc,qBAAK,CAAL,C;QACd,gBAAqB,cAAL,SAAK,C;QACrB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,
O;QAC3B,eAAe,SAAS,oBAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,qBAAK,CAAL,C;UACR,
QAAQ,SAAS,cAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAA
O,O;O;KApBX,C;4EAuBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MdruCA,iB;McquCA,sC;QAeiB,Q;QAFb,ICtuC
gD,qBAAU,CDsuC1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAA
U,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,Wd9uCG,MAAO,Kc8uCO,Qd9uCP,Ec8uCiB,C
d9uCjB,C;;QcgvCd,OAAO,Q;O;KAnBX,C;4EAsBA,yB;MAAA,sE;MAAA,oC;MAAA,uD;MdtwCA,iB;McswCA,
sC;QAeiB,Q;QAFb,IC5vCgD,qBAAU,CD4vC1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,CAAL,EAA
T,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,Wd/wCG,MAAO,Kc
+wCO,Qd/wCP,Ec+wCiB,Cd/wCjB,C;;QcixCd,OAAO,Q;O;KAnBX,C;4EAsBA,yB;MAAA,sE;MAAA,oC;MAAA
,uD;MAAA,sC;QAaiB,Q;QAFb,IChxCgD,qBAAU,CDgxC1D,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,iCAAK,
CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,2B
AAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAnBX,C;wFAsBA,yB;MAAA,oC;MAAA,uD;M
dvyCA,iB;McuyCA,sC;QAaiB,Q;QAFb,ICtyCgD,qBAAU,CDsyC1D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCA
AK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C;UACR,Wd
9yCG,MAAO,Kc8yCO,Qd9yCP,Ec8yCiB,Cd9yCjB,C;;QcgzCd,OAAO,Q;O;KAjBX,C;wFAoBA,yB;MAAA,oC;
MAAA,uD;Mdt0CA,iB;Mcs0CA,sC;QAaiB,Q;QAFb,IC1zCgD,qBAAU,CD0zC1D,C;UAAe,OAAO,I;QACtB,eAA
e,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C
;UACR,Wd70CG,MAAO,Kc60CO,Qd70CP,Ec60CiB,Cd70CjB,C;;Qc+0Cd,OAAO,Q;O;KAjBX,C;wFAoBA,yB;
MAAA,oC;MAAA,uD;MAAA,sC;QAWiB,Q;QAFb,IC50CgD,qBAAU,CD40C1D,C;UAAe,OAAO,I;QACtB,eAA
e,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT,C
;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;oFAoBA,yB;MAAA,sE
;MAAA,oC;MAAA,uD;MAAA,kD;QAaiB,Q;QAFb,ICl2CgD,qBAAU,CDk2C1D,C;UAAe,MAAM,6B;QACrB,eA
Ae,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAAQ,SAAS,iCAAK,CAAL,EAAT
,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OA
AO,Q;O;KAnBX,C;gGAsBA,yB;MAAA,oC;MAAA,uD;MAAA,kD;QAWiB,Q;QAFb,ICt3CgD,qBAAU,CDs3C1
D,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,iCAAK,CAAL,EAAT,C;QACF,+B;QAAb,aAAU,CAAV,iB;UACI,QAA
Q,SAAS,iCAAK,CAAL,EAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;
YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAjBX,C;IAoBA,iC;MAOiB,Q;MAFb,ICt4CgD,qBAAU,CDs4C1D,C;Q
AAe,OAAO,I;MACtB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAA
L,C;QACR,IAAI,MAAM,CAAV,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAWiB,Q;MAFb,ICx5CgD
,qBAAU,CDw5C1D,C;QAAe,MAAM,6B;MACrB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;
QACI,QAAQ,qBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CAAjC,C;UA
AoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAOiB,Q;MAFb,ICt6CgD,qBAAU,CDs6C1D,C;QAAe,OAAO,I;
MACtB,UAAU,qBAAK,CAAL,C;MACG,kC;MAAb,aAAU,CAAV,iB;QACI,QAAQ,qBAAK,CAAL,C;QACR,IA
AI,UAAW,SAAQ,gBAAR,EAAa,cAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAG
X,4B;MAMI,OCr7CgD,qBAAU,C;K;0EDw7C9D,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,uC;QAMoB,Q;Q
AAA,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAAM,IAAI,UAAU,oBAAV,CAAJ,C;YAAwB,OAAO,K;;QAC

rD,OAAO,I;O;KAPX,C;8EAUA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,oC;QAKmC,Q;QAAA,0B;QAAhB
,OAAgB,cAAhB,C;UAAgB,oC;UAAM,OAAO,oBAAP,C;;QAArC,gB;O;KALJ,C;4FAQA,yB;MAAA,6B;MAAA,
sC;MAzlBA,6C;MAAA,oC;MAAA,gC;MAylBA,2BAQiB,yB;QAjmBjB,6C;QAAA,oC;QAAA,gC;eAimBiB,0B;U
AAA,4B;YAAE,aAAe,c;YA1lBjB,gB;YADb,YAAY,C;YACC,0B;YAAb,OAAa,cAAb,C;cAAa,iC;cAAM,QAAO,
cAAP,EAAO,sBAAP,WAAgB,iBAAhB,C;;YA0lBmB,W;W;S;OAAzB,C;MARjB,oC;QAllBiB,gB;QADb,YAAY,
C;QACC,0B;QAAb,OAAa,cAAb,C;UAAa,iC;UAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,iBAAhB,C;;QA0lBn
B,gB;O;KARJ,C;8EAWA,yB;MAAA,4F;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,ICl+
CgD,qBAAU,CDk+C1D,C;UACI,MAAM,mCAA8B,uCAA9B,C;QACV,kBAAkB,qBAAK,CAAL,C;QACD,+B;Q
AAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,wBAAV,EAAuB,iCAAK,KAAL,EAAvB,E;;QAElB,OAAO,W;O;K
AnBX,C;4FAsBA,yB;MAAA,4F;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,ICx/CgD,qB
AAU,CDw/C1D,C;UACI,MAAM,mCAA8B,uCAA9B,C;QACV,kBAAkB,qBAAK,CAAL,C;QACD,+B;QAAjB,i
BAAc,CAAd,yB;UACI,cAAc,oBAAU,KAAV,EAAiB,wBAAjB,EAA8B,iCAAK,KAAL,EAA9B,E;;QAElB,OAA
O,W;O;KAnBX,C;wGAsBA,yB;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgBqB,Q;QAHjB,IC9gDgD,qB
AAU,CD8gD1D,C;UACI,OAAO,I;QACX,kBAAkB,qBAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI
,cAAc,oBAAU,KAAV,EAAiB,wBAAjB,EAA8B,iCAAK,KAAL,EAA9B,E;;QAElB,OAAO,W;O;KAnBX,C;0FAs
BA,yB;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAiBqB,Q;QAHjB,ICriDgD,qBAAU,CDqiD1D,C;UACI,O
AAO,I;QACX,kBAAkB,qBAAK,CAAL,C;QACD,+B;QAAjB,iBAAc,CAAd,yB;UACI,cAAc,oBAAU,wBAAV,E
AAuB,iCAAK,KAAL,EAAvB,E;;QAElB,OAAO,W;O;KApBX,C;uFAuBA,yB;MAAA,uD;MAAA,4F;MAAA,oC;
MAAA,gC;MAAA,uC;QAe0B,UAEU,M;QAJhC,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA
8B,uCAA9B,C;QACrB,kBAAkB,sBAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,oB
AAU,kCAAI,cAAJ,EAAI,sBAAJ,WAAV,EAAwB,wBAAxB,E;;QAElB,OAAO,W;O;KAnBX,C;qGAsBA,yB;MA
AA,uD;MAAA,4F;MAAA,oC;MAAA,gC;MAAA,uC;QAe0B,Q;QAFtB,YAAY,wB;QACZ,IAAI,QAAQ,CAAZ,C;
UAAe,MAAM,mCAA8B,uCAA9B,C;QACrB,kBAAkB,sBAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CA
AhB,C;UACI,cAAc,oBAAU,KAAV,EAAiB,iCAAI,KAAJ,EAAjB,EAA6B,wBAA7B,E;UACd,qB;;QAEJ,OAAO,
W;O;KApBX,C;iHAuBA,yB;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAe0B,Q;QAFtB,YAAY,wB;QACZ,I
AAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAkB,sBAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CA
AhB,C;UACI,cAAc,oBAAU,KAAV,EAAiB,iCAAI,KAAJ,EAAjB,EAA6B,wBAA7B,E;UACd,qB;;QAEJ,OAAO,
W;O;KApBX,C;mGAuBA,yB;MAAA,uD;MAAA,oC;MAAA,gC;MAAA,uC;QAgB0B,UAEU,M;QAJhC,YAAY,
wB;QACZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,kBAAkB,sBAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAA
O,SAAS,CAAhB,C;UACI,cAAc,oBAAU,kCAAI,cAAJ,EAAI,sBAAJ,WAAV,EAAwB,wBAAxB,E;;QAElB,OAA
O,W;O;KApBX,C;wFAuBA,yB;MAAA,gD;MAAA,gE;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,gD;QAgBoB,Q
;QAHhB,ICtpDgD,qBAAU,CDspD1D,C;UAAe,OAAO,OAAO,OAAP,C;QACgB,kBAAzB,eAAa,mBAAS,CAAT,
IAAb,C;QAAiC,8B;QAA9C,af32DO,W;Qe42DP,kBAAkB,O;QACF,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;
UACZ,cAAc,UAAU,WAAV,EAAuB,oBAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KApBX,C;
sGAuBA,yB;MAAA,gD;MAAA,gE;MAAA,mD;MAAA,oC;MAAA,gD;QAiBkB,gC;QAHd,IC9qDgD,qBAAU,C
D8qD1D,C;UAAe,OAAO,OAAO,OAAP,C;QACgB,kBAAzB,eAAa,mBAAS,CAAT,IAAb,C;QAAiC,8B;QAA9C,
afn4DO,W;Qeo4DP,kBAAkB,O;QACJ,6B;QAAA,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,cAAc,UAAU,KAA
V,EAAiB,WAAjB,EAA8B,iCAAK,KAAL,EAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KArB
X,C;4FAwBA,yB;MAAA,qD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,uC;QAgB0B,Q;QAHtB,ICrsDgD,qBAA
U,CDqsD1D,C;UAAe,OAAO,W;QACtB,sBAAkB,qBAAK,CAAlB,C;QACqC,kBAAxB,eAAgB,gBAAhB,
C;QAAgC,sBAAI,0BAAJ,C;QAA7C,af35DO,W;Qe45De,uB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,oBAAU,0
BAAV,EAAuB,iCAAK,KAAL,EAAvB,E;UACd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KApBX,C;0GAu
BA,yB;MAAA,qD;MAAA,gE;MAAA,oC;MAAA,gC;MAAA,uC;QAiB0B,Q;QAHtB,IC7tDgD,qBAAU,CD6tD1D
,C;UAAe,OAAO,W;QACtB,sBAAkB,qBAAK,CAAlB,C;QACqC,kBAAxB,eAAgB,gBAAhB,C;QAAgC,s
BAAI,0BAAJ,C;QAA7C,afn7DO,W;Qeo7De,uB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,oBAAU,KAAV,EAAAi
B,0BAAjB,EAA8B,iCAAK,KAAL,EAA9B,E;UACd,MAAO,WAAI,0BAAJ,C;;QAEX,OAAO,M;O;KArBX,C;0E
AwBA,yB;MA9FA,gD;MAAA,gE;MAAA,6C;MAAA,oC;MAAA,gC;MA8FA,gD;QAcW,sB;;UA5FS,Q;UAHhB,I
CtpDgD,qBAAU,CDspD1D,C;YAAe,qBAAO,OA+FH,OA/FG,C;YAAP,uB;;UACuB,kBAAzB,eAAa,mBAAS,CA

AT,IAAb,C;UAAiC,sBA8F3B,OA9F2B,C;UAA9C,af32DO,W;Ue42DP,kBA6FmB,O;UA5FH,0B;UAAhB,OAAg
B,cAAhB,C;YAAgB,oC;YACZ,cA2FwB,SA3FV,CAAU,WAAV,EAAuB,oBAAvB,C;YACd,MAAO,WAAI,WAA
J,C;;UAEX,qBAAO,M;;;QAwFP,yB;O;KAdJ,C;wFAiBA,yB;MAxFA,gD;MAAA,gE;MAAA,mD;MAAA,oC;MA
wFA,gD;QAeW,6B;;UAtFO,gC;UAHd,IC9qDgD,qBAAU,CD8qD1D,C;YAAe,4BAAO,OAyFI,OAzFJ,C;YAAP,8
B;;UACuB,kBAAzB,eAAa,mBAAS,CAAT,IAAb,C;UAAiC,sBAwFpB,OAxFoB,C;UAA9C,afn4DO,W;Ueo4DP,k
BAuF0B,O;UAtFZ,6B;UAAA,mB;UAAA,kB;UAAA,kB;UAAd,0D;YACI,cAqF+B,SArFjB,CAAU,KAAV,EAAi
B,WAAjB,EAA8B,iCAAK,KAAL,EAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAkFP,gC;O;
KAfJ,C;4EAkBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAOoB,Q;QADhB,UAAe,C;QACC,0B;QAAh
B,OAAgB,cAAhB,C;UAAgB,oC;UACZ,YAAO,SAAS,oBAAT,CAAP,I;;QAEJ,OAAO,G;O;KAVX,C;wFAaA,yB;
MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sC;QAOoB,Q;QADhB,UAAkB,G;QACF,0B;QAAhB,OAAgB,cAAhB,
C;UAAgB,oC;UACZ,OAAO,SAAS,oBAAT,C;;QAEX,OAAO,G;O;KAVX,C;4EAaA,yB;MAAA,6C;MAAA,oC;
MAAA,gC;MAAA,sC;QAUoB,Q;QADhB,UAAoB,C;QACJ,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,O
AAO,SAAS,oBAAT,C;;QAEX,OAAO,G;O;KAbX,C;4EAgBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,sC;
QAUoB,Q;QADhB,UAAe,C;QACC,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,YAAO,SAAS,oBAAT,CA
AP,I;;QAEJ,OAAO,G;O;KAbX,C;4EAgBA,yB;MAAA,SASoB,gB;MATpB,6C;MAAA,oC;MAAA,gC;MAAA,sC;
QAUoB,Q;QADhB,Y;QACgB,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,cAAO,SAAS,oBAAT,CAAP,C;
;QAEJ,OAAO,G;O;KAbX,C;4EAgBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;M9BzuDA,6B;M8ByuDA,sC;QAWo
B,Q;QADhB,U9BzuDmC,c8ByuDnB,C9BzuDmB,C;Q8B0uDnB,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAC
Z,M9BvjEiD,c8BujEjD,G9BvjE2D,KAAK,G8BujEzD,SAAS,oBAAT,C9BvjEoE,KAAX,IAAf,C;;Q8ByjErD,OAA
O,G;O;KAdX,C;4EAiBA,yB;MAAA,6C;MAAA,oC;MAAA,gC;MbvvDA,+B;MauvDA,sC;QAWoB,Q;QADhB,U
btvDqC,eAAW,oBasvD/B,CbtvD+B,CAAX,C;QauvDrB,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,Mbrk
EmD,eaqkEnD,GbrkE8D,KAAK,KaqkE5D,SAAS,oBAAT,CbrkEuE,KAAX,CAAhB,C;;QaukEvD,OAAO,G;O;K
AdX,C;IAiBA,oC;MAWI,OAAO,sBAAS,IAAT,EAAe,IAAf,EAAsC,IAAtC,C;K;IAGX,+C;MAgBI,OAAO,sBAA
S,IAAT,EAAe,IAAf,EAAsC,IAAtC,EAAwD,SAAxD,C;K;IAcsB,oC;MAAE,OAAA,EAAG,W;K;IAXtC,0C;MA
WI,OAAO,6BAAgB,IAAhB,EAAsB,sBAAtB,C;K;IAGX,uD;MAgBI,OAAO,8BAAiB,IAAjB,EAAuB,IAAvB,EA
A8C,IAA9C,EAAgE,SAAhE,C;K;oFAGX,yB;MAAA,yD;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,6B;MAAA,u
C;QAUoB,Q;QAFhB,YAAY,oB;QACZ,aAAa,oB;QACG,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UACZ,IAAI
,UAAU,oBAAV,CAAJ,C;YACI,KAAM,gBAAO,OAAP,C;;YAEN,MAAO,gBAAO,OAAP,C;;;QAGf,OAAO,cAA
K,KAAL,EAAY,MAAZ,C;O;KAjBX,C;oFAoBA,yB;MAAA,yD;MAAA,6C;MAAA,oC;MAAA,gC;MAAA,6B;M
AAA,uC;QAUoB,Q;QAFhB,YAAY,oB;QACZ,aAAa,oB;QACG,0B;QAAhB,OAAgB,cAAhB,C;UAAgB,oC;UAC
Z,IAAI,UAAU,oBAAV,CAAJ,C;YACI,KAAM,gBAAO,OAAP,C;;YAEN,MAAO,gBAAO,OAAP,C;;;QAGf,OAA
O,cAAK,KAAM,WAAX,EAAuB,MAAO,WAA9B,C;O;KAjBX,C;IAqCgD,6B;MAAE,OAAA,EAAG,W;K;IAjBr
D,2D;MAgB4C,oB;QAAA,OAAY,C;MAAG,8B;QAAA,iBAA0B,K;MACjF,OAAO,sBAAS,IAAT,EAAe,IAAf,E
AAqB,cAArB,EAAqC,eAArC,C;K;IAGX,sE;MAkBgD,oB;QAAA,OAAY,C;MAAG,8B;QAAA,iBAA0B,K;MAQ
hE,Q;MAPrB,oBAAoB,IAApB,EAA0B,IAA1B,C;MACA,eAAe,SAAK,O;MACpB,qBAAqB,YAAW,IAAX,SAAs
B,YAAW,IAAX,UAAmB,CAAvB,GAA0B,CAA1B,GAAiC,CAAnD,K;MACrB,aAAa,iBAAa,cAAb,C;MACb,YA
AY,C;MACZ,OAAgB,CAAT,qBAAiB,QAAxB,C;QACI,UAAU,QAAQ,IAAR,I;QACO,IAAI,MAAM,CAAN,IAA
W,MAAM,QAArB,C;UAAiC,IAAI,cAAJ,C;YAAoB,e;;YAAc,K;;UAAa,U;QAAjG,qB;QACA,MAAO,WAAI,UA
AU,8BAAY,KAAZ,EAAmB,UAAnB,CAAV,CAAJ,C;QACP,gBAAS,IAAT,I;;MAEJ,OAAO,M;K;IAoB6C,qC;M
AAE,OAAA,EAAG,W;K;IAjB7D,iE;MAgBoD,oB;QAAA,OAAY,C;MAAG,8B;QAAA,iBAA0B,K;MACzF,OAA
O,8BAAiB,IAAjB,EAAuB,IAAvB,EAA6B,cAA7B,EAA6C,uBAA7C,C;K;IAwByB,2F;MAAAA,wB;QAC5B,UAA
U,QAAQ,YAAR,I;QACV,iBAAqB,MAAM,CAAN,IAAW,MAAM,4BAArB,GAA6B,4BAA7B,GAAyC,G;QAD1
D,OAEA,kBAAU,0CAAY,KAAZ,EAAmB,UAAnB,CAAV,C;O;K;IAxBR,gF;MAkBwD,sB;QAAA,SAAY,C;MA
AG,8B;QAAA,iBAA0B,K;MAC7F,oBAAoB,IAApB,EAA0B,MAA1B,C;MACA,cAAc,KAAK,cAAJ,GAAoB,yB
AApB,GAAAiC,WAAQ,mBAAS,IAAT,GAAAgB,CAAhB,IAAR,CAAlC,EAAkE,MAAlE,C;MACd,OAA4B,OAAb,
aAAR,OAAQ,CAAa,EAAI,qDAAJ,C;K;IAOhC,kC;MAkBI,ad1nEO,MAAO,Kc0nEU,gBd1nEV,Ec+mEH,KAW2
B,Od1nExB,C;Mc2nEd,WAAW,iBAAa,MAAb,C;MACX,aAAU,CAAV,MAAkB,MAAlB,M;QACI,IAAK,WAdq
B,GAcP,iCAAK,CAAL,EAdO,EAcE,YAdrB,KAcqB,YAAM,CAAN,EAdF,CAcrB,C;;MAdT,OAgBO,I;K;wEAbX

,yB;MAAA,gE;MAAA,oC;MdxnEA,iB;McwnEA,8C;QAQI,ad1nEO,MAAO,Kc0nEK,SAAK,Od1nEV,Ec0nEkB,
KAAM,Od1nExB,C;Qc2nEd,WAAW,eAAa,MAAb,C;QACX,aAAU,CAAV,MAAkB,MAAlB,M;UACI,IAAK,WA
AI,UAAU,iCAAK,CAAL,EAAV,EAAmB,6BAAM,CAAN,EAAnB,CAAJ,C;;QAET,OAAO,I;O;KAbX,C;IAgBA,
kC;MASW,sB;;QAaP,WAAW,mBAAS,CAAT,I;QACX,IAAI,OAAO,CAAX,C;UAAc,qBAAO,W;UAAP,uB;;QA
Cd,aAAa,iBAAa,IAAb,C;QACb,iBAAc,CAAd,UAAsB,IAAtB,U;UACI,MAAO,WAjBkB,GAiBJ,iCAAK,KAAL,
EAjBI,EAiBS,iCAAK,QAAQ,CAAR,IAAL,EAjBT,CAiBlB,C;;QAEX,qBAAO,M;;;MAnBP,yB;K;uFAGJ,yB;MA
AA,qD;MAAA,gE;MAAA,oC;MAAA,uC;QAUI,WAAW,mBAAS,CAAT,I;QACX,IAAI,OAAO,CAAX,C;UAAc,
OAAO,W;QACrB,aAAa,eAAa,IAAb,C;QACb,iBAAc,CAAd,UAAsB,IAAtB,U;UACI,MAAO,WAAI,UAAU,iCA
AK,KAAL,EAAV,EAAuB,iCAAK,QAAQ,CAAR,IAAL,EAAvB,CAAJ,C;;QAEX,OAAO,M;O;KAhBX,C;IAwBo
B,8C;MAAA,mB;QAAE,OAAK,WAAL,eAAK,C;O;K;IAL3B,kC;MAIQ,wC;MAAA,S;QAAkB,OCloE0B,qBAA
U,C;;MDkoE1D,S;QAAiC,OAAO,W;MACxC,oCAAgB,8BAAhB,C;K;IAQgB,8C;MAAA,mB;QAAE,OAAK,WA
AL,eAAK,C;O;K;IAL3B,kC;MAIQ,wC;MAAA,S;QAAkB,OC1oE0B,qBAAU,C;;MD0oE1D,S;QAAiC,OAAO,e;
MACxC,oCAAgB,8BAAhB,C;K;IEp2EkC,yC;MAAA,wB;QAAW,OAAA,aAAK,KAAL,CjCiMV,K;O;K;IkCjMH
,wC;MAAA,wB;QAAW,OAAA,aAAK,KAAL,CjCyOV,K;O;K;IkCzOC,yC;MAAA,wB;QAAW,OAAA,aAAK,K
AAL,CjBoPV,K;O;K;IkBpPC,0C;MAAA,wB;QAAW,OAAA,aAAK,KAAL,ClC4MV,K;O;K;4FmCtQzC,qB;MA
UI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CA
AJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;4FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;M
AUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,C
AAJ,C;K;4FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;
MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;4FAGX,qB;MAUI,OAAO,sBAAI,
CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,q
B;MAUI,OAAO,sBAAI,CAAJ,C;K;4FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBA
AI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;6FAGX,qB;MAUI,OAAO,sBAAI,CAAJ,C;K;uGAuC
X,yB;MAkhHI,8D;MAlhHJ,iD;QASe,oBAAS,C;QAAT,S;UAAc,gBAygHT,cAAR,iBAAQ,C;;QAzgHhB,OAAO,
OAAsC,sBAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KATjE,C;uGAYA,yB;MA8gHI,8D;MA9gHJ,iD;QASe,o
BAAS,C;QAAT,S;UAAc,gBAqgHT,cAAR,iBAAQ,C;;QArgHhB,OAAO,OAAsC,sBAAI,KAAJ,CAAtC,GAAsD,a
AAa,KAAb,C;O;KATjE,C;uGAYA,yB;MA0gHI,8D;MA1gHJ,iD;QASe,oBAAS,C;QAAT,S;UAAc,gBAigHT,cA
AR,iBAAQ,C;;QAjgHhB,OAAO,OAAsC,sBAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KATjE,C;uGAYA,yB;
MAsgHI,8D;MAtgHJ,iD;QASe,oBAAS,C;QAAT,S;UAAc,gBA6/GT,cAAR,iBAAQ,C;;QA7/GhB,OAAO,OAAsC,
sBAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KATjE,C;uGAYA,yB;MAAA,sD;MAAA,mC;QASI,OAAY,UA
AL,SAAK,EAAU,KAAV,C;O;KAThB,C;uGAYA,yB;MAAA,sD;MAAA,mC;QASI,OAAY,UAAL,SAAK,EAAU,
KAAV,C;O;KAThB,C;uGAYA,yB;MAAA,sD;MAAA,mC;QASI,OAAY,UAAL,SAAK,EAAU,KAAV,C;O;KATh
B,C;uGAYA,yB;MAAA,sD;MAAA,mC;QASI,OAAY,UAAL,SAAK,EAAU,KAAV,C;O;KAThB,C;iFAYA,gC;M
ASW,sB;;QAkOS,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAlOH,SAkOO,CAAU,OAAV,C
AAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MAnOP,yB;K;iFAGJ,gC;MASW,sB;;QAiOS,Q;QAA
A,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAjOH,SAiOO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;
YAAP,uB;;;QAC9C,qBAAO,I;;;MAlOP,yB;K;iFAGJ,gC;MASW,sB;;QAgOS,Q;QAAA,2B;QAAhB,OAAgB,cAA
hB,C;UAAgB,yB;UAAM,IAhOH,SAgOO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBA
AO,I;;;MAjOP,yB;K;iFAGJ,gC;MASW,sB;;QA+NS,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM
,IA/NH,SA+NO,CAAU,OAAV,CAAJ,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MAhOP,yB;K;yFA
GJ,yB;MAgoBA,+C;MAkuFI,0D;MAl2GJ,uC;QASW,qB;;UAgoBO,Q;UAAA,OAAa,SAytFX,YAAR,iBAAQ,CA
ztFW,CAAb,W;UAAd,OAAc,cAAd,C;YAAc,uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IAloBc,SAkoBV,CAAU,
OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QApoBP,wB;O;KATJ,C;yFAYA,yB;MAooB
A,+C;MA0tFI,0D;MA91GJ,uC;QASW,qB;;UAooBO,Q;UAAA,OAAa,SAitFX,YAAR,iBAAQ,CAjtFW,CAAb,W;
UAAd,OAAc,cAAd,C;YAAc,uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IAtoBc,SAsoBV,CAAU,OAAV,CAAJ,C;
cAAwB,oBAAO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QAxoBP,wB;O;KATJ,C;yFAYA,yB;MAwoBA,+C;MAktFI,0D
;MA11GJ,uC;QASW,qB;;UAwoBO,Q;UAAA,OAAa,SAysFX,YAAR,iBAAQ,CAzsFW,CAAb,W;UAAd,OAAc,c
AAd,C;YAAc,uB;YACV,cAAc,sBAAK,KAAL,C;YACd,IA1oBc,SA0oBV,CAAU,OAAV,CAAJ,C;cAAwB,oBA

AO,O;cAAP,sB;;;UAE5B,oBAAO,I;;;QA5oBP,wB;O;KATJ,C;yFAYA,yB;MA4oBA,+C;MA0sFI,0D;MAt1GJ,uC;
QASW,qB;;UA4oBO,Q;UAAA,OAAa,SAisFX,YAAR,iBAAQ,CAjsFW,CAAb,W;UAAd,OAAc,cAAd,C;YAAc,u
B;YACV,cAAc,sBAAK,KAAL,C;YACd,IA9oBc,SA8oBV,CAAU,OAAV,CAAJ,C;cAAwB,oBAAO,O;cAAP,sB;;
;UAE5B,oBAAO,I;;;QAhpBP,wB;O;KATJ,C;mFAYA,yB;MAAA,8C;MpC1GA,6B;MoC0GA,4B;QASI,OpCzGm
C,coCyGpB,MAAR,iBAAQ,CpCzGoB,C;O;KoCgGvC,C;mFAYA,yB;MAAA,8C;MnBvGA,+B;MmBuGA,4B;QA
SI,OnBtGsC,emBsGvB,MAAR,iBAAQ,CnBtGuB,C;O;KmB6F1C,C;mFAYA,yB;MAAA,8C;MrChLA,+B;MqCgL
A,4B;QASI,OrC/KsC,eqC+KvB,MAAR,iBAAQ,CrC/KuB,C;O;KqCsK1C,C;mFAYA,yB;MAAA,8C;MnC/KA,iC;
MmC+KA,4B;QASI,OnC9KyC,gBmC8K1B,MAAR,iBAAQ,CnC9K0B,C;O;KmCqK7C,C;mFAYA,yB;MAAA,iE
;MAAA,uC;QAQoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;
YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KATV,C;mFAYA,yB;MAAA,iE;MAAA,uC;QAQo
B,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;
;QACrD,MAAM,gCAAuB,mDAAvB,C;O;KATV,C;mFAYA,yB;MAAA,iE;MAAA,uC;QAQoB,Q;QAAA,2B;QA
AhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,g
CAAuB,mDAAvB,C;O;KATV,C;mFAYA,yB;MAAA,iE;MAAA,uC;QAQoB,Q;QAAA,2B;QAAhB,OAAgB,cAA
hB,C;UAAgB,yB;UAAM,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QACrD,MAAM,gCAAuB,mDAAvB,
C;O;KATV,C;IAYA,mC;MAMI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,CAAL,C;K;IAGpC,mC;MAMI,O
AAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,CAAL,C;K;IAGpC,mC;MAMI,OAAW,mBAAJ,GAAe,IAAf,GAAyB
,sBAAK,CAAL,C;K;IAGpC,mC;MAMI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,CAAL,C;K;+FAGpC,gC;
MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,O
AAO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,I
AAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;+FAGX,gC;MAOoB,Q;MAAA,2B;MAAhB,
OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,O;;MACrD,OAAO,I;K;+F
AGX,gC;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;U
AAwB,OAAO,O;;MACrD,OAAO,I;K;2FAGX,yB;MAkqGI,8D;MAlqGJ,iD;QAOe,oBAAS,C;QAAT,S;UAAc,gB
A2pGT,cAAR,iBAAQ,C;;QA3pGhB,OAAO,OAAsC,sBAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;2
FAUA,yB;MAgqGI,8D;MAhqGJ,iD;QAOe,oBAAS,C;QAAT,S;UAAc,gBAypGT,cAAR,iBAAQ,C;;QAzpGhB,O
AAO,OAAsC,sBAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;2FAUA,yB;MA8pGI,8D;MA9pGJ,iD;Q
AOe,oBAAS,C;QAAT,S;UAAc,gBAupGT,cAAR,iBAAQ,C;;QAvpGhB,OAAO,OAAsC,sBAAI,KAAJ,CAAtC,G
AAsD,aAAa,KAAb,C;O;KAPjE,C;2FAUA,yB;MA4pGI,8D;MA5pGJ,iD;QAOe,oBAAS,C;QAAT,S;UAAc,gBAqp
GT,cAAR,iBAAQ,C;;QArpGhB,OAAO,OAAsC,sBAAI,KAAJ,CAAtC,GAAsD,aAAa,KAAb,C;O;KAPjE,C;IAUA
,wC;MAQe,oBAAS,C;MAAT,S;QAAc,gBAknGT,gBAAR,iBAAQ,C;;MAlnGhB,OAAO,OAAsC,sBAAI,KAAJ,C
AAtC,GAAsD,I;K;IAGjE,wC;MAQe,oBAAS,C;MAAT,S;QAAc,gBA+mGT,gBAAR,iBAAQ,C;;MA/mGhB,OAA
O,OAAsC,sBAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,wC;MAQe,oBAAS,C;MAAT,S;QAAc,gBA4mGT,gBAAR,i
BAAQ,C;;MA5mGhB,OAAO,OAAsC,sBAAI,KAAJ,CAAtC,GAAsD,I;K;IAGjE,wC;MAQe,oBAAS,C;MAAT,S;
QAAc,gBAymGT,gBAAR,iBAAQ,C;;MAzmGhB,OAAO,OAAsC,sBAAI,KAAJ,CAAtC,GAAsD,I;K;uFAGjE,yB;
MAAA,kD;MAAA,qC;QAOI,OAAe,QAAR,iBAAQ,EAAQ,OpChdU,KoCgdlB,C;O;KAPnB,C;uFAUA,yB;MAA
A,kD;MAAA,qC;QAOI,OAAe,QAAR,iBAAQ,EAAQ,OnB/cY,KmB+cpB,C;O;KAPnB,C;uFAUA,yB;MAAA,kD;
MAAA,qC;QAOI,OAAe,QAAR,iBAAQ,EAAQ,OrC5gBY,KqC4gBpB,C;O;KAPnB,C;uFAUA,yB;MAAA,kD;M
AAA,qC;QAOI,OAAe,QAAR,iBAAQ,EAAQ,OnC3gBc,KmC2gBtB,C;O;KAPnB,C;iGAUA,yB;MAAA,sC;MpCt
ZA,6B;MoCsZA,0BAOgC,yB;QpC7ZhC,6B;eoC6ZgC,6B;UAAA,qB;YAAE,yBpCnZK,coCmZK,EpCnZL,CoCm
ZL,C;W;S;OAAF,C;MAPhC,uC;QAOmB,kBAAR,iB;QAAQ,uB;;UvC+0Bf,0D;YACI,IuCh1B0B,UpCnZK,cHmu
CjB,YAAK,KAAL,CGnuCiB,CoCmZL,CvCg1B1B,C;cACI,sBAAO,K;cAAP,wB;;;UAGR,sBAAO,E;;;QuCp1BP,
0B;O;KAPJ,C;iGAUA,yB;MAAA,sC;MnBjZA,+B;MmBiZA,0BAOgC,yB;QnBxZhC,+B;emBwZgC,6B;UAAA,q
B;YAAE,yBnB9YQ,emB8YE,EnB9YF,CmB8YR,C;W;S;OAAF,C;MAPhC,uC;QAOmB,kBAAR,iB;QAAQ,uB;;U
vCi1Bf,0D;YACI,IuCl1B0B,UnB9YQ,epBguCpB,YAAK,KAAL,CoBhuCoB,CmB8YR,CvCk1B1B,C;cACI,sBAA
O,K;cAAP,wB;;;UAGR,sBAAO,E;;;QuCt1BP,0B;O;KAPJ,C;iGAUA,yB;MAAA,sC;MrCxdA,+B;MqCwdA,0BAO
gC,yB;QrC/dhC,+B;eqC+dgC,6B;UAAA,qB;YAAE,yBrCrdQ,eqCqdE,ErCrdF,CqCqdR,C;W;S;OAAF,C;MAPhC,
uC;QAOmB,kBAAR,iB;QAAQ,uB;;UvCmyBf,0D;YACI,IuCpyB0B,UrCrdQ,eFyvCpB,YAAK,KAAL,CEzvCoB,

CqCqdR,CvCoyB1B,C;cACI,sBAAO,K;cAAP,wB;;;UAGR,sBAAO,E;;;QuCxyBP,0B;O;KAPJ,C;iGAUA,yB;MA
AA,sC;MnCrdA,iC;MmCqdA,0BAOgC,yB;QnC5dhC,iC;emC4dgC,6B;UAAA,qB;YAAE,yBnCldW,gBmCkdD,En
CldC,CmCkdX,C;W;S;OAAF,C;MAPhC,uC;QAOmB,kBAAR,iB;QAAQ,uB;;UvCqyBf,0D;YACI,IuCtyB0B,UnCl
dW,gBJwvCvB,YAAK,KAAL,CIxvCuB,CmCkdX,CvCsyB1B,C;cACI,sBAAO,K;cAAP,wB;;;UAGR,sBAAO,E;;;
QuC1yBP,0B;O;KAPJ,C;+FAUA,yB;MAAA,sC;MvCs5BA,0D;MAAA,+C;MGp1CA,6B;MoC8bA,yBAO+B,yB;
QpCrc/B,6B;eoCqc+B,6B;UAAA,qB;YAAE,yBpC3bM,coC2bI,EpC3bJ,CoC2bN,C;W;S;OAAF,C;MAP/B,uC;QA
OmB,kBAAR,iB;QAAQ,sB;;UvCm5BD,Q;UAAA,OAAQ,SAAR,wBAAQ,CAAR,W;UAAd,OAAc,cAAd,C;YAA
c,uB;YACV,IuCp5ByB,UpC3bM,cH+0CjB,YAAK,KAAL,CG/0CiB,CoC2bN,CvCo5BzB,C;cACI,qBAAO,K;cAA
P,uB;;;UAGR,qBAAO,E;;;QuCx5BP,yB;O;KAPJ,C;+FAUA,yB;MAAA,sC;MvCw5BA,0D;MAAA,+C;MoBj1CA,
+B;MmBybA,yBAO+B,yB;QnBhc/B,+B;emBgc+B,6B;UAAA,qB;YAAE,yBnBtbS,emBsbC,EnBtbD,CmBsbT,C;
W;S;OAAF,C;MAP/B,uC;QAOmB,kBAAR,iB;QAAQ,sB;;UvCq5BD,Q;UAAA,OAAQ,SAAR,wBAAQ,CAAR,W
;UAAd,OAAc,cAAd,C;YAAc,uB;YACV,IuCt5ByB,UnBtbS,epB40CpB,YAAK,KAAL,CoB50CoB,CmBsbT,CvCs
5BzB,C;cACI,qBAAO,K;cAAP,uB;;;UAGR,qBAAO,E;;;QuC15BP,yB;O;KAPJ,C;+FAUA,yB;MAAA,sC;MvC02
BA,0D;MAAA,+C;ME12CA,+B;MqCggBA,yBAO+B,yB;QrCvgB/B,+B;eqCugB+B,6B;UAAA,qB;YAAE,yBrC7f
S,eqC6fC,ErC7fD,CqC6fT,C;W;S;OAAF,C;MAP/B,uC;QAOmB,kBAAR,iB;QAAQ,sB;;UvCu2BD,Q;UAAA,OA
AQ,SAAR,wBAAQ,CAAR,W;UAAd,OAAc,cAAd,C;YAAc,uB;YACV,IuCx2ByB,UrC7fS,eFq2CpB,YAAK,KAA
L,CEr2CoB,CqC6fT,CvCw2BzB,C;cACI,qBAAO,K;cAAP,uB;;;UAGR,qBAAO,E;;;QuC52BP,yB;O;KAPJ,C;+FA
UA,yB;MAAA,sC;MvC42BA,0D;MAAA,+C;MIz2CA,iC;MmC6fA,yBAO+B,yB;QnCpgB/B,iC;emCogB+B,6B;U
AAA,qB;YAAE,yBnC1fY,gBmC0fF,EnC1fE,CmC0fZ,C;W;S;OAAF,C;MAP/B,uC;QAOmB,kBAAR,iB;QAAQ,s
B;;UvCy2BD,Q;UAAA,OAAQ,SAAR,wBAAQ,CAAR,W;UAAd,OAAc,cAAd,C;YAAc,uB;YACV,IuC12ByB,Un
C1fY,gBJo2CvB,YAAK,KAAL,CIp2CuB,CmC0fZ,CvC02BzB,C;cACI,qBAAO,K;cAAP,uB;;;UAGR,qBAAO,E;;;
QuC92BP,yB;O;KAPJ,C;iFAUA,yB;MAAA,4C;MpCteA,6B;MoCseA,4B;QAWI,OpCvemC,coCuepB,KAAR,iBA
AQ,CpCveoB,C;O;KoC4dvC,C;iFAcA,yB;MAAA,4C;MnBreA,+B;MmBqeA,4B;QAWI,OnBtesC,emBsevB,KAA
R,iBAAQ,CnBteuB,C;O;KmB2d1C,C;iFAcA,yB;MAAA,4C;MrChjBA,+B;MqCgjBA,4B;QAWI,OrCjjBsC,eqCijB
vB,KAAR,iBAAQ,CrCjjBuB,C;O;KqCsiB1C,C;iFAcA,yB;MAAA,4C;MnCjjBA,iC;MmCijBA,4B;QAWI,OnCljB
yC,gBmCkjB1B,KAAR,iBAAQ,CnCljB0B,C;O;KmCuiB7C,C;iFAcA,yB;MAAA,+C;MAAA,iE;MA83FI,0D;MA9
3FJ,uC;QAWkB,Q;QAAA,OAAa,SAm3FX,YAn3FF,SAm3FN,QAAQ,CAn3FW,CAAb,W;QAAd,OAAc,cAAd,C;
UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAA
M,gCAAuB,mDAAvB,C;O;KAfV,C;iFAkBA,yB;MAAA,+C;MAAA,iE;MAo3FI,0D;MAp3FJ,uC;QAWkB,Q;QA
AA,OAAa,SAy2FX,YAz2FF,SAy2FN,QAAQ,CAz2FW,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,
sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,
C;O;KAfV,C;iFAkBA,yB;MAAA,+C;MAAA,iE;MA02FI,0D;MA12FJ,uC;QAWkB,Q;QAAA,OAAa,SA+1FX,YA
/1FF,SA+1FN,QAAQ,CA/1FW,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,
IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KAfV,C;iFAkBA,yB;
MAAA,+C;MAAA,iE;MAg2FI,0D;MAh2FJ,uC;QAWkB,Q;QAAA,OAAa,SAq1FX,YAr1FF,SAq1FN,QAAQ,CAr
1FW,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAA
J,C;YAAwB,OAAO,O;;QAEnC,MAAM,gCAAuB,mDAAvB,C;O;KAfV,C;+FAkBA,yB;MAAA,0D;MAAA,qC;Q
AOI,OAAe,YAAR,iBAAQ,EAAY,OpCxsBM,KoCwsBlB,C;O;KAPnB,C;+FAUA,yB;MAAA,0D;MAAA,qC;QA
OI,OAAe,YAAR,iBAAQ,EAAY,OnBvsBQ,KmBusBpB,C;O;KAPnB,C;+FAUA,yB;MAAA,0D;MAAA,qC;QAOI,
OAAe,YAAR,iBAAQ,EAAY,OrCpwBQ,KqCowBpB,C;O;KAPnB,C;+FAUA,yB;MAAA,0D;MAAA,qC;QAOI,O
AAe,YAAR,iBAAQ,EAAY,OnCnwBU,KmCmwBtB,C;O;KAPnB,C;IAUA,kC;MAQI,OAAW,mBAAJ,GAAe,IA
Af,GAAyB,sBAAK,iBAAO,CAAP,IAAL,C;K;IAGpC,kC;MAQI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,i
BAAO,CAAP,IAAL,C;K;IAGpC,kC;MAQI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,iBAAO,CAAP,IAAL,
C;K;IAGpC,kC;MAQI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,sBAAK,iBAAO,CAAP,IAAL,C;K;6FAGpC,yB;MA
AA,+C;MAkuFI,0D;MAluFJ,uC;QASkB,Q;QAAA,OAAa,SAytFX,YAztFF,SAytFN,QAAQ,CAztFW,CAAb,W;Q
AAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OA
AO,O;;QAEnC,OAAO,I;O;KAbX,C;6FAgBA,yB;MAAA,+C;MA0tFI,0D;MA1tFJ,uC;QASkB,Q;QAAA,OAAa,S
AitFX,YAjtFF,SAitFN,QAAQ,CAjtFW,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,

C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAbX,C;6FAgBA,yB;MAAA,+C;
MAktFI,0D;MAltFJ,uC;QASkB,Q;QAAA,OAAa,SAysFX,YAzsFF,SAysFN,QAAQ,CAzsFW,CAAb,W;QAAd,O
AAc,cAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UACd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;
QAEnC,OAAO,I;O;KAbX,C;6FAgBA,yB;MAAA,+C;MA0sFI,0D;MA1sFJ,uC;QASkB,Q;QAAA,OAAa,SAisFX,
YAjsFF,SAisFN,QAAQ,CAjsFW,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,cAAc,sBAAK,KAAL,C;UA
Cd,IAAI,UAAU,OAAV,CAAJ,C;YAAwB,OAAO,O;;QAEnC,OAAO,I;O;KAbX,C;qFAgBA,yB;MAAA,mC;MAA
A,gD;MAAA,4B;QASI,OAAO,kBAAO,cAAP,C;O;KATX,C;qFAYA,yB;MAAA,mC;MAAA,gD;MAAA,4B;QAS
I,OAAO,kBAAO,cAAP,C;O;KATX,C;qFAYA,yB;MAAA,mC;MAAA,gD;MAAA,4B;QASI,OAAO,kBAAO,cAA
P,C;O;KATX,C;qFAYA,yB;MAAA,mC;MAAA,gD;MAAA,4B;QASI,OAAO,kBAAO,cAAP,C;O;KATX,C;IAYA
,sC;MAQI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,sBAAI,MAAO,iBAAQ,cAAR,CA
AX,C;K;IAGX,sC;MAQI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,sBAAI,MAAO,iB
AAQ,cAAR,CAAX,C;K;IAGX,sC;MAQI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAvB,C;MACV,OAAO,sB
AAI,MAAO,iBAAQ,cAAR,CAAX,C;K;IAGX,sC;MAQI,IAAI,mBAAJ,C;QACI,MAAM,2BAAuB,iBAAvB,C;M
ACV,OAAO,sBAAI,MAAO,iBAAQ,cAAR,CAAX,C;K;iGAGX,yB;MAAA,mC;MAAA,4D;MAAA,4B;QAQI,OA
AO,wBAAa,cAAb,C;O;KARX,C;iGAWA,yB;MAAA,mC;MAAA,4D;MAAA,4B;QAQI,OAAO,wBAAa,cAAb,C;
O;KARX,C;iGAWA,yB;MAAA,mC;MAAA,4D;MAAA,4B;QAQI,OAAO,wBAAa,cAAb,C;O;KARX,C;iGAWA,
yB;MAAA,mC;MAAA,4D;MAAA,4B;QAQI,OAAO,wBAAa,cAAb,C;O;KARX,C;IAWA,4C;MAOI,IAAI,mBAA
J,C;QACI,OAAO,I;MACX,OAAO,sBAAI,MAAO,iBAAQ,cAAR,CAAX,C;K;IAGX,4C;MAOI,IAAI,mBAAJ,C;Q
ACI,OAAO,I;MACX,OAAO,sBAAI,MAAO,iBAAQ,cAAR,CAAX,C;K;IAGX,4C;MAOI,IAAI,mBAAJ,C;QACI,
OAAO,I;MACX,OAAO,sBAAI,MAAO,iBAAQ,cAAR,CAAX,C;K;IAGX,4C;MAOI,IAAI,mBAAJ,C;QACI,OAA
O,I;MACX,OAAO,sBAAI,MAAO,iBAAQ,cAAR,CAAX,C;K;qFAGX,yB;MAAA,gD;MpC17BA,6B;MoC07BA,4
B;QAOI,OpCv7BmC,coCu7BpB,OAAR,iBAAQ,CpCv7BoB,C;O;KoCg7BvC,C;qFAUA,yB;MAAA,gD;MnBr7B
A,+B;MmBq7BA,4B;QAOI,OnBl7BsC,emBk7BvB,OAAR,iBAAQ,CnBl7BuB,C;O;KmB26B1C,C;qFAUA,yB;M
AAA,gD;MrC5/BA,+B;MqC4/BA,4B;QAOI,OrCz/BsC,eqCy/BvB,OAAR,iBAAQ,CrCz/BuB,C;O;KqCk/B1C,C;q
FAUA,yB;MAAA,gD;MnCz/BA,iC;MmCy/BA,4B;QAOI,OnCt/ByC,gBmCs/B1B,OAAR,iBAAQ,CnCt/B0B,C;O;
KmC++B7C,C;qFAUA,yB;MAAA,kF;MAAA,iE;MAAA,wB;MAAA,8B;MAAA,uC;QASoB,UAST,M;QAXP,aA
AoB,I;QACpB,YAAY,K;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;
YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CA
AC,KAAL,C;UAAY,MAAM,gCAAuB,mDAAvB,C;QAElB,OAAO,0D;O;KAlBX,C;qFAqBA,yB;MAAA,kF;MA
AA,iE;MAAA,0B;MAAA,8B;MAAA,uC;QASoB,UAST,M;QAXP,aAAqB,I;QACrB,YAAY,K;QACI,2B;QAAhB,
OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAy
B,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAY,MAAM,gCAAuB,mDA
AvB,C;QAElB,OAAO,2D;O;KAlBX,C;qFAqBA,yB;MAAA,kF;MAAA,iE;MAAA,0B;MAAA,8B;MAAA,uC;QA
SoB,UAST,M;QAXP,aAAqB,I;QACrB,YAAY,K;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,
UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAA
Q,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAY,MAAM,gCAAuB,mDAAvB,C;QAElB,OAAO,2D;O;KAlBX,C;qFAq
BA,yB;MAAA,kF;MAAA,iE;MAAA,4B;MAAA,8B;MAAA,uC;QASoB,UAST,M;QAXP,aAAsB,I;QACtB,YAA
Y,K;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,UAAU,OAAV,CAAJ,C;YACI,IAAI,KAAJ,C;
cAAW,MAAM,8BAAyB,gDAAzB,C;YACjB,SAAS,O;YACT,QAAQ,I;;;QAGhB,IAAI,CAAC,KAAL,C;UAAY,
MAAM,gCAAuB,mDAAvB,C;QAElB,OAAO,4D;O;KAlBX,C;IAqBA,oC;MAMI,OAAW,mBAAQ,CAAZ,GAAe,
sBAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,oC;MAMI,OAAW,mBAAQ,CAAZ,GAAe,sBAAK,CAAL,CAAf,GAA
4B,I;K;IAGvC,oC;MAMI,OAAW,mBAAQ,CAAZ,GAAe,sBAAK,CAAL,CAAf,GAA4B,I;K;IAGvC,oC;MAMI,O
AAW,mBAAQ,CAAZ,GAAe,sBAAK,CAAL,CAAf,GAA4B,I;K;iGAGvC,gC;MASoB,Q;MAFhB,aAAoB,I;MAC
pB,YAAY,K;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI
,KAAJ,C;YAAW,OAAO,I;UAClB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;M
ACnB,OAAO,M;K;iGAGX,gC;MASoB,Q;MAFhB,aAAqB,I;MACrB,YAAY,K;MACI,2B;MAAhB,OAAgB,cAAh
B,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UAClB,SAAS,O;UA
CT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MASoB,Q;MAF

hB,aAAqB,I;MACrB,YAAY,K;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,C
AAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;UAClB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;
QAAY,OAAO,I;MACnB,OAAO,M;K;iGAGX,gC;MASoB,Q;MAFhB,aAAsB,I;MACtB,YAAY,K;MACI,2B;MA
AhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,IAAI,KAAJ,C;YAAW,OAAO,I;
UAClB,SAAS,O;UACT,QAAQ,I;;;MAGhB,IAAI,CAAC,KAAL,C;QAAY,OAAO,I;MACnB,OAAO,M;K;IAGX,+
B;MxBzhDI,IAAI,EwBmiDI,KAAK,CxBniDT,CAAJ,C;QACI,cwBkiDc,sD;QxBjiDd,MAAM,gCAAyB,OAAQ,W
AAjC,C;;MwBkiDV,OAAO,uBAAoB,gBAAV,iBAAO,CAAP,IAAU,EAAc,CAAd,CAApB,C;K;IAGX,+B;MxBvi
DI,IAAI,EwBijDI,KAAK,CxBjjDT,CAAJ,C;QACI,cwBgjDc,sD;QxB/iDd,MAAM,gCAAyB,OAAQ,WAAjC,C;;M
wBgjDV,OAAO,uBAAoB,gBAAV,iBAAO,CAAP,IAAU,EAAc,CAAd,CAApB,C;K;IAGX,+B;MxBrjDI,IAAI,Ew
B+jDI,KAAK,CxB/jDT,CAAJ,C;QACI,cwB8jDc,sD;QxB7jDd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwB8jDV,O
AAO,uBAAoB,gBAAV,iBAAO,CAAP,IAAU,EAAc,CAAd,CAApB,C;K;IAGX,+B;MxBnkDI,IAAI,EwB6kDI,K
AAK,CxB7kDT,CAAJ,C;QACI,cwB4kDc,sD;QxB3kDd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwB4kDV,OAAO,
uBAAoB,gBAAV,iBAAO,CAAP,IAAU,EAAc,CAAd,CAApB,C;K;IAGX,mC;MxBjlDI,IAAI,EwB2lDI,KAAK,C
xB3lDT,CAAJ,C;QACI,cwB0lDc,sD;QxBzlDd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwB0lDV,OAAO,mBAAg
B,gBAAV,iBAAO,CAAP,IAAU,EAAc,CAAd,CAAhB,C;K;IAGX,mC;MxB/lDI,IAAI,EwBymDI,KAAK,CxBzm
DT,CAAJ,C;QACI,cwBwmDc,sD;QxBvmDd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBwmDV,OAAO,mBAAg
B,gBAAV,iBAAO,CAAP,IAAU,EAAc,CAAd,CAAhB,C;K;IAGX,mC;MxB7mDI,IAAI,EwBunDI,KAAK,CxBvn
DT,CAAJ,C;QACI,cwBsnDc,sD;QxBrnDd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBsnDV,OAAO,mBAAgB,gB
AAV,iBAAO,CAAP,IAAU,EAAc,CAAd,CAAhB,C;K;IAGX,mC;MxB3nDI,IAAI,EwBqoDI,KAAK,CxBroDT,C
AAJ,C;QACI,cwBooDc,sD;QxBnoDd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBooDV,OAAO,mBAAgB,gBAA
V,iBAAO,CAAP,IAAU,EAAc,CAAd,CAAhB,C;K;mGAGX,yB;MAAA,4C;MAAA,qD;MAkqEI,8D;MAlqEJ,uC;
QASI,iBAypEgB,cAAR,iBAAQ,CAzpEhB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,
CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAdX,C;mGAiBA,yB;MAAA,4C;
MAAA,qD;MAypEI,8D;MAzpEJ,uC;QASI,iBAgpEgB,cAAR,iBAAQ,CAhpEhB,WAA+B,CAA/B,U;UACI,IAAI,
CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;
O;KAdX,C;mGAiBA,yB;MAAA,4C;MAAA,qD;MAgpEI,8D;MAhpEJ,uC;QASI,iBAuoEgB,cAAR,iBAAQ,CAvo
EhB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAA
Q,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAdX,C;mGAiBA,yB;MAAA,4C;MAAA,qD;MAuoEI,8D;MAvoEJ,uC;Q
ASI,iBA8nEgB,cAAR,iBAAQ,CA9nEhB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,C
AAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,W;O;KAdX,C;2FAiBA,yB;MAAA,+D;MA
AA,uC;QAWiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,QA
AJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL
,WAAW,I;;;QAEnB,OAAO,I;O;KAlBX,C;2FAqBA,yB;MAAA,+D;MAAA,uC;QAWiB,Q;QAFb,eAAe,K;QACf,
WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,I
AAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAEnB,OAAO,I;O;KAlBX,
C;2FAqBA,yB;MAAA,+D;MAAA,uC;QAWiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACE,2B;QAAb,OAAa,cAA
b,C;UAAa,sB;UACT,IAAI,QAAJ,C;YACI,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YAC
D,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;;QAEnB,OAAO,I;O;KAlBX,C;2FAqBA,yB;MAAA,+D;MAAA,uC;QA
WiB,Q;QAFb,eAAe,K;QACf,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,QAAJ,C;YACI
,IAAK,WAAI,IAAJ,C;eACJ,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACD,IAAK,WAAI,IAAJ,C;YACL,WAAW,I;;
;QAEnB,OAAO,I;O;KAlBX,C;qFAqBA,yB;MAAA,+D;MAAA,uC;QASW,kBAAS,gB;QAgRA,Q;QAAA,2B;QA
AhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAhRa,SAgRT,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAI,OAA
J,C;;QAhR1D,OAiRO,W;O;KA1RX,C;qFAYA,yB;MAAA,+D;MAAA,uC;QASW,kBAAS,gB;QAiRA,Q;QAAA,2
B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAjRc,SAiRV,CAAU,OAAV,CAAJ,C;YAAwB,WAAY,WAAI,
OAAJ,C;;QAjR1D,OAkRO,W;O;KA3RX,C;qFAYA,yB;MAAA,+D;MAAA,uC;QASW,kBAAS,gB;QAkRA,Q;Q
AAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAlRc,SAkRV,CAAU,OAAV,CAAJ,C;YAAwB,WAAY
,WAAI,OAAJ,C;;QAlR1D,OAmRO,W;O;KA5RX,C;qFAYA,yB;MAAA,+D;MAAA,uC;QASW,kBAAS,gB;QAm
RA,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAnRe,SAmRX,CAAU,OAAV,CAAJ,C;YAAw

B,WAAY,WAAI,OAAJ,C;;QAnR1D,OAoRO,W;O;KA7RX,C;kGAYA,yB;MAAA,+D;MAAA,uC;QAWW,kBAA
gB,gB;QAm6HV,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA12HT,IAzDsC,SAyDlC,EA
02HkB,cA12HlB,EA02HkB,sBA12HlB,WA02H2B,IA12H3B,CAAJ,C;YAA2C,sBA02HZ,IA12HY,C;;QAzD/C,O
A2DO,W;O;KAtEX,C;mGAcA,yB;MAAA,+D;MAAA,uC;QAWW,kBAAgB,gB;QAk6HV,gB;QADb,YAAY,C;Q
ACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UAt2HT,IA5DuC,SA4DnC,EAs2HkB,cAt2HlB,EAs2HkB,sBAt2HlB,W
As2H2B,IAt2H3B,CAAJ,C;YAA2C,sBAs2HZ,IAt2HY,C;;QA5D/C,OA8DO,W;O;KAzEX,C;mGAcA,yB;MAAA,
+D;MAAA,uC;QAWW,kBAAgB,gB;QAi6HV,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;U
Al2HT,IA/DuC,SA+DnC,EAk2HkB,cAl2HlB,EAk2HkB,sBAl2HlB,WAk2H2B,IAl2H3B,CAAJ,C;YAA2C,sBAk2
HZ,IAl2HY,C;;QA/D/C,OAiEO,W;O;KA5EX,C;mGAcA,yB;MAAA,+D;MAAA,uC;QAWW,kBAAgB,gB;QAg6
HV,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UA91HT,IAlEwC,SAkEpC,EA81HkB,cA91
HlB,EA81HkB,sBA91HlB,WA81H2B,IA91H3B,CAAJ,C;YAA2C,sBA81HZ,IA91HY,C;;QAlE/C,OAoEO,W;O;K
A/EX,C;uGAcA,6C;MAs3HiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QA12HT,IAAI,
WA02HkB,cA12HlB,EA02HkB,sBA12HlB,WA02H2B,IA12H3B,CAAJ,C;UAA2C,sBA02HZ,IA12HY,C;;MAE/
C,OAAO,W;K;uGAGX,6C;MAk3HiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAt2HT,
IAAI,WAs2HkB,cAt2HlB,EAs2HkB,sBAt2HlB,WAs2H2B,IAt2H3B,CAAJ,C;UAA2C,sBAs2HZ,IAt2HY,C;;MA
E/C,OAAO,W;K;uGAGX,6C;MA82HiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAl2H
T,IAAI,WAk2HkB,cAl2HlB,EAk2HkB,sBAl2HlB,WAk2H2B,IAl2H3B,CAAJ,C;UAA2C,sBAk2HZ,IAl2HY,C;;
MAE/C,OAAO,W;K;uGAGX,6C;MA02HiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;Q
A91HT,IAAI,WA81HkB,cA91HlB,EA81HkB,sBA91HlB,WA81H2B,IA91H3B,CAAJ,C;UAA2C,sBA81HZ,IA91
HY,C;;MAE/C,OAAO,W;K;2FAGX,yB;MAAA,+D;MAAA,uC;QASW,kBAAY,gB;QAgDH,Q;QAAA,2B;QAAh
B,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CAhDY,SAgDX,CAAU,OAAV,CAAL,C;YAAyB,WAAY,WAAI,
OAAJ,C;;QAhD3D,OAiDO,W;O;KA1DX,C;2FAYA,yB;MAAA,+D;MAAA,uC;QASW,kBAAY,gB;QAiDH,Q;Q
AAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CAjDa,SAiDZ,CAAU,OAAV,CAAL,C;YAAyB,
WAAY,WAAI,OAAJ,C;;QAjD3D,OAkDO,W;O;KA3DX,C;2FAYA,yB;MAAA,+D;MAAA,uC;QASW,kBAAY,g
B;QAkDH,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CAlDa,SAkDZ,CAAU,OAAV,CA
AL,C;YAAyB,WAAY,WAAI,OAAJ,C;;QAlD3D,OAmDO,W;O;KA5DX,C;2FAYA,yB;MAAA,+D;MAAA,uC;Q
ASW,kBAAY,gB;QAmDH,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CAnDc,SAmDb,C
AAU,OAAV,CAAL,C;YAAyB,WAAY,WAAI,OAAJ,C;;QAnD3D,OAoDO,W;O;KA7DX,C;+FAYA,6C;MASoB,
Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WA
AY,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;+FAGX,6C;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAg
B,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;+FA
GX,6C;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL
,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAC3D,OAAO,W;K;+FAGX,6C;MASoB,Q;MAAA,2B;MAAhB,OAAgB,c
AAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,WAAY,WAAI,OAAJ,C;;MAC3D,O
AAO,W;K;yFAGX,6C;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAA
V,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;yFAGX,6C;MASoB,Q;MAAA,2B;MAAhB,O
AAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,O
AAO,W;K;yFAGX,6C;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAA
V,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,OAAO,W;K;yFAGX,6C;MASoB,Q;MAAA,2B;MAAhB,O
AAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,WAAY,WAAI,OAAJ,C;;MAC1D,O
AAO,W;K;IAGX,sC;MAMI,IAAI,OAAQ,UAAZ,C;QAAuB,OlC3jEe,W;;MkC4jEtC,OAA4D,SA0iDrD,cAAkB,c
AAR,iBAAQ,EA1iDN,OAAQ,MA0iDF,EA1iDS,OAAQ,aAAR,GAAAuB,CAAvB,IA0iDT,CAAlB,CA1iDqD,C;K;I
AGhE,sC;MAMI,IAAI,OAAQ,UAAZ,C;QAAuB,OlCrkEe,W;;MkCskEtC,OAA4D,SAgjDrD,eAAmB,cAAR,iBA
AQ,EAhjDP,OAAQ,MAgjDD,EAhjDQ,OAAQ,aAAR,GAAAuB,CAAvB,IAgjDR,CAAnB,CAhjDqD,C;K;IAGhE,s
C;MAMI,IAAI,OAAQ,UAAZ,C;QAAuB,OlC/kEe,W;;MkCglEtC,OAA4D,UAsjDrD,eAAmB,cAAR,iBAAQ,EAtj
DP,OAAQ,MAsjDD,EAtjDQ,OAAQ,aAAR,GAAAuB,CAAvB,IAsjDR,CAAnB,CAtjDqD,C;K;IAGhE,sC;MAMI,I
AAI,OAAQ,UAAZ,C;QAAuB,OlCzlEe,W;;MkC0lEtC,OAA4D,UA4jDrD,gBAAoB,cAAR,iBAAQ,EA5jDR,OAA
Q,MA4jDA,EA5jDO,OAAQ,aAAR,GAAAuB,CAAvB,IA4jDP,CAApB,CA5jDqD,C;K;IAGhE,sC;MASkB,Q;MAH

d,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACtB,WAA
W,iBAAgB,IAAhB,C;MACG,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAK,WAAI,sBAAI,KAAJ,CAAJ,C;;
MAET,OAAO,I;K;IAGX,sC;MASkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAA
Q,CAAZ,C;QAAe,OAAO,W;MACtB,WAAW,iBAAiB,IAAjB,C;MACG,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QA
CV,IAAK,WAAI,sBAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,sC;MASkB,Q;MAHd,WAAmB,wBAAR,OA
AQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,W;MACtB,WAAW,iBAAiB,IAAjB,C;MA
CG,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAK,WAAI,sBAAI,KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,s
C;MASkB,Q;MAHd,WAAmB,wBAAR,OAAQ,EAAwB,EAAxB,C;MACnB,IAAI,SAAQ,CAAZ,C;QAAe,OAAO,
W;MACtB,WAAW,iBAAkB,IAAlB,C;MACG,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,IAAK,WAAI,sBAAI,
KAAJ,CAAJ,C;;MAET,OAAO,I;K;IAGX,2C;MAMI,OAAO,cAAkB,aAAR,iBAAQ,EAAAW,OAAX,CAAlB,C;K;I
AGX,2C;MAMI,OAAO,eAAmB,aAAR,iBAAQ,EAAAW,OAAX,CAAnB,C;K;IAGX,2C;MAMI,OAAO,eAAmB,a
AAR,iBAAQ,EAAAW,OAAX,CAAnB,C;K;IAGX,2C;MAMI,OAAO,gBAAoB,aAAR,iBAAQ,EAAAW,OAAX,CAA
pB,C;K;IAGX,2C;MAMI,OAAO,cAAkB,cAAR,iBAAQ,EAAAW,OAAX,CAAlB,C;K;IAGX,2C;MAMI,OAAO,eA
AmB,cAAR,iBAAQ,EAAAW,OAAX,CAAnB,C;K;IAGX,2C;MAMI,OAAO,eAAmB,aAAR,iBAAQ,EAAAW,OAAX
,CAAnB,C;K;IAGX,2C;MAMI,OAAO,gBAAoB,cAAR,iBAAQ,EAAAW,OAAX,CAApB,C;K;IAGX,+B;MAgBiB,
Q;MxBjyEb,IAAI,EwB2xEI,KAAK,CxB3xET,CAAJ,C;QACI,cwB0xEc,sD;QxBzxEd,MAAM,gCAAyB,OAAQ,
WAAjC,C;;MwB0xEV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;QAAe,OAAO,m
B;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WA
AW,iBAAgB,CAAhB,C;MACE,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,m
CAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,+B;MAgBiB,Q;MxBzzEb,IAAI,EwBmzEI,KAAK,CxBnzET,
CAAJ,C;QACI,cwBkzEc,sD;QxBjzEd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBkzEV,IAAI,MAAK,CAAT,C;Q
AAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,
OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAiB,CAAjB,C;MACE,2B;MAAb,OAAa,c
AAb,C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX
,+B;MAgBiB,Q;MxBj1Eb,IAAI,EwB20EI,KAAK,CxB30ET,CAAJ,C;QACI,cwB00Ec,sD;QxBz0Ed,MAAM,gCA
AyB,OAAQ,WAAjC,C;;MwB00EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;QA
Ae,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAAY,C
;MACZ,WAAW,iBAAiB,CAAjB,C;MACE,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QA
CL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OAAO,I;K;IAGX,+B;MAgBiB,Q;MxBz2Eb,IAAI,EwBm2EI,KAAK
,CxBn2ET,CAAJ,C;QACI,cwBk2Ec,sD;QxBj2Ed,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBk2EV,IAAI,MAAK,
CAAT,C;QAAY,OAAO,W;MACnB,IAAI,KAAK,cAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QA
AY,OAAO,OAAO,sBAAK,CAAL,CAAP,C;MACnB,YAAY,C;MACZ,WAAW,iBAAkB,CAAlB,C;MACE,2B;M
AAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAK,WAAI,IAAJ,C;QACL,IAAI,mCAAW,CAAf,C;UACI,K;;MAER,OA
AO,I;K;IAGX,mC;MxBj3EI,IAAI,EwB23EI,KAAK,CxB33ET,CAAJ,C;QACI,cwB03Ec,sD;QxBz3Ed,MAAM,gC
AAyB,OAAQ,WAAjC,C;;MwB03EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,c;MACX,IAAI,
KAAK,IAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CAAP,I
AAL,CAAP,C;MACnB,WAAW,iBAAgB,CAAhB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QAC
I,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,mC;MxBt4EI,IAAI,EwBg5EI,KAAK,CxBh5ET
,CAAJ,C;QACI,cwB+4Ec,sD;QxB94Ed,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwB+4EV,IAAI,MAAK,CAAT,C;
QAAY,OAAO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAA
T,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAiB,CAAjB,C;MACX,iB
AAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAG
X,mC;MxB35EI,IAAI,EwBq6EI,KAAK,CxBr6ET,CAAJ,C;QACI,cwBo6Ec,sD;QxBn6Ed,MAAM,gCAAyB,OAA
Q,WAAjC,C;;MwBo6EV,IAAI,MAAK,CAAT,C;QAAY,OAAO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,IAA
T,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAAY,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,CAA
P,C;MACnB,WAAW,iBAAiB,CAAjB,C;MACX,iBAAc,OAAO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WA
AI,sBAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;IAGX,mC;MxBh7EI,IAAI,EwB07EI,KAAK,CxB17ET,CAAJ,C;Q
ACI,cwBy7Ec,sD;QxBx7Ed,MAAM,gCAAyB,OAAQ,WAAjC,C;;MwBy7EV,IAAI,MAAK,CAAT,C;QAAY,OA

AO,W;MACnB,WAAW,c;MACX,IAAI,KAAK,IAAT,C;QAAe,OAAO,mB;MACtB,IAAI,MAAK,CAAT,C;QAA
Y,OAAO,OAAO,sBAAK,OAAO,CAAP,IAAL,CAAP,C;MACnB,WAAW,iBAAkB,CAAlB,C;MACX,iBAAc,OA
AO,CAAP,IAAd,UAA6B,IAA7B,U;QACI,IAAK,WAAI,sBAAK,KAAL,CAAJ,C;MACT,OAAO,I;K;mGAGX,yB;
MAAA,4C;MAAA,gD;MAs2CI,8D;MAt2CJ,uC;QASI,iBA61CgB,cAAR,iBAAQ,CA71ChB,WAA+B,CAA/B,U;
UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAG
f,OAAO,iB;O;KAdX,C;mGAiBA,yB;MAAA,4C;MAAA,gD;MA61CI,8D;MA71CJ,uC;QASI,iBAo1CgB,cAAR,iB
AAQ,CAp1ChB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gB
AAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAdX,C;mGAiBA,yB;MAAA,4C;MAAA,gD;MAo1CI,8D;M
Ap1CJ,uC;QASI,iBA20CgB,cAAR,iBAAQ,CA30ChB,WAA+B,CAA/B,U;UACI,IAAI,CAAC,UAAU,sBAAK,KA
AL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,OAAO,iB;O;KAdX,C;mGAiBA,yB;M
AAA,4C;MAAA,gD;MA20CI,8D;MA30CJ,uC;QASI,iBAk0CgB,cAAR,iBAAQ,CAl0ChB,WAA+B,CAA/B,U;UA
CI,IAAI,CAAC,UAAU,sBAAK,KAAL,CAAV,CAAL,C;YACI,OAAO,gBAAK,QAAQ,CAAR,IAAL,C;;;QAGf,O
AAO,iB;O;KAdX,C;2FAiBA,yB;MAAA,+D;MAAA,uC;QAUiB,Q;QADb,WAAW,gB;QACE,2B;QAAb,OAAa,c
AAb,C;UAAa,sB;UACT,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAA
O,I;O;KAfX,C;2FAkBA,yB;MAAA,+D;MAAA,uC;QAUiB,Q;QADb,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,
C;UAAa,sB;UACT,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;
KAfX,C;2FAkBA,yB;MAAA,+D;MAAA,uC;QAUiB,Q;QADb,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UA
Aa,sB;UACT,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KAf
X,C;2FAkBA,yB;MAAA,+D;MAAA,uC;QAUiB,Q;QADb,WAAW,gB;QACE,2B;QAAb,OAAa,cAAb,C;UAAa,s
B;UACT,IAAI,CAAC,UAAU,IAAV,CAAL,C;YACI,K;UACJ,IAAK,WAAI,IAAJ,C;;QAET,OAAO,I;O;KAfX,C;u
FAkBA,yB;MAAA,kD;MAAA,4B;QAOY,QAAR,iBAAQ,C;O;KAPZ,C;uFAUA,yB;MAAA,kD;MAAA,4B;QAO
Y,QAAR,iBAAQ,C;O;KAPZ,C;uFAUA,yB;MAAA,kD;MAAA,4B;QAOY,QAAR,iBAAQ,C;O;KAPZ,C;uFAUA,
yB;MAAA,kD;MAAA,4B;QAOY,QAAR,iBAAQ,C;O;KAPZ,C;uFAUA,yB;MAAA,kD;MAAA,gD;QAaY,QAAR
,iBAAQ,EAAQ,SAAR,EAAmB,OAAnB,C;O;KAbZ,C;uFAgBA,yB;MAAA,kD;MAAA,gD;QAaY,QAAR,iBAAQ
,EAAQ,SAAR,EAAmB,OAAnB,C;O;KAbZ,C;uFAgBA,yB;MAAA,kD;MAAA,gD;QAaY,QAAR,iBAAQ,EAAQ,
SAAR,EAAmB,OAAnB,C;O;KAbZ,C;sFAgBA,yB;MAAA,kD;MAAA,gD;QAaY,QAAR,iBAAQ,EAAQ,SAAR,E
AAmB,OAAnB,C;O;KAbZ,C;IAgBA,gC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,W;MACtB,WAAW,0B;MACN,
WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,gC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,W;MACtB,WAAW,0B;MA
CN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,gC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,W;MACtB,WAAW,0B;
MACN,WAAL,IAAK,C;MACL,OAAO,I;K;IAGX,gC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,W;MACtB,WAAW,
0B;MACN,WAAL,IAAK,C;MACL,OAAO,I;K;kGAGX,yB;MAAA,8D;MAAA,uC;MAAA,4B;QAOI,OAAO,mB
AAkB,cAAR,iBAAQ,CAAlB,C;O;KAPX,C;kGAUA,yB;MAAA,8D;MAAA,yC;MAAA,4B;QAOI,OAAO,oBAA
mB,cAAR,iBAAQ,CAAnB,C;O;KAPX,C;mGAUA,yB;MAAA,8D;MAAA,yC;MAAA,4B;QAOI,OAAO,oBAAm
B,cAAR,iBAAQ,CAAnB,C;O;KAPX,C;mGAUA,yB;MAAA,8D;MAAA,2C;MAAA,4B;QAOI,OAAO,qBAAoB,c
AAR,iBAAQ,CAApB,C;O;KAPX,C;IAUA,+B;MAMI,sBAAQ,4BAAR,C;K;IAGJ,+B;MAMI,sBAAQ,4BAAR,C;
K;IAGJ,+B;MAMI,sBAAQ,4BAAR,C;K;IAGJ,+B;MAMI,sBAAQ,4BAAR,C;K;IAGJ,uC;MAQI,aA8+BgB,gBAA
R,iBAAQ,CA9+BhB,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,sBA
AK,CAAL,C;QACX,sBAAK,CAAL,EAAU,sBAAK,CAAL,CAAV,C;QACA,sBAAK,CAAL,EAAU,IAAV,C;;K;I
AIR,uC;MAQI,aAs+BgB,gBAAR,iBAAQ,CAt+BhB,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CA
AJ,IAAR,C;QACf,WAAW,sBAAK,CAAL,C;QACX,sBAAK,CAAL,EAAU,sBAAK,CAAL,CAAV,C;QACA,sBA
AK,CAAL,EAAU,IAAV,C;;K;IAIR,uC;MAQI,aA89BgB,gBAAR,iBAAQ,CA99BhB,OAA2B,CAA3B,M;QACI,Q
AAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,sBAAK,CAAL,C;QACX,sBAAK,CAAL,EAAU,sBAA
K,CAAL,CAAV,C;QACA,sBAAK,CAAL,EAAU,IAAV,C;;K;IAIR,uC;MAQI,aAs9BgB,gBAAR,iBAAQ,CAt9Bh
B,OAA2B,CAA3B,M;QACI,QAAQ,MAAO,iBAAQ,IAAI,CAAJ,IAAR,C;QACf,WAAW,sBAAK,CAAL,C;QAC
X,sBAAK,CAAL,EAAU,sBAAK,CAAL,CAAV,C;QACA,sBAAK,CAAL,EAAU,IAAV,C;;K;IAIR,sC;MAMI,IA
AI,iBAAO,CAAX,C;QACI,iB;QApSI,UAAR,iBAAQ,C;;K;IAySZ,sC;MAMI,IAAI,iBAAO,CAAX,C;QACI,iB;Q
AtSI,UAAR,iBAAQ,C;;K;IA2SZ,sC;MAMI,IAAI,iBAAO,CAAX,C;QACI,iB;QAxSI,UAAR,iBAAQ,C;;K;IA6SZ,
sC;MAMI,IAAI,iBAAO,CAAX,C;QACI,iB;QA1SI,UAAR,iBAAQ,C;;K;IA+SZ,6B;MAMoB,kBA+nBT,cAAU,iB

vBh9EO,QuBg9EjB,C;MA/nBiB,mB;MAAxB,OAAiC,SrB33F1B,WqB23F0B,C;K;IAGrC,8B;MAMoB,kBAkoBT
,eAAmB,UAAR,iBAAQ,CAAnB,C;MAloBiB,mB;MAAxB,OAAiC,SrBp4F1B,WqBo4F0B,C;K;IAGrC,8B;MAM
oB,kBAqoBT,eAAW,iBvB5/EM,QuB4/EjB,C;MAroBiB,mB;MAAxB,OAAiC,UrB74F1B,WqB64F0B,C;K;IAGrC
,8B;MAMoB,kBAwoBT,gBAAY,iBvB9/EK,QuB8/EjB,C;MAxoBiB,mB;MAAxB,OAAiC,UrBt5F1B,WqBs5F0B,
C;K;IAGrC,kC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBA0lBd,cA1lBA,SA0lBU,QvBh9EO,QuBg9EjB
,C;MA1lBsB,mB;MAA7B,OrBh6FO,W;K;IqBm6FX,kC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBA4lB
d,eAAmB,UA5lBnB,SA4lBW,QAAQ,CAAnB,C;MA5lBsB,mB;MAA7B,OrB16FO,W;K;IqB66FX,kC;MAMI,IA
AI,mBAAJ,C;QAAe,OAAO,S;MACD,kBA8lBd,eA9lBA,SA8lBW,QvB5/EM,QuB4/EjB,C;MA9lBsB,mB;MAA7
B,OrBp7FO,W;K;IqBu7FX,mC;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBAgmBd,gBAhmBA,SAgmBY,
QvB9/EK,QuB8/EjB,C;MAhmBsB,mB;MAA7B,OrB97FO,W;K;IqBi8FX,4C;MAMI,IAAI,mBAAJ,C;QAAe,OAA
O,S;MACD,kBAkjBd,cAljBA,SAkjBU,QvBh9EO,QuBg9EjB,C;MAljBsB,8B;MAA7B,OrBx8FO,W;K;IqB28FX,4
C;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBAojBd,eAAmB,UApjBnB,SAojBW,QAAQ,CAAnB,C;MAp
jBsB,8B;MAA7B,OrBl9FO,W;K;IqBq9FX,4C;MAMI,IAAI,mBAAJ,C;QAAe,OAAO,S;MACD,kBAsjBd,eAtjBA,
SAsjBW,QvB5/EM,QuB4/EjB,C;MAtjBsB,8B;MAA7B,OrB59FO,W;K;IqB+9FX,6C;MAMI,IAAI,mBAAJ,C;QA
Ae,OAAO,S;MACD,kBAwjBd,gBAxjBA,SAwjBY,QvB9/EK,QuB8/EjB,C;MAxjBsB,8B;MAA7B,OrBt+FO,W;K;
IqBy+FX,uC;MAQoB,kBAygBT,cAAU,iBvBh9EO,QuBg9EjB,C;MAzgBiB,mB;MAAxB,OAAiC,YrBj/F1B,WqBi
/F0B,C;K;IAGrC,wC;MAQoB,kBA0gBT,eAAmB,UAAR,iBAAQ,CAAnB,C;MA1gBiB,mB;MAAxB,OAAiC,YrB
5/F1B,WqB4/F0B,C;K;IAGrC,wC;MAQoB,kBA2gBT,eAAW,iBvB5/EM,QuB4/EjB,C;MA3gBiB,mB;MAAxB,O
AAiC,YrBvgG1B,WqBugG0B,C;K;IAGrC,wC;MAQoB,kBA4gBT,gBAAY,iBvB9/EK,QuB8/EjB,C;MA5gBiB,m
B;MAAxB,OAAiC,YrBlhG1B,WqBkhG0B,C;K;4FAGrC,qB;MAQI,OAAO,iB;K;0FAGX,qB;MAQI,OAAO,iB;K;
4FA+BX,qB;MAQI,OAAO,iB;K;8FAGX,qB;MAQI,OAAO,iB;K;8FAGX,yB;MAAA,yC;MAAA,4B;QAQI,OAA
O,oBAAW,SAAX,C;O;KARX,C;4FAWA,yB;MAAA,uC;MAAA,4B;QAQI,OAAO,mBAAU,SAAV,C;O;KARX,C
;8FAWA,yB;MAAA,yC;MAAA,4B;QAQI,OAAO,oBAAW,SAAX,C;O;KARX,C;gGAWA,yB;MAAA,2C;MAAA
,4B;QAQI,OAAO,qBAAY,SAAZ,C;O;KARX,C;IAWA,2C;MASI,OAAY,gBAAL,SAAK,EAAc,KAAd,C;K;IAGh
B,2C;MASI,OAAY,gBAAL,SAAK,EAAc,KAAd,C;K;IAGhB,2C;MASI,OAAY,gBAAL,SAAK,EAAc,KAAd,C;K
;IAGhB,2C;MASI,OAAY,gBAAL,SAAK,EAAc,KAAd,C;K;IAGhB,2C;MAOI,OAAqB,cAAd,4CAAc,EAAc,oCA
Ad,C;K;IAGzB,2C;MAOI,OAAqB,cAAd,4CAAc,EAAc,oCAAd,C;K;IAGzB,2C;MAOI,OAAqB,cAAd,4CAAc,E
AAc,oCAAd,C;K;IAGzB,2C;MAOI,OAAqB,cAAd,4CAAc,EAAc,oCAAd,C;K;IAGzB,sC;MAQI,OAAY,kBAAL,
SAAK,C;K;IAGhB,sC;MAQI,OAAY,kBAAL,SAAK,C;K;IAGhB,sC;MAQI,OAAY,kBAAL,SAAK,C;K;IAGhB,s
C;MAQI,OAAY,kBAAL,SAAK,C;K;IAGhB,sC;MAMI,OAAqB,gBAAd,4CAAc,C;K;IAGzB,sC;MAMI,OAAqB,
gBAAd,4CAAc,C;K;IAGzB,sC;MAMI,OAAqB,gBAAd,4CAAc,C;K;IAGzB,sC;MAMI,OAAqB,gBAAd,4CAAc,
C;K;IAGzB,sC;MAUI,OAAY,kBAAL,SAAK,C;K;IAGhB,sC;MAUI,OAAY,kBAAL,SAAK,C;K;IAGhB,sC;MAU
I,OAAY,kBAAL,SAAK,C;K;IAGhB,sC;MAUI,OAAY,kBAAL,SAAK,C;K;IAGhB,sC;MAQW,Q;MAAP,OAAO,
sDAAmB,IAAnB,EAAyB,GAAzB,EAA8B,GAA9B,2BAAsC,M;K;IAGjD,sC;MAQW,Q;MAAP,OAAO,sDAAmB
,IAAnB,EAAyB,GAAzB,EAA8B,GAA9B,2BAAsC,M;K;IAGjD,sC;MAQW,Q;MAAP,OAAO,sDAAmB,IAAnB,
EAAyB,GAAzB,EAA8B,GAA9B,2BAAsC,M;K;IAGjD,sC;MAQW,Q;MAAP,OAAO,sDAAmB,IAAnB,EAAyB,
GAAzB,EAA8B,GAA9B,2BAAsC,M;K;sFAGjD,yB;MvB5hFA,8C;MuB4hFA,kF;QAmB6D,iC;UAAA,oBAAyB,
C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,c;QvB3hF1H,UuB4hFA,iBvB5hFA,EuB4hFiB,WAA
Y,QvB5hF7B,EuB4hFsC,iBvB5hFtC,EuB4hFyD,UvB5hFzD,EuB4hFqE,QvB5hFrE,C;QuB6hFA,OAAO,W;O;KA
rBX,C;wFAwBA,yB;MvB5hFA,8C;MuB4hFA,kF;QAmB+D,iC;UAAA,oBAAyB,C;QAAG,0B;UAAA,aAAkB,C;
QAAG,wB;UAAA,WAAgB,c;QvB3hF5H,UuB4hFA,iBvB5hFA,EuB4hFiB,WAAY,QvB5hF7B,EuB4hFsC,iBvB5
hFtC,EuB4hFyD,UvB5hFzD,EuB4hFqE,QvB5hFrE,C;QuB6hFA,OAAO,W;O;KArBX,C;wFAwBA,yB;MvB5nFA,
8C;MuB4nFA,kF;QAmB+D,iC;UAAA,oBAAyB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,c;Qv
B3nF5H,UuB4nFA,iBvB5nFA,EuB4nFiB,WAAY,QvB5nF7B,EuB4nFsC,iBvB5nFtC,EuB4nFyD,UvB5nFzD,EuB
4nFqE,QvB5nFrE,C;QuB6nFA,OAAO,W;O;KArBX,C;wFAwBA,yB;MvB5nFA,8C;MuB4nFA,kF;QAmBiE,iC;U
AAA,oBAAyB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,c;QvB3nF9H,UuB4nFA,iBvB5nFA,E
uB4nFiB,WAAY,QvB5nF7B,EuB4nFsC,iBvB5nFtC,EuB4nFyD,UvB5nFzD,EuB4nFqE,QvB5nFrE,C;QuB6nFA,O
AAO,W;O;KArBX,C;kFAwBA,yB;MAAA,uC;MAAA,4B;QASI,OAAO,mBAAU,iBvBh9EO,QuBg9EjB,C;O;KA

TX,C;oFAYA,yB;MAAA,gD;MAAA,yC;MAAA,4B;QASI,OAAO,oBAAmB,OAAR,iBAAQ,CAAnB,C;O;KATX,
C;oFAYA,yB;MAAA,yC;MAAA,4B;QASI,OAAO,oBAAW,iBvB5/EM,QuB4/EjB,C;O;KATX,C;oFAYA,yB;MA
AA,2C;MAAA,4B;QASI,OAAO,qBAAY,iBvB9/EK,QuB8/EjB,C;O;KATX,C;oFAYA,yB;MAAA,gD;MAAA,uC;
MAAA,qC;QAWI,OAAO,mBAAkB,OAAR,iBAAQ,EAAO,OAAP,CAAlB,C;O;KAXX,C;oFAcA,yB;MAAA,gD;
MAAA,yC;MAAA,qC;QAWI,OAAO,oBAAmB,OAAR,iBAAQ,EAAO,OAAP,CAAnB,C;O;KAXX,C;oFAcA,yB;
MAAA,+C;MAAA,yC;MAAA,qC;QAWI,OAAO,oBAAmB,OAAR,iBAAQ,EAAO,OAAP,CAAnB,C;O;KAXX,C;
oFAcA,yB;MAAA,gD;MAAA,2C;MAAA,qC;QAWI,OAAO,qBAAoB,OAAR,iBAAQ,EAAO,OAAP,CAApB,C;O
;KAXX,C;4FAcA,yB;MAAA,0D;MAAA,uC;MAAA,gD;QAaI,OAAO,mBAAkB,YAAR,iBAAQ,EAAY,SAAZ,E
AAuB,OAAvB,CAAlB,C;O;KAbX,C;8FAgBA,yB;MAAA,0D;MAAA,yC;MAAA,gD;QAaI,OAAO,oBAAmB,YA
AR,iBAAQ,EAAY,SAAZ,EAAuB,OAAvB,CAAnB,C;O;KAbX,C;8FAgBA,yB;MAAA,0D;MAAA,yC;MAAA,gD
;QAaI,OAAO,oBAAmB,YAAR,iBAAQ,EAAY,SAAZ,EAAuB,OAAvB,CAAnB,C;O;KAbX,C;6FAgBA,yB;MAA
A,0D;MAAA,2C;MAAA,gD;QAaI,OAAO,qBAAoB,YAAR,iBAAQ,EAAY,SAAZ,EAAuB,OAAvB,CAApB,C;O;
KAbX,C;IAgBA,sD;MAWyC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAChE,OAAR,iBAAQ,EAAK,O
pCj8GoB,KoCi8GzB,EAAsB,SAAtB,EAAiC,OAAjC,C;K;IAGZ,wD;MAW2C,yB;QAAA,YAAiB,C;MAAG,uB;Q
AAA,UAAe,c;MAClE,OAAR,iBAAQ,EAAK,OnBr8GsB,KmBq8G3B,EAAuB,SAAvB,EAAkC,OAAlC,C;K;IAG
Z,wD;MAW2C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAClE,OAAR,iBAAQ,EAAK,OrCvgHsB,KqC
ugH3B,EAAuB,SAAvB,EAAkC,OAAlC,C;K;IAGZ,wD;MAW6C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,
c;MACpE,OAAR,iBAAQ,EAAK,OnC3gHwB,KmC2gH7B,EAAwB,SAAxB,EAAmC,OAAnC,C;K;8FASR,yB;M
AAA,0D;MAAA,4B;QAAQ,OAAQ,YAAR,iBAAQ,C;O;KAAhB,C;8FAQA,yB;MAAA,0D;MAAA,4B;QAAQ,O
AAQ,YAAR,iBAAQ,C;O;KAAhB,C;+FAQA,yB;MAAA,0D;MAAA,4B;QAAQ,OAAQ,YAAR,iBAAQ,C;O;KAA
hB,C;+FAQA,yB;MAAA,0D;MAAA,4B;QAAQ,OAAQ,YAAR,iBAAQ,C;O;KAAhB,C;kGAQA,yB;MAAA,8D;
MAAA,4B;QAAQ,OAAQ,cAAR,iBAAQ,C;O;KAAhB,C;kGAQA,yB;MAAA,8D;MAAA,4B;QAAQ,OAAQ,cAA
R,iBAAQ,C;O;KAAhB,C;mGAQA,yB;MAAA,8D;MAAA,4B;QAAQ,OAAQ,cAAR,iBAAQ,C;O;KAAhB,C;mG
AQA,yB;MAAA,8D;MAAA,4B;QAAQ,OAAQ,cAAR,iBAAQ,C;O;KAAhB,C;iFAEJ,yB;MAAA,uC;MvB3oEA,i
D;MuB2oEA,qC;QAOqB,4B;QAAA,gBAAU,OpCxjHM,K;QoCwjHjC,OAAO,mBvB7oEA,2BAxIK,gBAAW,SA
AX,EAwIL,CuB6oEA,C;O;KAPX,C;iFAUA,yB;MAAA,yC;MvB7oEA,iD;MuB6oEA,qC;QAOI,OAAO,oBvB/oE
A,qBuB+oEW,iBvB/oEX,EAxIK,mBuBuxEgB,OnBvjHO,KJgyCvB,CAwIL,CuB+oEA,C;O;KAPX,C;iFAUA,yB;
MAAA,yC;MvB/qEA,iD;MuB+qEA,qC;QAOsB,4B;QAAA,gBAAU,OrCpnHO,K;QqConHnC,OAAO,oBvBjrEA,
2BAxIK,eAAY,SAAZ,EAwIL,CuBirEA,C;O;KAPX,C;iFAUA,yB;MAAA,2C;MvBjrEA,iD;MuBirEA,qC;QAOuB
,4B;QAAA,gBAAU,OnCnnHQ,K;QmCmnHrC,OAAO,qBvBnrEA,2BAxIK,gBAAa,SAAb,EAwIL,CuBmrEA,C;O;
KAPX,C;IAUA,sC;MAQoB,UAAiB,M;MAFjC,YAAY,c;MACZ,aAAqB,UAAR,iBAAQ,EAAO,iBAAO,QAAS,K
AAhB,IAAP,C;MACL,0B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAU,OAAO,cAAP,EAAO,sBAAP,YAAkB,
OpCjmHX,K;;MoCkmHjC,OAAO,cAAU,MAAV,C;K;IAGX,sC;MAQoB,UAAiB,M;MAFjC,YAAY,c;MACZ,aA
AqB,UAAR,iBAAQ,EAAO,iBAAO,QAAS,KAAhB,IAAP,C;MACL,0B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;
QAAU,OAAO,cAAP,EAAO,sBAAP,YAAkB,OnBlmHT,K;;MmBmmHnC,OAAO,eAAW,MAAX,C;K;IAGX,sC;
MAQoB,UAAiB,M;MAFjC,YAAY,c;MACZ,aAAqB,UAAR,iBAAQ,EAAO,iBAAO,QAAS,KAAhB,IAAP,C;MA
CL,0B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAU,OAAO,cAAP,EAAO,sBAAP,YAAkB,OrCjqHT,K;;MqCk
qHnC,OAAO,eAAW,MAAX,C;K;IAGX,sC;MAQoB,UAAiB,M;MAFjC,YAAY,c;MACZ,aAAqB,UAAR,iBAAQ,
EAAO,iBAAO,QAAS,KAAhB,IAAP,C;MACL,0B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAU,OAAO,cAAP,
EAAO,sBAAP,YAAkB,OnClqHP,K;;MmCmqHrC,OAAO,gBAAY,MAAZ,C;K;iFAGX,yB;MAAA,uC;MvBnuEA
,iD;MuBmuEA,sC;QAOI,OAAO,mBvBruEA,qBuBquEU,iBvBruEV,EuBquEoB,QAAS,QvBruE7B,CuBquEA,C;O
;KAPX,C;iFAUA,yB;MAAA,yC;MvBruEA,iD;MuBquEA,sC;QAOI,OAAO,oBvBvuEA,qBuBuuEW,iBvBvuEX,E
uBuuEqB,QAAS,QvBvuE9B,CuBuuEA,C;O;KAPX,C;iFAUA,yB;MAAA,yC;MvBvwEA,iD;MuBuwEA,sC;QAOI
,OAAO,oBvBzwEA,qBuBywEW,iBvBzwEX,EuBywEqB,QAAS,QvBzwE9B,CuBywEA,C;O;KAPX,C;iFAUA,yB
;MAAA,2C;MvBzwEA,iD;MuBywEA,sC;QAOI,OAAO,qBvB3wEA,qBuB2wEY,iBvB3wEZ,EuB2wEsB,QAAS,
QvB3wE/B,CuB2wEA,C;O;KAPX,C;IAUA,2B;MAQI,IAAI,iBAAO,CAAX,C;QAAc,YAAU,SAAV,EAAgB,CA
AhB,EAAmB,cAAnB,C;K;IAGlB,2B;MAQI,IAAI,iBAAO,CAAX,C;QAAc,YAAU,SAAV,EAAgB,CAAhB,EAA
mB,cAAnB,C;K;IAGlB,2B;MAQI,IAAI,iBAAO,CAAX,C;QAAc,YAAU,SAAV,EAAgB,CAAhB,EAAmB,cAAn

B,C;K;IAGlB,2B;MAQI,IAAI,iBAAO,CAAX,C;QAAc,YAAU,SAAV,EAAgB,CAAhB,EAAmB,cAAnB,C;K;IAG
lB,+C;MAa0B,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACzD,oCAAa,2BAAkB,SAAlB,EAA6B,OAA
7B,EAAsC,cAAtC,C;MACb,YAAU,SAAV,EAAgB,SAAhB,EAA2B,OAA3B,C;K;IAGJ,+C;MAa2B,yB;QAAA,Y
AAiB,C;MAAG,uB;QAAA,UAAe,c;MAC1D,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,cAAtC,C;MACb
,YAAU,SAAV,EAAgB,SAAhB,EAA2B,OAA3B,C;K;IAGJ,+C;MAa2B,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,
UAAe,c;MAC1D,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,cAAtC,C;MACb,YAAU,SAAV,EAAgB,SA
AhB,EAA2B,OAA3B,C;K;IAGJ,+C;MAa4B,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAC3D,oCAAa,2
BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,cAAtC,C;MACb,YAAU,SAAV,EAAgB,SAAhB,EAA2B,OAA3B,C;K;I
AGJ,0D;MAaI,kBAAK,SAAL,EAAgB,OAAhB,C;MAh8CQ,WAAR,iBAAQ,EAi8CA,SAj8CA,EAi8CW,OAj8CX
,C;K;IAo8CZ,0D;MAaI,kBAAK,SAAL,EAAgB,OAAhB,C;MAj8CQ,WAAR,iBAAQ,EAk8CA,SAl8CA,EAk8CW
,OAl8CX,C;K;IAq8CZ,0D;MAaI,kBAAK,SAAL,EAAgB,OAAhB,C;MAl8CQ,UAAR,iBAAQ,EAm8CA,SAn8CA
,EAm8CW,OAn8CX,C;K;IAs8CZ,0D;MAaI,kBAAK,SAAL,EAAgB,OAAhB,C;MAn8CQ,WAAR,iBAAQ,EAo8C
A,SAp8CA,EAo8CW,OAp8CX,C;K;8FAu8CZ,qB;MAQI,OAAO,iBvB/jGiB,Q;K;4FuBkkG5B,qB;MAQI,OAAO,i
BvBtjGiB,Q;K;8FuByjG5B,yB;MAAA,gD;MAAA,4B;QAQI,OAAe,OAAR,iBAAQ,C;O;KARnB,C;gGAWA,qB;
MAQI,OAAO,iBvBtlGiB,Q;K;IuB+lGL,gD;MAAA,wB;QAAW,qCAAK,KAAL,C;O;K;IANlC,iC;MAMI,OAAO,i
BAAM,cAAN,EAAY,8BAAZ,C;K;IASY,kD;MAAA,wB;QAAW,qCAAK,KAAL,C;O;K;IANlC,mC;MAMI,OAA
O,iBAAM,cAAN,EAAY,gCAAZ,C;K;IASY,kD;MAAA,wB;QAAW,qCAAK,KAAL,C;O;K;IANlC,mC;MAMI,O
AAO,iBAAM,cAAN,EAAY,gCAAZ,C;K;IASY,kD;MAAA,wB;QAAW,qCAAK,KAAL,C;O;K;IANlC,mC;MAMI
,OAAO,iBAAM,cAAN,EAAY,gCAAZ,C;K;IASiB,gD;MAAA,wB;QAAW,yBAAK,KAAL,C;O;K;IANvC,iC;MA
MI,OJxqIO,eAAW,+BIwqIA,gBJxqIA,GAAgB,kBIwqIV,8BJxqIU,CAAhB,CAAX,C;K;gGI2qIX,yB;MAAA,yC;
MAAA,4B;QAQI,OAAO,oBAAW,SvBxpGM,QuBwpGjB,C;O;KARX,C;IAiB2B,8C;MAAA,wB;QAAW,wBAAK
,KAAL,C;O;K;IANtC,gC;MAMI,OH5rIO,cAAU,gCG4rIA,gBH5rIA,GAAe,iBG4rIT,6BH5rIS,CAAf,CAAV,C;K;
8FG+rIX,yB;MAAA,uC;MAAA,4B;QAQI,OAAO,mBAAU,SvBxpGO,QuBwpGjB,C;O;KARX,C;IAiB4B,gD;MA
AA,wB;QAAW,yBAAK,KAAL,C;O;K;IANvC,iC;MAMI,OFhtIO,eAAW,kBEgtIA,gBFhtIA,EAAgB,kBEgtIV,8B
FhtIU,CAAhB,CAAX,C;K;gGEmtIX,yB;MAAA,gD;MAAA,yC;MAAA,4B;QAQI,OAAO,oBAAgB,OAAL,SAA
K,CAAhB,C;O;KARX,C;IAiB6B,kD;MAAA,wB;QAAW,0BAAK,KAAL,C;O;K;IANxC,kC;MAMI,ODpuIO,gBA
AY,gCCouIA,gBDpuIA,GAAiB,mBCouIX,+BDpuIW,CAAjB,CAAZ,C;K;kGCuuIX,yB;MAAA,2C;MAAA,4B;Q
AQI,OAAO,qBAAY,SvB1sGK,QuB0sGjB,C;O;KARX,C;mGAWA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA
,2C;QAcI,aAAa,mBAAyC,cAAlB,YAAY,cAAZ,CAAkB,EAAc,EAAd,CAAzC,C;QAsEG,Q;QAAA,2B;QAAhB,
OAAgB,cAAhB,C;UAAgB,yB;UArEO,MAsEP,aAAI,OAAJ,EAtEe,aAsEF,CAAc,OAAd,CAAb,C;;QAtEhB,OAA
uB,M;O;KAf3B,C;mGAkBA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAcI,aAAa,mBAA0C,cAAlB,YA
AY,cAAZ,CAAkB,EAAc,EAAd,CAA1C,C;QAsEG,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UArEO
,MAsEP,aAAI,OAAJ,EAtEe,aAsEF,CAAc,OAAd,CAAb,C;;QAtEhB,OAAuB,M;O;KAf3B,C;kGAkBA,yB;MAA
A,0D;MAAA,yD;MAAA,uE;MAAA,2C;QAcI,aAAa,mBAA0C,cAAlB,YAAY,cAAZ,CAAkB,EAAc,EAAd,CAA1
C,C;QAsEG,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UArEO,MAsEP,aAAI,OAAJ,EAtEe,aAsEF,CA
Ac,OAAd,CAAb,C;;QAtEhB,OAAuB,M;O;KAf3B,C;mGAkBA,yB;MAAA,0D;MAAA,yD;MAAA,uE;MAAA,2C
;QAcI,aAAa,mBAA2C,cAAlB,YAAY,cAAZ,CAAkB,EAAc,EAAd,CAA3C,C;QAsEG,Q;QAAA,2B;QAAhB,OA
AgB,cAAhB,C;UAAgB,yB;UArEO,MAsEP,aAAI,OAAJ,EAtEe,aAsEF,CAAc,OAAd,CAAb,C;;QAtEhB,OAAuB,
M;O;KAf3B,C;uGAkBA,iD;MAYoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,WAAY,aAAI,
OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;uGAGX,iD;MAYoB,Q;MAAA,2B;MAAhB,OAAgB,c
AAhB,C;QAAgB,yB;QACZ,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;uGAGX,iD;
MAYoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,C
AAb,C;;MAEhB,OAAO,W;K;uGAGX,iD;MAYoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,
WAAY,aAAI,OAAJ,EAAa,cAAc,OAAd,CAAb,C;;MAEhB,OAAO,W;K;uFAGX,yB;MAAA,+D;MAoLA,gD;MA
pLA,uC;QASW,kBAAU,gB;QAkLD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAnL6B,SAm
LlB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QApLhB,OAAsLO,W;O;KA/LX,C;uFAYA,yB;MAA
A,+D;MAsLA,gD;MAtLA,uC;QASW,kBAAU,gB;QAoLD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;
UACZ,WArL6B,SAqLlB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAtLhB,OAwLO,W;O;KAjM

X,C;uFAYA,yB;MAAA,+D;MAwLA,gD;MAxLA,uC;QASW,kBAAU,gB;QAsLD,Q;QAAA,2B;QAAhB,OAAgB,
cAAhB,C;UAAgB,yB;UACZ,WAvL6B,SAuLlB,CAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAxLh
B,OA0LO,W;O;KAnMX,C;uFAYA,yB;MAAA,+D;MA0LA,gD;MA1LA,uC;QASW,kBAAU,gB;QAwLD,Q;QAA
A,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAzL6B,SAyLlB,CAAU,OAAV,C;UACC,OAAZ,WAAY,E
AAO,IAAP,C;;QA1LhB,OA4LO,W;O;KArMX,C;qGAYA,yB;MAAA,+D;MA4DA,gD;MA5DA,uC;QAYW,kBA
AiB,gB;QA2DR,gB;QADhB,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WA5DoC,SA4D
zB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QA7DhB,OA+DO,
W;O;KA3EX,C;qGAeA,yB;MAAA,+D;MA+DA,gD;MA/DA,uC;QAYW,kBAAiB,gB;QA8DR,gB;QADhB,YAA
Y,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WA/DoC,SA+DzB,EAAU,cAAV,EAAU,sBAAV,
WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAhEhB,OAkEO,W;O;KA9EX,C;qGAeA,yB;MAA
A,+D;MAkEA,gD;MAlEA,uC;QAYW,kBAAiB,gB;QAiER,gB;QADhB,YAAY,C;QACI,2B;QAAhB,OAAgB,cA
AhB,C;UAAgB,yB;UACZ,WAlEoC,SAkEzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,
WAAY,EAAO,IAAP,C;;QAnEhB,OAqEO,W;O;KAjFX,C;qGAeA,yB;MAAA,+D;MAqEA,gD;MArEA,uC;QAY
W,kBAAiB,gB;QAoER,gB;QADhB,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WArEoC,
SAqEzB,EAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAtEhB,OA
wEO,W;O;KApFX,C;yGAeA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACI,2B;QAAhB,
OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ
,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAfX,C;yGAkBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;Q
AFzB,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,
WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAfX,C;yGAkBA,yB;MAAA,g
D;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACI,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WA
AW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,
W;O;KAfX,C;yGAkBA,yB;MAAA,gD;MAAA,oD;QAWoB,UACS,M;QAFzB,YAAY,C;QACI,2B;QAAhB,OAAg
B,cAAhB,C;UAAgB,yB;UACZ,WAAW,WAAU,cAAV,EAAU,sBAAV,WAAmB,OAAnB,C;UACC,OAAZ,WAA
Y,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAfX,C;2FAkBA,yB;MAAA,gD;MAAA,oD;QAOoB,Q;QAAA,2B;QAA
hB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAE
hB,OAAO,W;O;KAXX,C;2FAcA,yB;MAAA,gD;MAAA,oD;QAOoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;U
AAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAXX,
C;2FAcA,yB;MAAA,gD;MAAA,oD;QAOoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAA
W,UAAU,OAAV,C;UACC,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAXX,C;2FAcA,yB;MAAA,gD
;MAAA,oD;QAOoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,WAAW,UAAU,OAAV,C;UAC
C,OAAZ,WAAY,EAAO,IAAP,C;;QAEhB,OAAO,W;O;KAXX,C;uFAcA,yB;MAAA,wE;MA4HA,+D;MA5HA,y
C;QAYW,kBAAU,oB;QA4HD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UA7HoD,WA6H1C,
CAAY,OAAZ,C;UjC59IP,U;UADP,YiC89Ie,WjC99IH,WiC89IwB,GjC99IxB,C;UACL,IAAI,aAAJ,C;YACH,aiC4
9IuC,gB;YAA5B,WjC39IX,aiC29IgC,GjC39IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCw9IA,iB;UACA,IAAK,
WAAI,OAAJ,C;;QA/HT,OAiIO,W;O;KA7IX,C;uFAeA,yB;MAAA,wE;MAiIA,+D;MAjIA,yC;QAYW,kBAAU,o
B;QAiID,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAlIqD,WAkI3C,CAAY,OAAZ,C;UjCh/I
P,U;UADP,YiCk/Ie,WjCl/IH,WiCk/IwB,GjCl/IxB,C;UACL,IAAI,aAAJ,C;YACH,aiCg/IuC,gB;YAA5B,WjC/+IX,
aiC++IgC,GjC/+IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiC4+IA,iB;UACA,IAAK,WAAI,OAAJ,C;;QApIT,OAs
IO,W;O;KAlJX,C;sFAeA,yB;MAAA,wE;MAsIA,+D;MAtIA,yC;QAYW,kBAAU,oB;QAsID,Q;QAAA,2B;QAAh
B,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAvIqD,WAuI3C,CAAY,OAAZ,C;UjCpgJP,U;UADP,YiCsgJe,WjCtgJH
,WiCsgJwB,GjCtgJxB,C;UACL,IAAI,aAAJ,C;YACH,aiCogJuC,gB;YAA5B,WjCngJX,aiCmgJgC,GjCngJhC,EAA
S,MAAT,C;YACA,e;;YAEA,c;;UiCggJA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAzIT,OA2IO,W;O;KAvJX,C;uFAe
A,yB;MAAA,wE;MA2IA,+D;MA3IA,yC;QAYW,kBAAU,oB;QA2ID,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;U
AAgB,yB;UACZ,UA5IsD,WA4I5C,CAAY,OAAZ,C;UjCxhJP,U;UADP,YiC0hJe,WjC1hJH,WiC0hJwB,GjC1hJx
B,C;UACL,IAAI,aAAJ,C;YACH,aiCwhJuC,gB;YAA5B,WjCvhJX,aiCuhJgC,GjCvhJhC,EAAS,MAAT,C;YACA,e
;;YAEA,c;;UiCohJA,iB;UACA,IAAK,WAAI,OAAJ,C;;QA9IT,OAgJO,W;O;KA5JX,C;uFAeA,yB;MAAA,wE;MA
gJA,+D;MAhJA,yD;QAaW,kBAAU,oB;QAgJD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAj

JiD,WAiJvC,CAAY,OAAZ,C;UjC7iJP,U;UADP,YiC+iJe,WjC/iJH,WiC+iJwB,GjC/iJxB,C;UACL,IAAI,aAAJ,C;Y
ACH,aiC6iJuC,gB;YAA5B,WjC5iJX,aiC4iJgC,GjC5iJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCyiJA,iB;UACA
,IAAK,WAnJyD,cAmJrD,CAAe,OAAf,CAAJ,C;;QAnJT,OAqJO,W;O;KAlKX,C;uFAgBA,yB;MAAA,wE;MAqJA
,+D;MArJA,yD;QAaW,kBAAU,oB;QAqJD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAtJiD,
WAsJvC,CAAY,OAAZ,C;UjClkJP,U;UADP,YiCokJe,WjCpkJH,WiCokJwB,GjCpkJxB,C;UACL,IAAI,aAAJ,C;Y
ACH,aiCkkJuC,gB;YAA5B,WjCjkJX,aiCikJgC,GjCjkJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiC8jJA,iB;UAC
A,IAAK,WAxJyD,cAwJrD,CAAe,OAAf,CAAJ,C;;QAxJT,OA0JO,W;O;KAvKX,C;uFAgBA,yB;MAAA,wE;MA0
JA,+D;MA1JA,yD;QAaW,kBAAU,oB;QA0JD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UA3J
iD,WA2JvC,CAAY,OAAZ,C;UjCvlJP,U;UADP,YiCylJe,WjCzlJH,WiCylJwB,GjCzlJxB,C;UACL,IAAI,aAAJ,C;Y
ACH,aiCulJuC,gB;YAA5B,WjCtlJX,aiCslJgC,GjCtlJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCmlJA,iB;UACA,
IAAK,WA7JyD,cA6JrD,CAAe,OAAf,CAAJ,C;;QA7JT,OA+JO,W;O;KA5KX,C;uFAgBA,yB;MAAA,wE;MA+JA
,+D;MA/JA,yD;QAaW,kBAAU,oB;QA+JD,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAhKiD
,WAgKvC,CAAY,OAAZ,C;UjC5mJP,U;UADP,YiC8mJe,WjC9mJH,WiC8mJwB,GjC9mJxB,C;UACL,IAAI,aAAJ
,C;YACH,aiC4mJuC,gB;YAA5B,WjC3mJX,aiC2mJgC,GjC3mJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCwmJ
A,iB;UACA,IAAK,WAlKyD,cAkKrD,CAAe,OAAf,CAAJ,C;;QAlKT,OAoKO,W;O;KAjLX,C;2FAgBA,yB;MAA
A,+D;MAAA,sD;QAYoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;U
jC59IP,U;UADP,YiC89Ie,WjC99IH,WiC89IwB,GjC99IxB,C;UACL,IAAI,aAAJ,C;YACH,aiC49IuC,gB;YAA5B,
WjC39IX,aiC29IgC,GjC39IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCw9IA,iB;UACA,IAAK,WAAI,OAAJ,C;;
QAET,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAAA,+D;MAAA,sD;QAYoB,Q;QAAA,2B;QAAhB,OAAgB,cAAh
B,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;UjCh/IP,U;UADP,YiCk/Ie,WjCl/IH,WiCk/IwB,GjCl/IxB,C;UAC
L,IAAI,aAAJ,C;YACH,aiCg/IuC,gB;YAA5B,WjC/+IX,aiC++IgC,GjC/+IhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;
UiC4+IA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAAA,+D;MAAA,sD;Q
AYoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;UjCpgJP,U;UADP,Y
iCsgJe,WjCtgJH,WiCsgJwB,GjCtgJxB,C;UACL,IAAI,aAAJ,C;YACH,aiCogJuC,gB;YAA5B,WjCngJX,aiCmgJgC,
GjCngJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCggJA,iB;UACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;K
AjBX,C;2FAoBA,yB;MAAA,+D;MAAA,sD;QAYoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAC
Z,UAAU,YAAY,OAAZ,C;UjCxhJP,U;UADP,YiC0hJe,WjC1hJH,WiC0hJwB,GjC1hJxB,C;UACL,IAAI,aAAJ,C;
YACH,aiCwhJuC,gB;YAA5B,WjCvhJX,aiCuhJgC,GjCvhJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCohJA,iB;U
ACA,IAAK,WAAI,OAAJ,C;;QAET,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAAAA
,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;UjC7iJP,U;UADP,YiC+iJe,WjC/iJH,
WiC+iJwB,GjC/iJxB,C;UACL,IAAI,aAAJ,C;YACH,aiC6iJuC,gB;YAA5B,WjC5iJX,aiC4iJgC,GjC5iJhC,EAAS,M
AAT,C;YACA,e;;YAEA,c;;UiCyiJA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAlBX,
C;2FAqBA,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAA
U,YAAY,OAAZ,C;UjClkJP,U;UADP,YiCokJe,WjCpkJH,WiCokJwB,GjCpkJxB,C;UACL,IAAI,aAAJ,C;YACH,ai
CkkJuC,gB;YAA5B,WjCjkJX,aiCikJgC,GjCjkJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiC8jJA,iB;UACA,IAAK,
WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAlBX,C;2FAqBA,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAA
A,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UAAU,YAAY,OAAZ,C;UjCvlJP,U;UADP,YiCylJe,WjCzlJ
H,WiCylJwB,GjCzlJxB,C;UACL,IAAI,aAAJ,C;YACH,aiCulJuC,gB;YAA5B,WjCtlJX,aiCslJgC,GjCtlJhC,EAAS,
MAAT,C;YACA,e;;YAEA,c;;UiCmlJA,iB;UACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAlB
X,C;2FAqBA,yB;MAAA,+D;MAAA,sE;QAaoB,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,UA
AU,YAAY,OAAZ,C;UjC5mJP,U;UADP,YiC8mJe,WjC9mJH,WiC8mJwB,GjC9mJxB,C;UACL,IAAI,aAAJ,C;YA
CH,aiC4mJuC,gB;YAA5B,WjC3mJX,aiC2mJgC,GjC3mJhC,EAAS,MAAT,C;YACA,e;;YAEA,c;;UiCwmJA,iB;U
ACA,IAAK,WAAI,eAAe,OAAf,CAAJ,C;;QAET,OAAO,W;O;KAlBX,C;+EAqBA,yB;MAAA,gE;MAAA,uC;QA
UW,kBAAM,eAAa,cAAb,C;QAsKA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAY,WAvKiB,SAu
Kb,CAAU,IAAV,CAAJ,C;;QAvKhB,OAwKO,W;O;KAlLX,C;+EAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM
,eAAa,cAAb,C;QAsKA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAY,WAvKiB,SAuKb,CAAU,I
AAV,CAAJ,C;;QAvKhB,OAwKO,W;O;KAlLX,C;8EAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM,eAAa,cAA
b,C;QAsKA,Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAY,WAvKiB,SAuKb,CAAU,IAAV,CAAJ,

C;;QAvKhB,OAwKO,W;O;KAlLX,C;+EAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAM,eAAa,cAAb,C;QAsKA,
Q;QAAA,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAY,WAvKiB,SAuKb,CAAU,IAAV,CAAJ,C;;QAvKhB,
OAwKO,W;O;KAlLX,C;4FAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAa,eAAa,cAAb,C;QAqDP,gB;QADb,YA
AY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAY,WAtDwB,SAsDpB,EAAU,cAAV,EAAU,sBAA
V,WAAmB,IAAnB,CAAJ,C;;QAtDhB,OAuDO,W;O;KAjEX,C;6FAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAa
,eAAa,cAAb,C;QAwDP,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAY,WAzDw
B,SAyDpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QAzDhB,OA0DO,W;O;KApEX,C;6FAaA,y
B;MAAA,gE;MAAA,uC;QAUW,kBAAa,eAAa,cAAb,C;QA2DP,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cA
Ab,C;UAAa,sB;UACT,WAAY,WA5DwB,SA4DpB,EAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;QA
5DhB,OA6DO,W;O;KAvEX,C;4FAaA,yB;MAAA,gE;MAAA,uC;QAUW,kBAAa,eAAa,cAAb,C;QA8DP,gB;QA
Db,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,WAAY,WA/DwB,SA+DpB,EAAU,cAAV,EAA
U,sBAAV,WAAmB,IAAnB,CAAJ,C;;QA/DhB,OAgEO,W;O;KA1EX,C;iGAaA,6C;MAWiB,UACiB,M;MAF9B,
YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAY,WAAI,WAAU,cAAV,EAAU,sBAAV,WA
AmB,IAAnB,CAAJ,C;;MAChB,OAAO,W;K;iGAGX,6C;MAWiB,UACiB,M;MAF9B,YAAY,C;MACC,2B;MAA
b,OAAa,cAAb,C;QAAa,sB;QACT,WAAY,WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;MA
ChB,OAAO,W;K;iGAGX,6C;MAWiB,UACiB,M;MAF9B,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;
QACT,WAAY,WAAI,WAAU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;MAChB,OAAO,W;K;iGAGX,6
C;MAWiB,UACiB,M;MAF9B,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAY,WAAI,WA
AU,cAAV,EAAU,sBAAV,WAAmB,IAAnB,CAAJ,C;;MAChB,OAAO,W;K;mFAGX,6C;MAQiB,Q;MAAA,2B;M
AAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAY,WAAI,UAAU,IAAV,CAAJ,C;;MAChB,OAAO,W;K;mFAGX,6C;
MAQiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAY,WAAI,UAAU,IAAV,CAAJ,C;;MAChB,O
AAO,W;K;mFAGX,6C;MAQiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAY,WAAI,UAAU,IA
AV,CAAJ,C;;MAChB,OAAO,W;K;mFAGX,6C;MAQiB,Q;MAAA,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,
WAAY,WAAI,UAAU,IAAV,CAAJ,C;;MAChB,OAAO,W;K;IAUiB,6C;MAAA,mB;QAAE,gC;O;K;IAP9B,iC;M
AOI,OAAO,qBAAiB,8BAAjB,C;K;IAUiB,6C;MAAA,mB;QAAE,gC;O;K;IAP9B,iC;MAOI,OAAO,qBAAiB,8BA
AjB,C;K;IAUiB,6C;MAAA,mB;QAAE,gC;O;K;IAP9B,iC;MAOI,OAAO,qBAAiB,8BAAjB,C;K;IAUiB,6C;MAA
A,mB;QAAE,gC;O;K;IAP9B,iC;MAOI,OAAO,qBAAiB,8BAAjB,C;K;+EAGX,gC;MAaoB,Q;MAAA,2B;MAAhB
,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACtD,OAAO,
I;K;+EAGX,gC;MAaoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UAAU,OAA
V,CAAL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;+EAGX,gC;MAaoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C
;QAAgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAyB,OAAO,K;;MACtD,OAAO,I;K;+EAGX,gC;M
AaoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,CAAC,UAAU,OAAV,CAAL,C;UAAy
B,OAAO,K;;MACtD,OAAO,I;K;+EAGX,yB;MAAA,0C;MAAA,4B;QASI,OAAe,IAAR,iBAAQ,C;O;KATnB,C;+
EAYA,yB;MAAA,0C;MAAA,4B;QASI,OAAe,IAAR,iBAAQ,C;O;KATnB,C;+EAYA,yB;MAAA,0C;MAAA,4B;
QASI,OAAe,IAAR,iBAAQ,C;O;KATnB,C;+EAYA,yB;MAAA,0C;MAAA,4B;QASI,OAAe,IAAR,iBAAQ,C;O;K
ATnB,C;+EAYA,gC;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,
CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;+EAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;
QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;+EAGX,gC;MASoB,Q
;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,I;;MA
CrD,OAAO,K;K;+EAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU
,OAAV,CAAJ,C;UAAwB,OAAO,I;;MACrD,OAAO,K;K;mFAGX,gC;MAQoB,Q;MADhB,YAAY,C;MACI,2B;M
AAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;m
FAGX,gC;MAQoB,Q;MADhB,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,
OAAV,CAAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;mFAGX,gC;MAQoB,Q;MADhB,YAAY,C;MACI,2B;MAAhB
,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;mFAGX
,gC;MAQoB,Q;MADhB,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAA
V,CAAJ,C;UAAwB,qB;;MAC9C,OAAO,K;K;iFAGX,yC;MAaoB,Q;MADhB,kBAAkB,O;MACF,2B;MAAhB,OA
AgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;iFAGX,yC;M

AaoB,Q;MADhB,kBAAkB,O;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,UAAU,WAAV,E
AAuB,OAAvB,C;;MACpC,OAAO,W;K;iFAGX,yC;MAaoB,Q;MADhB,kBAAkB,O;MACF,2B;MAAhB,OAAgB,
cAAhB,C;QAAgB,yB;QAAM,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;;MACpC,OAAO,W;K;iFAGX,yC;MAao
B,Q;MADhB,kBAAkB,O;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,UAAU,WAAV,EAAu
B,OAAvB,C;;MACpC,OAAO,W;K;+FAGX,yC;MAeoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MACF
,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAnB,EA
AgC,OAAhC,C;;MACpC,OAAO,W;K;+FAGX,yC;MAeoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;MA
CF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAnB,E
AAgC,OAAhC,C;;MACpC,OAAO,W;K;+FAGX,yC;MAeoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;M
ACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAnB,
EAAgC,OAAhC,C;;MACpC,OAAO,W;K;+FAGX,yC;MAeoB,UAA8B,M;MAF9C,YAAY,C;MACZ,kBAAkB,O;
MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,cAAc,WAAU,cAAV,EAAU,sBAAV,WAAmB,WAAn
B,EAAgC,OAAhC,C;;MACpC,OAAO,W;K;0FAGX,yB;MA1vDI,8D;MA0vDJ,gD;QAeoC,Q;QAHhC,YAtwDgB,
cAAR,iBAAQ,C;QAuwDhB,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,YAAJ,EA
AI,oBAAJ,QAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAjBX,C;2FAoBA,yB;MAtwDI,8D;MAswDJ,gD;Q
AeoC,Q;QAHhC,YAlxDgB,cAAR,iBAAQ,C;QAmxDhB,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cA
Ac,UAAU,uBAAI,YAAJ,EAAI,oBAAJ,QAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAjBX,C;2FAoBA,yB;
MAlxDI,8D;MAkxDJ,gD;QAeoC,Q;QAHhC,YA9xDgB,cAAR,iBAAQ,C;QA+xDhB,kBAAkB,O;QAClB,OAAO,
SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,YAAJ,EAAI,oBAAJ,QAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W
;O;KAjBX,C;2FAoBA,yB;MA9xDI,8D;MA8xDJ,gD;QAeoC,Q;QAHhC,YA1yDgB,cAAR,iBAAQ,C;QA2yDhB,k
BAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,YAAJ,EAAI,oBAAJ,QAAV,EAAwB,W
AAxB,C;;QAElB,OAAO,W;O;KAjBX,C;yGAoBA,yB;MA10DI,8D;MA00DJ,gD;QAaI,YAv1DgB,cAAR,iBAAQ,
C;QAw1DhB,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,CA
AjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAnBX,C;yGAsBA,yB;MAx1DI,8D;MAw1DJ,gD;QAaI,
YAr2DgB,cAAR,iBAAQ,C;QAs2DhB,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,
EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAnBX,C;yGAsBA,yB;MAt2DI
,8D;MAs2DJ,gD;QAaI,YAn3DgB,cAAR,iBAAQ,C;QAo3DhB,kBAAkB,O;QAClB,OAAO,SAAS,CAAhB,C;UAC
I,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAnBX,
C;yGAsBA,yB;MAp3DI,8D;MAo3DJ,gD;QAaI,YAj4DgB,cAAR,iBAAQ,C;QAk4DhB,kBAAkB,O;QAClB,OAA
O,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QA
EJ,OAAO,W;O;KAnBX,C;uFAsBA,6B;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,O
AAO,OAAP,C;;K;uFAG1B,6B;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,O
AAP,C;;K;uFAG1B,6B;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;
K;uFAG1B,6B;MAOoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,C;;K;qGAG
1B,6B;MAUiB,UAAa,M;MAD1B,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,E
AAO,sBAAP,WAAgB,IAAhB,C;;K;qGAGvB,6B;MAUiB,UAAa,M;MAD1B,YAAY,C;MACC,2B;MAAb,OAAa,
cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;K;qGAGvB,6B;MAUiB,UAAa,M;M
AD1B,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAA
hB,C;;K;qGAGvB,6B;MAUiB,UAAa,M;MAD1B,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,
QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;K;IAGvB,2B;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,
6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OA1gEG,gBAAR,iBAAQ,C;MA0gEhB,aAAU,CAAV,iB;QACI,QA
AQ,sBAAK,CAAL,C;QACR,IpC5xL8D,YoC4xL1D,GpC5xL2E,KAAjB,EoC4xLpD,CpC5xLiF,KAA7B,CoC4xL
1D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2B;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6
B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OArhEG,gBAAR,iBAAQ,C;MAqhEhB,aAAU,CAAV,iB;QACI,QAA
Q,sBAAK,CAAL,C;QACR,InBvyL+D,amBuyL3D,GnBvyL6E,KAAlB,EmBuyLrD,CnBvyLmF,KAA9B,CmBuyL
3D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2B;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6
B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OAhiEG,gBAAR,iBAAQ,C;MAgiEhB,aAAU,CAAV,iB;QACI,QAA
Q,sBAAK,CAAL,C;QACR,IrCl1L4E,0BqCk1LxE,GrC7lL8B,KAAL,GAAiB,GArP8B,EqCk1LlE,CrC7lLwB,KA

AL,GAAiB,GArP8B,CqCk1LxE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2B;MAYiB,Q;MAFb,IAA
I,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OA3iEG,gBAAR,iBAAQ,C;MA2iEhB,a
AAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,InC71L6E,0BmC61LzE,GnC/mL8B,KAAL,GAAiB,KA9
O+B,EmC61LnE,CnC/mLwB,KAAL,GAAiB,KA9O+B,CmC61LzE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;
K;mFAGX,yB;MAAA,sE;MA1kEI,8D;MA0kEJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,cAAc,sBA
AK,CAAL,C;QACd,gBAzlEgB,cAylEA,SAzlER,QAAQ,C;QA0lEhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;Q
AC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ
,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;
KA1BX,C;mFA6BA,yB;MAAA,sE;MA/lEI,8D;MA+lEJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,cA
Ac,sBAAK,CAAL,C;QACd,gBA9mEgB,cA8mEA,SA9mER,QAAQ,C;QA+mEhB,IAAI,cAAa,CAAjB,C;UAAoB,
OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;U
ACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,
OAAO,O;O;KA1BX,C;mFA6BA,yB;MAAA,sE;MApnEI,8D;MAonEJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6
B;QACrB,cAAc,sBAAK,CAAL,C;QACd,gBAnoEgB,cAmoEA,SAnoER,QAAQ,C;QAooEhB,IAAI,cAAa,CAAjB,
C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,
CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,
C;;;QAGnB,OAAO,O;O;KA1BX,C;mFA6BA,yB;MAAA,sE;MAzoEI,8D;MAyoEJ,sC;QAaI,IAAI,mBAAJ,C;UA
Ae,MAAM,6B;QACrB,cAAc,sBAAK,CAAL,C;QACd,gBAxpEgB,cAwpEA,SAxpER,QAAQ,C;QAypEhB,IAAI,c
AAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QA
AQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YA
CV,WAAW,C;;;QAGnB,OAAO,O;O;KA1BX,C;+FA6BA,yB;MA9rEI,8D;MA8rEJ,sC;QASI,IAAI,mBAAJ,C;UA
Ae,OAAO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBAzsEgB,cAysEA,SAzsER,QAAQ,C;QA0sEhB,IAAI,cAAa,
CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,s
BAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,W
AAW,C;;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MA/sEI,8D;MA+sEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OA
AO,I;QACtB,cAAc,sBAAK,CAAL,C;QACd,gBA1tEgB,cA0tEA,SA1tER,QAAQ,C;QA2tEhB,IAAI,cAAa,CAAjB
,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,
CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,
C;;;QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MAhuEI,8D;MAguEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;
QACtB,cAAc,sBAAK,CAAL,C;QACd,gBA3uEgB,cA2uEA,SA3uER,QAAQ,C;QA4uEhB,IAAI,cAAa,CAAjB,C;
UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CA
AL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;
QAGnB,OAAO,O;O;KAtBX,C;+FAyBA,yB;MAjvEI,8D;MAivEJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QAC
tB,cAAc,sBAAK,CAAL,C;QACd,gBA5vEgB,cA4vEA,SA5vER,QAAQ,C;QA6vEhB,IAAI,cAAa,CAAjB,C;UAA
oB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,
C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QA
GnB,OAAO,O;O;KAtBX,C;kFAyBA,yB;MAAA,sE;MAlyEI,8D;MpBvwHJ,iB;MoByiMA,sC;QAgBiB,Q;QAFb,I
AAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAlzEG,cAAR,iBAAQ,
C;QAkzEhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBnjMG,MAAO,KoBmjM
O,QpBnjMP,EoBmjMiB,CpBnjMjB,C;;QoBqjMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAjzEI,8D;Mp
B/wHJ,iB;MoBgkMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CA
AL,CAAT,C;QACF,OAj0EG,cAAR,iBAAQ,C;QAi0EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,C
AAT,C;UACR,WpB1kMG,MAAO,KoB0kMO,QpB1kMP,EoB0kMiB,CpB1kMjB,C;;QoB4kMd,OAAO,Q;O;KAp
BX,C;mFAuBA,yB;MAAA,sE;MAh0EI,8D;MpBvxHJ,iB;MoBulMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe
,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAh1EG,cAAR,iBAAQ,C;QAg1EhB,aAAU,C
AAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBjmMG,MAAO,KoBimMO,QpBjmMP,EoBim
MiB,CpBjmMjB,C;;QoBmmMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA/0EI,8D;MpB/xHJ,iB;MoB8
mMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QA

CF,OA/1EG,cAAR,iBAAQ,C;QA+1EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,
WpBxnMG,MAAO,KoBwnMO,QpBxnMP,EoBwnMiB,CpBxnMjB,C;;QoB0nMd,OAAO,Q;O;KApBX,C;mFAuB
A,yB;MAAA,sE;MA93EI,8D;MpBlxHJ,iB;MoBgpMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;
QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA94EG,cAAR,iBAAQ,C;QA84EhB,aAAU,CAAV,iB;UA
CI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB1pMG,MAAO,KoB0pMO,QpB1pMP,EoB0pMiB,CpB1pM
jB,C;;QoB4pMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA74EI,8D;MpB1xHJ,iB;MoBuqMA,sC;QAgB
iB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA75EG,cA
AR,iBAAQ,C;QA65EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBjrMG,MAA
O,KoBirMO,QpBjrMP,EoBirMiB,CpBjrMjB,C;;QoBmrMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA5
5EI,8D;MpBlyHJ,iB;MoB8rMA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sB
AAK,CAAL,CAAT,C;QACF,OA56EG,cAAR,iBAAQ,C;QA46EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAA
K,CAAL,CAAT,C;UACR,WpBxsMG,MAAO,KoBwsMO,QpBxsMP,EoBwsMiB,CpBxsMjB,C;;QoB0sMd,OAAO
,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA36EI,8D;MpB1yHJ,iB;MoBqtMA,sC;QAgBiB,Q;QAFb,IAAI,mBAA
J,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA37EG,cAAR,iBAAQ,C;QA27EhB
,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB/tMG,MAAO,KoB+tMO,QpB/tMP,E
oB+tMiB,CpB/tMjB,C;;QoBiuMd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA19EI,8D;MA09EJ,sC;QAci
B,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAx+EG,cA
AR,iBAAQ,C;QAw+EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,C
AAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAz+EI,8D;MAy+EJ,
sC;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAv
/EG,cAAR,iBAAQ,C;QAu/EhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BA
AW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAx/EI,8D;MA
w/EJ,sC;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QAC
F,OAtgFG,cAAR,iBAAQ,C;QAsgFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IA
AI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAvgFI
,8D;MAugFJ,sC;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,
C;QACF,OArhFG,cAAR,iBAAQ,C;QAqhFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;U
ACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;8FAuBA,yB;MAtjFI,8D;
MpBvwHJ,iB;MoB6zMA,sC;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAA
L,CAAT,C;QACF,OApkFG,cAAR,iBAAQ,C;QAokFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,C
AAT,C;UACR,WpBr0MG,MAAO,KoBq0MO,QpBr0MP,EoBq0MiB,CpBr0MjB,C;;QoBu0Md,OAAO,Q;O;KAlB
X,C;+FAqBA,yB;MAnkFI,8D;MpB/wHJ,iB;MoBk1MA,sC;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QA
CtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAjlFG,cAAR,iBAAQ,C;QAilFhB,aAAU,CAAV,iB;UACI,QA
AQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB11MG,MAAO,KoB01MO,QpB11MP,EoB01MiB,CpB11MjB,C;;
QoB41Md,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAhlFI,8D;MpBvxHJ,iB;MoBu2MA,sC;QAciB,Q;QAFb,IAAI,m
BAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA9lFG,cAAR,iBAAQ,C;QA8lFh
B,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB/2MG,MAAO,KoB+2MO,QpB/2M
P,EoB+2MiB,CpB/2MjB,C;;QoBi3Md,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA7lFI,8D;MpB/xHJ,iB;MoB43MA,s
C;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA3mF
G,cAAR,iBAAQ,C;QA2mFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBp4MG
,MAAO,KoBo4MO,QpBp4MP,EoBo4MiB,CpBp4MjB,C;;QoBs4Md,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA1oFI
,8D;MpBlxHJ,iB;MoB45MA,sC;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,C
AAL,CAAT,C;QACF,OAxpFG,cAAR,iBAAQ,C;QAwpFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAA
L,CAAT,C;UACR,WpBp6MG,MAAO,KoBo6MO,QpBp6MP,EoBo6MiB,CpBp6MjB,C;;QoBs6Md,OAAO,Q;O;K
AlBX,C;+FAqBA,yB;MAvpFI,8D;MpB1xHJ,iB;MoBi7MA,sC;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;
QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OArqFG,cAAR,iBAAQ,C;QAqqFhB,aAAU,CAAV,iB;UAC
I,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBz7MG,MAAO,KoBy7MO,QpBz7MP,EoBy7MiB,CpBz7Mj
B,C;;QoB27Md,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MApqFI,8D;MpBlyHJ,iB;MoBs8MA,sC;QAciB,Q;QAFb,IA

AI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAlrFG,cAAR,iBAAQ,C;QA
krFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB98MG,MAAO,KoB88MO,Qp
B98MP,EoB88MiB,CpB98MjB,C;;QoBg9Md,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAjrFI,8D;MpB1yHJ,iB;MoB
29MA,sC;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,
OA/rFG,cAAR,iBAAQ,C;QA+rFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB
n+MG,MAAO,KoBm+MO,QpBn+MP,EoBm+MiB,CpBn+MjB,C;;QoBq+Md,OAAO,Q;O;KAlBX,C;+FAqBA,yB
;MA9tFI,8D;MA8tFJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,C
AAT,C;QACF,OA1uFG,cAAR,iBAAQ,C;QA0uFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT
,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA3uF
I,8D;MA2uFJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C
;QACF,OAvvFG,cAAR,iBAAQ,C;QAuvFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UA
CR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAxvFI,8D;
MAwvFJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QA
CF,OApwFG,cAAR,iBAAQ,C;QAowFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,
IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MArwFI,8D;MAq
wFJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,O
AjxFG,cAAR,iBAAQ,C;QAixFhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2
BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;2FAqBA,yB;MAAA,sE;MAlzFI,8D;M
AkzFJ,kD;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QA
CF,OAh0FG,cAAR,iBAAQ,C;QAg0FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,I
AAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;K
ApBX,C;0FAuBA,yB;MAAA,sE;MAj0FI,8D;MAi0FJ,kD;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QA
CrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/0FG,cAAR,iBAAQ,C;QA+0FhB,aAAU,CAAV,iB;UACI,Q
AAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAt
C,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MAh1FI,8D;MAg1FJ,kD;QAciB,
Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA91FG,cAA
R,iBAAQ,C;QA81FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAA
Q,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,
yB;MAAA,sE;MA/1FI,8D;MA+1FJ,kD;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,
sBAAK,CAAL,CAAT,C;QACF,OA72FG,cAAR,iBAAQ,C;QA62FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBA
AK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAA
W,C;;;QAGnB,OAAO,Q;O;KApBX,C;uGAuBA,yB;MA94FI,8D;MA84FJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;U
AAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA15FG,cAAR,iBAAQ,C;QA05FhB,aAAU,
CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CA
AX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;sGAqBA,yB;MA35FI,8D;MA25FJ,kD;
QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAv6FG,
cAAR,iBAAQ,C;QAu6FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,
SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;uGA
qBA,yB;MAx6FI,8D;MAw6FJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAA
K,CAAL,CAAT,C;QACF,OAp7FG,cAAR,iBAAQ,C;QAo7FhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,C
AAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAAkC,CAAtC,C;YACI,WAAW,C;;;
QAGnB,OAAO,Q;O;KAlBX,C;uGAqBA,yB;MAr7FI,8D;MAq7FJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,O
AAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAj8FG,cAAR,iBAAQ,C;QAi8FhB,aAAU,CAAV,i
B;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAkB,CAAlB,CAAX,GAA
kC,CAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;IAqBA,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QA
Ae,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OA1+FG,gBAAR,iBAAQ,C;MA0+FhB,aAAU,CAAV,iB;
QACI,QAAQ,sBAAK,CAAL,C;QACR,IpC5vN8D,YoC4vN1D,GpC5vN2E,KAAjB,EoC4vNpD,CpC5vNiF,KAA7
B,CoC4vN1D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,

OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OAj/FG,gBAAR,iBAAQ,C;MAi/FhB,aAAU,CAAV,iB;QACI,
QAAQ,sBAAK,CAAL,C;QACR,InBnwN+D,amBmwN3D,GnBnwN6E,KAAlB,EmBmwNrD,CnBnwNmF,KAA9
B,CmBmwN3D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QA
Ae,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OAx/FG,gBAAR,iBAAQ,C;MAw/FhB,aAAU,CAAV,iB;Q
ACI,QAAQ,sBAAK,CAAL,C;QACR,IrC1yN4E,0BqC0yNxE,GrCrjN8B,KAAL,GAAiB,GArP8B,EqC0yNlE,CrCr
jNwB,KAAL,GAAiB,GArP8B,CqC0yNxE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;M
AFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OA//FG,gBAAR,iBAAQ,C;MA+/
FhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,InCjzN6E,0BmCizNzE,GnCnkN8B,KAAL,GAAiB,
KA9O+B,EmCizNnE,CnCnkNwB,KAAL,GAAiB,KA9O+B,CmCizNzE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAA
O,G;K;IAGX,2C;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,
OA1iGG,gBAAR,iBAAQ,C;MA0iGhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SA
AQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2C;MAYiB,
Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OArjGG,gBAAR,iBAAQ,
C;MAqjGhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,
CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2C;MAYiB,Q;MAFb,IAAI,mBAAJ,C;
QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OAhkGG,gBAAR,iBAAQ,C;MAgkGhB,aAAU,CAA
V,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;
UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2C;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB
,UAAU,sBAAK,CAAL,C;MACG,OA3kGG,gBAAR,iBAAQ,C;MA2kGhB,aAAU,CAAV,iB;QACI,QAAQ,sBAA
K,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE
9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;M
ACG,OAlnGG,gBAAR,iBAAQ,C;MAknGhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UA
AW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;M
AQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OAznGG,gBAAR,iBA
AQ,C;MAynGhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CA
Ab,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ
,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OAhoGG,gBAAR,iBAAQ,C;MAgoGhB,aAAU,CAA
V,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;
UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,U
AAU,sBAAK,CAAL,C;MACG,OAvoGG,gBAAR,iBAAQ,C;MAuoGhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,
CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9
C,OAAO,G;K;IAGX,2B;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;
MACG,OAlrGG,gBAAR,iBAAQ,C;MAkrGhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IpCp8N8
D,YoCo8N1D,GpCp8N2E,KAAjB,EoCo8NpD,CpCp8NiF,KAA7B,CoCo8N1D,IAAJ,C;UAAa,MAAM,C;;MAEv
B,OAAO,G;K;IAGX,2B;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;
MACG,OA7rGG,gBAAR,iBAAQ,C;MA6rGhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,InB/8N+
D,amB+8N3D,GnB/8N6E,KAAlB,EmB+8NrD,CnB/8NmF,KAA9B,CmB+8N3D,IAAJ,C;UAAa,MAAM,C;;MAE
vB,OAAO,G;K;IAGX,2B;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C
;MACG,OAxsGG,gBAAR,iBAAQ,C;MAwsGhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IrC1/N
4E,0BqC0/NxE,GrCrwN8B,KAAL,GAAiB,GArP8B,EqC0/NlE,CrCrwNwB,KAAL,GAAiB,GArP8B,CqC0/NxE,I
AAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2B;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;M
ACrB,UAAU,sBAAK,CAAL,C;MACG,OAntGG,gBAAR,iBAAQ,C;MAmtGhB,aAAU,CAAV,iB;QACI,QAAQ,s
BAAK,CAAL,C;QACR,InCrgO6E,0BmCqgOzE,GnCvxN8B,KAAL,GAAiB,KA9O+B,EmCqgOnE,CnCvxNwB,K
AAL,GAAiB,KA9O+B,CmCqgOzE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;mFAGX,yB;MAAA,sE;MAl
vGI,8D;MAkvGJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,cAAc,sBAAK,CAAL,C;QACd,gBAjwGgB
,cAiwGA,SAjwGR,QAAQ,C;QAkwGhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAAS,OAAT,C;
QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;UACR,IAAI
,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KA1BX,C;mFA6BA,yB;MA

AA,sE;MAvwGI,8D;MAuwGJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,cAAc,sBAAK,CAAL,C;QAC
d,gBAtxGgB,cAsxGA,SAtxGR,QAAQ,C;QAuxGhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3B,eAAe,SAA
S,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SAAS,CAAT,C;
UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KA1BX,C;mFA6
BA,yB;MAAA,sE;MA5xGI,8D;MA4xGJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,cAAc,sBAAK,CA
AL,C;QACd,gBA3yGgB,cA2yGA,SA3yGR,QAAQ,C;QA4yGhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QAC3
B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,SA
AS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;KA
1BX,C;mFA6BA,yB;MAAA,sE;MAjzGI,8D;MAizGJ,sC;QAaI,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,cAAc,
sBAAK,CAAL,C;QACd,gBAh0GgB,cAg0GA,SAh0GR,QAAQ,C;QAi0GhB,IAAI,cAAa,CAAjB,C;UAAoB,OAA
O,O;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,
QAAQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAA
O,O;O;KA1BX,C;+FA6BA,yB;MAt2GI,8D;MAs2GJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sB
AAK,CAAL,C;QACd,gBAj3GgB,cAi3GA,SAj3GR,QAAQ,C;QAk3GhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O
;QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QA
AQ,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,
O;O;KAtBX,C;+FAyBA,yB;MAv3GI,8D;MAu3GJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBA
AK,CAAL,C;QACd,gBAl4GgB,cAk4GA,SAl4GR,QAAQ,C;QAm4GhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;
QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAA
Q,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;
O;KAtBX,C;+FAyBA,yB;MAx4GI,8D;MAw4GJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAA
K,CAAL,C;QACd,gBAn5GgB,cAm5GA,SAn5GR,QAAQ,C;QAo5GhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;
QAC3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAA
Q,SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;
O;KAtBX,C;+FAyBA,yB;MAz5GI,8D;MAy5GJ,sC;QASI,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,cAAc,sBAAK,
CAAL,C;QACd,gBAp6GgB,cAo6GA,SAp6GR,QAAQ,C;QAq6GhB,IAAI,cAAa,CAAjB,C;UAAoB,OAAO,O;QA
C3B,eAAe,SAAS,OAAT,C;QACf,aAAU,CAAV,OAAa,SAAb,M;UACI,QAAQ,sBAAK,CAAL,C;UACR,QAAQ,
SAAS,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,UAAU,C;YACV,WAAW,C;;;QAGnB,OAAO,O;O;
KAtBX,C;kFAyBA,yB;MAAA,sE;MA18GI,8D;MpBnjHJ,iB;MoB6/NA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UA
Ae,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA19GG,cAAR,iBAAQ,C;QA09GhB,aAA
U,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBvgOG,MAAO,KoBugOO,QpBvgOP,EoBu
gOiB,CpBvgOjB,C;;QoBygOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAz9GI,8D;MpB3jHJ,iB;MoBoh
OA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QAC
F,OAz+GG,cAAR,iBAAQ,C;QAy+GhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,
WpB9hOG,MAAO,KoB8hOO,QpB9hOP,EoB8hOiB,CpB9hOjB,C;;QoBgiOd,OAAO,Q;O;KApBX,C;mFAuBA,y
B;MAAA,sE;MAx+GI,8D;MpBnkHJ,iB;MoB2iOA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QAC
rB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAx/GG,cAAR,iBAAQ,C;QAw/GhB,aAAU,CAAV,iB;UACI,Q
AAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBrjOG,MAAO,KoBqjOO,QpBrjOP,EoBqjOiB,CpBrjOjB,C;;QoB
ujOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAv/GI,8D;MpB3kHJ,iB;MoBkkOA,sC;QAgBiB,Q;QAFb,
IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAvgHG,cAAR,iBAAQ
,C;QAugHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB5kOG,MAAO,KoB4kO
O,QpB5kOP,EoB4kOiB,CpB5kOjB,C;;QoB8kOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAtiHI,8D;Mp
B9jHJ,iB;MoBomOA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CA
AL,CAAT,C;QACF,OAtjHG,cAAR,iBAAQ,C;QAsjHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,C
AAT,C;UACR,WpB9mOG,MAAO,KoB8mOO,QpB9mOP,EoB8mOiB,CpB9mOjB,C;;QoBgnOd,OAAO,Q;O;KA
pBX,C;mFAuBA,yB;MAAA,sE;MArjHI,8D;MpBtkHJ,iB;MoB2nOA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAA
e,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OArkHG,cAAR,iBAAQ,C;QAqkHhB,aAAU,
CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBroOG,MAAO,KoBqoOO,QpBroOP,EoBqoO

iB,CpBroOjB,C;;QoBuoOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MApkHI,8D;MpB9kHJ,iB;MoBkpO
A,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,
OAplHG,cAAR,iBAAQ,C;QAolHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpB
5pOG,MAAO,KoB4pOO,QpB5pOP,EoB4pOiB,CpB5pOjB,C;;QoB8pOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;M
AAA,sE;MAnlHI,8D;MpBtlHJ,iB;MoByqOA,sC;QAgBiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,e
AAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAnmHG,cAAR,iBAAQ,C;QAmmHhB,aAAU,CAAV,iB;UACI,QA
AQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBnrOG,MAAO,KoBmrOO,QpBnrOP,EoBmrOiB,CpBnrOjB,C;;Qo
BqrOd,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAloHI,8D;MAkoHJ,sC;QAciB,Q;QAFb,IAAI,mBAAJ,C
;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAhpHG,cAAR,iBAAQ,C;QAgpHhB,a
AAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WA
AW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAjpHI,8D;MAipHJ,sC;QAciB,Q;QAFb,IAAI,
mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/pHG,cAAR,iBAAQ,C;QA
+pHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;Y
ACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MAhqHI,8D;MAgqHJ,sC;QAciB,Q;QA
Fb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA9qHG,cAAR,iBA
AQ,C;QA8qHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,K
AAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;mFAuBA,yB;MAAA,sE;MA/qHI,8D;MA+qHJ,sC;QAc
iB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA7rHG,cA
AR,iBAAQ,C;QA6rHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,C
AAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;8FAuBA,yB;MA9tHI,8D;MpBnjHJ,iB;MoBixO
A,sC;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA5
uHG,cAAR,iBAAQ,C;QA4uHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBzx
OG,MAAO,KoByxOO,QpBzxOP,EoByxOiB,CpBzxOjB,C;;QoB2xOd,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA3u
HI,8D;MpB3jHJ,iB;MoBsyOA,sC;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,
CAAL,CAAT,C;QACF,OAzvHG,cAAR,iBAAQ,C;QAyvHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CA
AL,CAAT,C;UACR,WpB9yOG,MAAO,KoB8yOO,QpB9yOP,EoB8yOiB,CpB9yOjB,C;;QoBgzOd,OAAO,Q;O;K
AlBX,C;+FAqBA,yB;MAxvHI,8D;MpBnkHJ,iB;MoB2zOA,sC;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;
QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAtwHG,cAAR,iBAAQ,C;QAswHhB,aAAU,CAAV,iB;UA
CI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBn0OG,MAAO,KoBm0OO,QpBn0OP,EoBm0OiB,CpBn0Oj
B,C;;QoBq0Od,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MArwHI,8D;MpB3kHJ,iB;MoBg1OA,sC;QAciB,Q;QAFb,IA
AI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAnxHG,cAAR,iBAAQ,C;Q
AmxHhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBx1OG,MAAO,KoBw1OO,
QpBx1OP,EoBw1OiB,CpBx1OjB,C;;QoB01Od,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAlzHI,8D;MpB9jHJ,iB;Mo
Bg3OA,sC;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF
,OAh0HG,cAAR,iBAAQ,C;QAg0HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,Wp
Bx3OG,MAAO,KoBw3OO,QpBx3OP,EoBw3OiB,CpBx3OjB,C;;QoB03Od,OAAO,Q;O;KAlBX,C;+FAqBA,yB;
MA/zHI,8D;MpBtkHJ,iB;MoBq4OA,sC;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sB
AAK,CAAL,CAAT,C;QACF,OA70HG,cAAR,iBAAQ,C;QA60HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAA
K,CAAL,CAAT,C;UACR,WpB74OG,MAAO,KoB64OO,QpB74OP,EoB64OiB,CpB74OjB,C;;QoB+4Od,OAAO,
Q;O;KAlBX,C;+FAqBA,yB;MA50HI,8D;MpB9kHJ,iB;MoB05OA,sC;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,O
AAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA11HG,cAAR,iBAAQ,C;QA01HhB,aAAU,CAAV,
iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBl6OG,MAAO,KoBk6OO,QpBl6OP,EoBk6OiB,CpBl
6OjB,C;;QoBo6Od,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAz1HI,8D;MpBtlHJ,iB;MoB+6OA,sC;QAciB,Q;QAFb,I
AAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAv2HG,cAAR,iBAAQ,C;
QAu2HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,WpBv7OG,MAAO,KoBu7OO,
QpBv7OP,EoBu7OiB,CpBv7OjB,C;;QoBy7Od,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAt4HI,8D;MAs4HJ,sC;QA
YiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAl5HG,cA
AR,iBAAQ,C;QAk5HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,C

AAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAn5HI,8D;MAm5HJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/5HG,cAAR,iBAAQ,C;QA+5HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MAh6HI,8D;MAg6HJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA56HG,cAAR,iBAAQ,C;QA46HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;+FAqBA,yB;MA76HI,8D;MA66HJ,sC;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAz7HG,cAAR,iBAAQ,C;QAy7HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,2BAAW,CAAX,KAAJ,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;2FAqBA,yB;MAAA,sE;MA19HI,8D;MA09HJ,kD;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAx+HG,cAAR,iBAAQ,C;QAw+HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAAkB,CAAAlB,CAAAX,GAAAkC,CAAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;0FAuBA,yB;MAAA,sE;MAz+HI,8D;MAy+HJ,kD;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAv/HG,cAAR,iBAAQ,C;QAu/HhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAAkB,CAAAlB,CAAAX,GAAAkC,CAAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MAx/HI,8D;MAw/HJ,kD;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAtgIG,cAAR,iBAAQ,C;QAsgIhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAAkB,CAAAlB,CAAAX,GAAAkC,CAAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;2FAuBA,yB;MAAA,sE;MAvgII,8D;MAugIJ,kD;QAciB,Q;QAFb,IAAI,mBAAJ,C;UAAe,MAAM,6B;QACrB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OArhIG,cAAR,iBAAQ,C;QAqhIhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAAkB,CAAAlB,CAAAX,GAAAkC,CAAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KApBX,C;uGAuBA,yB;MAtjII,8D;MAsjIJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAlkIG,cAAR,iBAAQ,C;QAkkIhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAAkB,CAAAlB,CAAAX,GAAAkC,CAAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;sGAqBA,yB;MAnkII,8D;MAmkIJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA/kIG,cAAR,iBAAQ,C;QA+kIhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAAkB,CAAAlB,CAAAX,GAAAkC,CAAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;uGAqBA,yB;MAhlII,8D;MAglIJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OA5lIG,cAAR,iBAAQ,C;QA4lIhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAAkB,CAAAlB,CAAAX,GAAAkC,CAAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;uGAqBA,yB;MA7lII,8D;MA6lIJ,kD;QAYiB,Q;QAFb,IAAI,mBAAJ,C;UAAe,OAAO,I;QACtB,eAAe,SAAS,sBAAK,CAAL,CAAT,C;QACF,OAzmIG,cAAR,iBAAQ,C;QAymIhB,aAAU,CAAV,iB;UACI,QAAQ,SAAS,sBAAK,CAAL,CAAT,C;UACR,IAAI,UAAW,SAAQ,QAAR,EAAAkB,CAAAlB,CAAAX,GAAAkC,CAAAtC,C;YACI,WAAW,C;;;QAGnB,OAAO,Q;O;KAlBX,C;IAqBA,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OAlpIG,gBAAR,iBAAQ,C;MAkpIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IpCp6P8D,YoCo6P1D,GpCp6P2E,KAAjB,EoCo6PpD,CpCp6PiF,KAA7B,CoCo6P1D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OAzpIG,gBAAR,iBAAQ,C;MAypIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,InB36P+D,amB26P3D,GnB36P6E,KAAlB,EmB26PrD,CnB36PmF,KAA9B,CmB26P3D,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OAhqIG,gBAAR,iBAAQ,C;MAgqIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IrCl9P4E,0BqCk9PxE,GrC7tP8B,KAAL,GAAAiB,GArP8B,EqCk9PlE,CrC7tPwB,KAAL,GAAAiB,GArP8B,CqCk9PxE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,iC;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OAvqIG,gBAAR,iBAAQ,C;MAuqIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,InCz9P6E,0BmCy9PzE,GnC3uP8B,KAAL,GAAAiB,KA9O+B,EmCy9PnE,CnC3uPwB,KAAL,GAA

iB,KA9O+B,CmCy9PzE,IAAJ,C;UAAa,MAAM,C;;MAEvB,OAAO,G;K;IAGX,2C;MAYiB,Q;MAFb,IAAI,mBA
AJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OAltIG,gBAAR,iBAAQ,C;MAktIhB,aAAU,C
AAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAj
C,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2C;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MA
CrB,UAAU,sBAAK,CAAL,C;MACG,OA7tIG,gBAAR,iBAAQ,C;MA6tIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAA
K,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE
9C,OAAO,G;K;IAGX,2C;MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C
;MACG,OAxuIG,gBAAR,iBAAQ,C;MAwuIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,U
AAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,2C;
MAYiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,MAAM,6B;MACrB,UAAU,sBAAK,CAAL,C;MACG,OAnvIG,gBAAR,
iBAAQ,C;MAmvIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa
,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mB
AAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OA1xIG,gBAAR,iBAAQ,C;MA0xIhB,aAAU,C
AAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAj
C,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACt
B,UAAU,sBAAK,CAAL,C;MACG,OAjyIG,gBAAR,iBAAQ,C;MAiyIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,
CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9
C,OAAO,G;K;IAGX,iD;MAQiB,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MA
CG,OAxyIG,gBAAR,iBAAQ,C;MAwyIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW
,SAAQ,GAAR,EAAa,CAAb,CAAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;IAGX,iD;MAQi
B,Q;MAFb,IAAI,mBAAJ,C;QAAe,OAAO,I;MACtB,UAAU,sBAAK,CAAL,C;MACG,OA/yIG,gBAAR,iBAAQ,C
;MA+yIhB,aAAU,CAAV,iB;QACI,QAAQ,sBAAK,CAAL,C;QACR,IAAI,UAAW,SAAQ,GAAR,EAAa,CAAb,C
AAX,GAA6B,CAAjC,C;UAAoC,MAAM,C;;MAE9C,OAAO,G;K;iFAGX,qB;MASI,OAAO,mB;K;iFAGX,qB;M
ASI,OAAO,mB;K;iFAGX,qB;MASI,OAAO,mB;K;iFAGX,qB;MASI,OAAO,mB;K;iFAGX,gC;MASoB,Q;MAAA
,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,O
AAO,I;K;iFAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,
CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MASoB,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;Q
AAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MACrD,OAAO,I;K;iFAGX,gC;MASoB,Q;M
AAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,IAAI,UAAU,OAAV,CAAJ,C;UAAwB,OAAO,K;;MAC
rD,OAAO,I;K;qFAGX,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OAAP,
C;;MAArC,gB;K;qFAGJ,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,OA
AP,C;;MAArC,gB;K;qFAGJ,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAAO,
OAAP,C;;MAArC,gB;K;qFAGJ,6B;MAOmC,Q;MAAA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAM,OAA
O,OAAP,C;;MAArC,gB;K;mGAGJ,6B;MAtgFiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,s
B;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAghFnB,gB;K;mGAGJ,6B;MAtgFiB,gB;MADb,Y
AAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;M
AghFnB,gB;K;mGAGJ,6B;MAtgFiB,gB;MADb,YAAY,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,Q
AAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAghFnB,gB;K;mGAGJ,6B;MAtgFiB,gB;MADb,YAAY,C;MAC
C,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QAAM,QAAO,cAAP,EAAO,sBAAP,WAAgB,IAAhB,C;;MAghFnB,gB;K
;qFAGJ,yB;MAAA,4F;MA9gJI,8D;MA8gJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+
BAA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,OAjiJD,cAAR,iBAAQ,C;QAiiJhB,iBAAc,CAAd,yB;UACI,
cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAElB,OAAO,W;O;KAtBX,C;qFAyBA,yB;MAAA,4F;
MA/hJI,8D;MA+hJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAA
kB,sBAAK,CAAL,C;QACD,OAljJD,cAAR,iBAAQ,C;QAkjJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,E
AAuB,sBAAK,KAAL,CAAvB,C;;QAElB,OAAO,W;O;KAtBX,C;qFAyBA,yB;MAAA,4F;MAhjJI,8D;MAgjJJ,uC;
QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,O
ACD,OAnkJD,cAAR,iBAAQ,C;QAmkJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL
,CAAvB,C;;QAElB,OAAO,W;O;KAtBX,C;qFAyBA,yB;MAAA,4F;MAjkJI,8D;MAikJJ,uC;QAmBqB,Q;QAHjB,I

AAI,mBAAJ,C;UACI,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,OAplJD,cAAR,i
BAAQ,C;QAolJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAElB,OA
AO,W;O;KAtBX,C;mGAyBA,yB;MAAA,4F;MAlnJI,8D;MAknJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI
,MAAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,OAroJD,cAAR,iBAAQ,C;QAqoJhB,iB
AAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAElB,OAAO,
W;O;KAtBX,C;mGAyBA,yB;MAAA,4F;MAnoJI,8D;MAmoJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,M
AAM,mCAA8B,+BAA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,OAtpJD,cAAR,iBAAQ,C;QAspJhB,iBAA
c,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;
KAtBX,C;mGAyBA,yB;MAAA,4F;MAppJI,8D;MAopJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,
mCAA8B,+BAA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,OAvqJD,cAAR,iBAAQ,C;QAuqJhB,iBAAc,CA
Ad,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KAt
BX,C;mGAyBA,yB;MAAA,4F;MArqJI,8D;MAqqJJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,MAAM,mC
AA8B,+BAA9B,C;QACV,kBAAkB,sBAAK,CAAL,C;QACD,OAxrJD,cAAR,iBAAQ,C;QAwrJhB,iBAAc,CAAd,
yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KAtBX,
C;+GAyBA,yB;MAttJI,8D;MAstJJ,uC;QAkBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAAkB,sBA
AK,CAAL,C;QACD,OAxuJD,cAAR,iBAAQ,C;QAwuJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,
WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KArBX,C;+GAwBA,yB;MAtuJI,8D;MAsuJJ,uC
;QAkBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAAkB,sBAAK,CAAL,C;QACD,OAxvJD,cAAR,i
BAAQ,C;QAwvJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA
9B,C;;QAElB,OAAO,W;O;KArBX,C;+GAwBA,yB;MAtvJI,8D;MAsvJJ,uC;QAkBqB,Q;QAHjB,IAAI,mBAAJ,C;
UACI,OAAO,I;QACX,kBAAkB,sBAAK,CAAL,C;QACD,OAxwJD,cAAR,iBAAQ,C;QAwwJhB,iBAAc,CAAd,y
B;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KArBX,C
;+GAwBA,yB;MAtwJI,8D;MAswJJ,uC;QAkBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAAkB,sB
AAK,CAAL,C;QACD,OAxxJD,cAAR,iBAAQ,C;QAwxJhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,KAAV,EAAiB
,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;;QAElB,OAAO,W;O;KArBX,C;iGAwBA,yB;MAtzJI,8D;MAszJJ,uC;
QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAAkB,sBAAK,CAAL,C;QACD,OAz0JD,cAAR,i
BAAQ,C;QAy0JhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAElB,O
AAO,W;O;KAtBX,C;iGAyBA,yB;MAv0JI,8D;MAu0JJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;
QACX,kBAAkB,sBAAK,CAAL,C;QACD,OA11JD,cAAR,iBAAQ,C;QA01JhB,iBAAc,CAAd,yB;UACI,cAAc,U
AAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAElB,OAAO,W;O;KAtBX,C;iGAyBA,yB;MAx1JI,8D;MAw1
JJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OAAO,I;QACX,kBAAkB,sBAAK,CAAL,C;QACD,OA32JD,c
AAR,iBAAQ,C;QA22JhB,iBAAc,CAAd,yB;UACI,cAAc,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QA
ElB,OAAO,W;O;KAtBX,C;iGAyBA,yB;MAz2JI,8D;MAy2JJ,uC;QAmBqB,Q;QAHjB,IAAI,mBAAJ,C;UACI,OA
AO,I;QACX,kBAAkB,sBAAK,CAAL,C;QACD,OA53JD,cAAR,iBAAQ,C;QA43JhB,iBAAc,CAAd,yB;UACI,cA
Ac,UAAU,WAAV,EAAuB,sBAAK,KAAL,CAAvB,C;;QAElB,OAAO,W;O;KAtBX,C;+FAyBA,yB;MAAA,4F;M
A15JI,8D;MA05JJ,uC;QAkB0B,UAEU,M;QAJhC,YA16JgB,cAAR,iBAAQ,C;QA26JhB,IAAI,QAAQ,CAAZ,C;U
AAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CAA
hB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAtBX,C
;+FAyBA,yB;MAAA,4F;MA36JI,8D;MA26JJ,uC;QAkB0B,UAEU,M;QAJhC,YA37JgB,cAAR,iBAAQ,C;QA47Jh
B,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,uBAAI,YAAJ,EAAI,oBAAJ,Q;
QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;Q
AElB,OAAO,W;O;KAtBX,C;+FAyBA,yB;MAAA,4F;MA57JI,8D;MA47JJ,uC;QAkB0B,UAEU,M;QAJhC,YA58J
gB,cAAR,iBAAQ,C;QA68JhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,uB
AAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UA
AV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAtBX,C;+FAyBA,yB;MAAA,4F;MA78JI,8D;MA68JJ,uC;QAkB0
B,UAEU,M;QAJhC,YA79JgB,cAAR,iBAAQ,C;QA89JhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA
9B,C;QACrB,kBAAkB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,uBA
AI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAtBX,C;6GAyBA,yB;MAAA,4F;MA9

/JI,8D;MA8/JJ,uC;QAkB0B,Q;QAFtB,YA9gKgB,cAAR,iBAAQ,C;QA+gKhB,IAAI,QAAQ,CAAZ,C;UAAe,MAA
M,mCAA8B,+BAA9B,C;QACrB,kBAAkB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CAAhB,C;UAC
I,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAvBX,
C;6GA0BA,yB;MAAA,4F;MAhhKI,8D;MAghKJ,uC;QAkB0B,Q;QAFtB,YAhiKgB,cAAR,iBAAQ,C;QAiiKhB,I
AAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QA
ClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UAC
d,qB;;QAEJ,OAAO,W;O;KAvBX,C;6GA0BA,yB;MAAA,4F;MAliKI,8D;MAkiKJ,uC;QAkB0B,Q;QAFtB,YAljK
gB,cAAR,iBAAQ,C;QAmjKhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8B,+BAA9B,C;QACrB,kBAAkB,u
BAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UAAU,KAAV,EAAiB,sBAAI,KAAJ
,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAvBX,C;6GA0BA,yB;MAAA,4F;MApjKI,8D;MAoj
KJ,uC;QAkB0B,Q;QAFtB,YApkKgB,cAAR,iBAAQ,C;QAqkKhB,IAAI,QAAQ,CAAZ,C;UAAe,MAAM,mCAA8
B,+BAA9B,C;QACrB,kBAAkB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UA
AU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAvBX,C;yHA0BA,
yB;MAtmKI,8D;MAsmKJ,uC;QAiB0B,Q;QAFtB,YArnKgB,cAAR,iBAAQ,C;QAsnKhB,IAAI,QAAQ,CAAZ,C;U
AAe,OAAO,I;QACtB,kBAAkB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UA
AU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAtBX,C;yHAyBA,
yB;MAvnKI,8D;MAunKJ,uC;QAiB0B,Q;QAFtB,YAtoKgB,cAAR,iBAAQ,C;QAuoKhB,IAAI,QAAQ,CAAZ,C;U
AAe,OAAO,I;QACtB,kBAAkB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,UA
AU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAtBX,C;yHAyBA,
yB;MAxoKI,8D;MAwoKJ,uC;QAiB0B,Q;QAFtB,YAvpKgB,cAAR,iBAAQ,C;QAwpKhB,IAAI,QAAQ,CAAZ,C;
UAAe,OAAO,I;QACtB,kBAAkB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,U
AAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAtBX,C;yHAyB
A,yB;MAzpKI,8D;MAypKJ,uC;QAiB0B,Q;QAFtB,YAxqKgB,cAAR,iBAAQ,C;QAyqKhB,IAAI,QAAQ,CAAZ,C
;UAAe,OAAO,I;QACtB,kBAAkB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CAAhB,C;UACI,cAAc,
UAAU,KAAV,EAAiB,sBAAI,KAAJ,CAAjB,EAA6B,WAA7B,C;UACd,qB;;QAEJ,OAAO,W;O;KAtBX,C;2GAy
BA,yB;MA1sKI,8D;MA0sKJ,uC;QAkB0B,UAEU,M;QAJhC,YA1tKgB,cAAR,iBAAQ,C;QA2tKhB,IAAI,QAAQ,
CAAZ,C;UAAe,OAAO,I;QACtB,kBAAkB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CAAhB,C;UAC
I,cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAtBX,C;2GAyBA,
yB;MA3tKI,8D;MA2tKJ,uC;QAkB0B,UAEU,M;QAJhC,YA3uKgB,cAAR,iBAAQ,C;QA4uKhB,IAAI,QAAQ,CA
AZ,C;UAAe,OAAO,I;QACtB,kBAAkB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CAAhB,C;UACI,c
AAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAtBX,C;2GAyBA,y
B;MA5uKI,8D;MA4uKJ,uC;QAkB0B,UAEU,M;QAJhC,YA5vKgB,cAAR,iBAAQ,C;QA6vKhB,IAAI,QAAQ,CA
AZ,C;UAAe,OAAO,I;QACtB,kBAAkB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CAAhB,C;UACI,c
AAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAtBX,C;2GAyBA,y
B;MA7vKI,8D;MA6vKJ,uC;QAkB0B,UAEU,M;QAJhC,YA7wKgB,cAAR,iBAAQ,C;QA8wKhB,IAAI,QAAQ,C
AAZ,C;UAAe,OAAO,I;QACtB,kBAAkB,uBAAI,YAAJ,EAAI,oBAAJ,Q;QAClB,OAAO,SAAS,CAAhB,C;UACI,
cAAc,UAAU,uBAAI,cAAJ,EAAI,sBAAJ,UAAV,EAAwB,WAAxB,C;;QAElB,OAAO,W;O;KAtBX,C;+FAyBA,y
B;MAAA,gD;MAAA,gE;MAAA,gD;QAkBoB,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACCc,
kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBzjSO,W;QqB0jSP,kBAAkB,O;QACF,2B;QAAhB
,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;
QAEX,OAAO,M;O;KAtBX,C;+FAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAkBoB,Q;QAHhB,IAAI,mBAAJ
,C;UAAe,OAAO,OAAO,OAAP,C;QACCc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBllSO,W
;QqBmlSP,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,O
AAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;+FAyBA,yB;MAAA,gD;MAAA,gE;MA
AA,gD;QAkBoB,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACCc,kBAAvB,eAAa,iBAAO,CAA
P,IAAb,C;QAA+B,8B;QAA5C,arB3mSO,W;QqB4mSP,kBAAkB,O;QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAg
B,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtB
X,C;+FAyBA,yB;MAAA,gD;MAAA,gE;MAAA,gD;QAkBoB,Q;QAHhB,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,

OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arBpoSO,W;QqBqoSP,kBAAkB,O;
QACF,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAc,UAAU,WAAV,EAAuB,OAAvB,C;UACd,MAAO
,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;6GAyBA,yB;MAAA,gD;MAAA,gE;MAl7KI,0D;MAk7KJ,gD;Q
AmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C
;QAA+B,8B;QAA5C,arB9pSO,W;QqB+pSP,kBAAkB,O;QACJ,OAr8KE,YAAR,iBAAQ,C;QAq8KF,mB;QAAA,
kB;QAAA,kB;QAAd,0D;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,
MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;6GA0BA,yB;MAAA,gD;MAAA,gE;MAp8KI,0D;MAo8
KJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,CAA
P,IAAb,C;QAA+B,8B;QAA5C,arBxrSO,W;QqByrSP,kBAAkB,O;QACJ,OAv9KE,YAAR,iBAAQ,C;QAu9KF,mB
;QAAA,kB;QAAA,kB;QAAd,0D;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;
UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;6GA0BA,yB;MAAA,gD;MAAA,gE;MAt9KI,0D;
MAs9KJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,iBAAO,
CAAP,IAAb,C;QAA+B,8B;QAA5C,arBltSO,W;QqBmtSP,kBAAkB,O;QACJ,OAz+KE,YAAR,iBAAQ,C;QAy+K
F,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA
9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;6GA0BA,yB;MAAA,gD;MAAA,gE;MAx+
KI,0D;MAw+KJ,gD;QAmBkB,gC;QAHd,IAAI,mBAAJ,C;UAAe,OAAO,OAAO,OAAP,C;QACc,kBAAvB,eAAa,
iBAAO,CAAP,IAAb,C;QAA+B,8B;QAA5C,arB5uSO,W;QqB6uSP,kBAAkB,O;QACJ,OA3/KE,YAAR,iBAAQ,C
;QA2/KF,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,cAAc,UAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KA
AL,CAA9B,C;UACd,MAAO,WAAI,WAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;mGA0BA,yB;MAAA,qD;MAAA,g
E;MAAA,uC;QAkB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAlB,C;Q
ACmC,kBAAtB,eAAgB,cAAhB,C;QAA8B,sBAAI,aAAJ,C;QAA3C,arBtwSO,W;QqBuwSe,qB;QAAtB,iBAAc,C
AAd,wB;UACI,gBAAc,UAAU,aAAV,EAAuB,sBAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,
OAAO,M;O;KAtBX,C;mGAyBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAkB0B,Q;QAHtB,IAAI,mBAAJ,C;UA
Ae,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAlB,C;QACoC,kBAAvB,eAAiB,cAAjB,C;QAA+B,sBAAI,aA
AJ,C;QAA5C,arB/xSO,W;QqBgySe,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,aAAV,EAAuB,sBAAK,
KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;mGAyBA,yB;MAAA,qD;MAAA
,gE;MAAA,uC;QAkB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAlB,C;
QACoC,kBAAvB,eAAiB,cAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,arBxzSO,W;QqByzSe,qB;QAAtB,iBAAc,C
AAd,wB;UACI,gBAAc,UAAU,aAAV,EAAuB,sBAAK,KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,
OAAO,M;O;KAtBX,C;mGAyBA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAkB0B,Q;QAHtB,IAAI,mBAAJ,C;UA
Ae,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAlB,C;QACqC,kBAAxB,eAAkB,cAAlB,C;QAAgC,sBAAI,aA
AJ,C;QAA7C,arBj1SO,W;QqBk1Se,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,aAAV,EAAuB,sBAAK,
KAAL,CAAvB,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAtBX,C;iHAyBA,yB;MAAA,qD;MAAA,
gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAlB,C;
QACmC,kBAAtB,eAAgB,cAAhB,C;QAA8B,sBAAI,aAAJ,C;QAA3C,arB32SO,W;QqB42Se,qB;QAAtB,iBAAc,C
AAd,wB;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,a
AAJ,C;;QAEX,OAAO,M;O;KAvBX,C;iHA0BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,
mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAlB,C;QACoC,kBAAvB,eAAiB,cAAjB,C;QAA
+B,sBAAI,aAAJ,C;QAA5C,arBr4SO,W;QqBs4Se,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,KAAV,EA
AiB,aAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;iH
A0BA,yB;MAAA,qD;MAAA,gE;MAAA,uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBA
AkB,sBAAK,CAAL,CAAlB,C;QACoC,kBAAvB,eAAiB,cAAjB,C;QAA+B,sBAAI,aAAJ,C;QAA5C,arB/5SO,W;
QqBg6Se,qB;QAAtB,iBAAc,CAAd,wB;UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,sBAAK,KAAL,CA
A9B,C;UACd,MAAO,WAAI,aAAJ,C;;QAEX,OAAO,M;O;KAvBX,C;iHA0BA,yB;MAAA,qD;MAAA,gE;MAAA
,uC;QAmB0B,Q;QAHtB,IAAI,mBAAJ,C;UAAe,OAAO,W;QACtB,sBAAkB,sBAAK,CAAL,CAAlB,C;QACqC,k
BAAxB,eAAkB,cAAlB,C;QAAgC,sBAAI,aAAJ,C;QAA7C,arBz7SO,W;QqB07Se,qB;QAAtB,iBAAc,CAAd,wB;
UACI,gBAAc,UAAU,KAAV,EAAiB,aAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;UACd,MAAO,WAAI,aAAJ,C;;Q
AEX,OAAO,M;O;KAvBX,C;iFA0BA,yB;MAxZA,gD;MAAA,gE;MAwZA,gD;QAgBW,sB;;UAtZS,Q;UAHhB,IA

AI,mBAAJ,C;YAAe,qBAAO,OAyZH,OAzZG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UA
A+B,sBAwZzB,OAxZyB,C;UAA5C,arBzjSO,W;UqB0jSP,kBAuZmB,O;UAtZH,2B;UAAhB,OAAgB,cAAhB,C;Y
AAgB,yB;YACZ,cAqZwB,SArZV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,qB
AAO,M;;;QAkZP,yB;O;KAhBJ,C;iFAmBA,yB;MAlZA,gD;MAAA,gE;MAkZA,gD;QAgBW,sB;;UAhZS,Q;UAH
hB,IAAI,mBAAJ,C;YAAe,qBAAO,OAmZH,OAnZG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,
C;UAA+B,sBAkZzB,OAlZyB,C;UAA5C,arBllSO,W;UqBmlSP,kBAiZmB,O;UAhZH,2B;UAAhB,OAAgB,cAAh
B,C;YAAgB,yB;YACZ,cA+YwB,SA/YV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,WAAJ,C;;UA
EX,qBAAO,M;;;QA4YP,yB;O;KAhBJ,C;iFAmBA,yB;MA5YA,gD;MAAA,gE;MA4YA,gD;QAgBW,sB;;UA1YS,
Q;UAHhB,IAAI,mBAAJ,C;YAAe,qBAAO,OA6YH,OA7YG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,iBAAO,CAA
P,IAAb,C;UAA+B,sBA4YzB,OA5YyB,C;UAA5C,arB3mSO,W;UqB4mSP,kBA2YmB,O;UA1YH,2B;UAAhB,O
AAgB,cAAhB,C;YAAgB,yB;YACZ,cAyYwB,SAzYV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MAAO,WAAI,
WAAJ,C;;UAEX,qBAAO,M;;;QAsYP,yB;O;KAhBJ,C;iFAmBA,yB;MAtYA,gD;MAAA,gE;MAsYA,gD;QAgBW,
sB;;UApYS,Q;UAHhB,IAAI,mBAAJ,C;YAAe,qBAAO,OAuYH,OAvYG,C;YAAP,uB;;UACqB,kBAAvB,eAAa,i
BAAO,CAAP,IAAb,C;UAA+B,sBAsYzB,OAtYyB,C;UAA5C,arBpoSO,W;UqBqoSP,kBAqYmB,O;UApYH,2B;
UAAhB,OAAgB,cAAhB,C;YAAgB,yB;YACZ,cAmYwB,SAnYV,CAAU,WAAV,EAAuB,OAAvB,C;YACd,MA
AO,WAAI,WAAJ,C;;UAEX,qBAAO,M;;;QAgYP,yB;O;KAhBJ,C;+FAmBA,yB;MAhYA,gD;MAAA,gE;MAl7KI,
0D;MAkzLJ,gD;QAiBW,6B;;UA9XO,gC;UAHd,IAAI,mBAAJ,C;YAAe,4BAAO,OAiYI,OAjYJ,C;YAAP,8B;;UA
CqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBAgYlB,OAhYkB,C;UAA5C,arB9pSO,W;UqB+pSP,kBA
+X0B,O;UA9XZ,OAr8KE,YAAR,iBAAQ,C;UAq8KF,mB;UAAA,kB;UAAA,kB;UAAd,0D;YACI,cA6X+B,SA7
XjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4
BAAO,M;;;QA0XP,gC;O;KAjBJ,C;+FAoBA,yB;MA1XA,gD;MAAA,gE;MAp8KI,0D;MA8zLJ,gD;QAiBW,6B;;
UAxXO,gC;UAHd,IAAI,mBAAJ,C;YAAe,4BAAO,OA2XI,OA3XJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,iBAAO
,CAAP,IAAb,C;UAA+B,sBA0XlB,OA1XkB,C;UAA5C,arBxrSO,W;UqByrSP,kBAyX0B,O;UAxXZ,OAv9KE,YA
AR,iBAAQ,C;UAu9KF,mB;UAAA,kB;UAAA,kB;UAAd,0D;YACI,cAuX+B,SAvXjB,CAAU,KAAV,EAAiB,WA
AjB,EAA8B,sBAAK,KAAL,CAA9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QAoXP,gC;O;KAjBJ
,C;+FAoBA,yB;MApXA,gD;MAAA,gE;MAt9KI,0D;MA00LJ,gD;QAiBW,6B;;UAlXO,gC;UAHd,IAAI,mBAAJ,
C;YAAe,4BAAO,OAqXI,OArXJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBAoXl
B,OApXkB,C;UAA5C,arBltSO,W;UqBmtSP,kBAmX0B,O;UAlXZ,OAz+KE,YAAR,iBAAQ,C;UAy+KF,mB;UA
AA,kB;UAAA,kB;UAAd,0D;YACI,cAiX+B,SAjXjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CA
A9B,C;YACd,MAAO,WAAI,WAAJ,C;;UAEX,4BAAO,M;;;QA8WP,gC;O;KAjBJ,C;+FAoBA,yB;MA9WA,gD;M
AAA,gE;MAx+KI,0D;MAs1LJ,gD;QAiBW,6B;;UA5WO,gC;UAHd,IAAI,mBAAJ,C;YAAe,4BAAO,OA+WI,OA/
WJ,C;YAAP,8B;;UACqB,kBAAvB,eAAa,iBAAO,CAAP,IAAb,C;UAA+B,sBA8WlB,OA9WkB,C;UAA5C,arB5u
SO,W;UqB6uSP,kBA6W0B,O;UA5WZ,OA3/KE,YAAR,iBAAQ,C;UA2/KF,mB;UAAA,kB;UAAA,kB;UAAd,0D;
YACI,cA2W+B,SA3WjB,CAAU,KAAV,EAAiB,WAAjB,EAA8B,sBAAK,KAAL,CAA9B,C;YACd,MAAO,WAA
I,WAAJ,C;;UAEX,4BAAO,M;;;QAwWP,gC;O;KAjBJ,C;mFAoBA,yB;MAAA,wB;MAAA,sC;QAUoB,Q;QADhB
,UAAgB,W;QACA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MpC3mTiD,SoC2mTjD,GpC3mT2D,KAA
K,GoC2mTzD,SAAS,OAAT,CpC3mToE,KAAX,IAAf,C;;QoC6mTrD,OAAO,G;O;KAbX,C;mFAgBA,yB;MAAA
,wB;MAAA,sC;QAUoB,Q;QADhB,UAAgB,W;QACA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MpC3n
TiD,SoC2nTjD,GpC3nT2D,KAAK,GoC2nTzD,SAAS,OAAT,CpC3nToE,KAAX,IAAf,C;;QoC6nTrD,OAAO,G;O;
KAbX,C;mFAgBA,yB;MAAA,wB;MAAA,sC;QAUoB,Q;QADhB,UAAgB,W;QACA,2B;QAAhB,OAAgB,cAAh
B,C;UAAgB,yB;UACZ,MpC3oTiD,SoC2oTjD,GpC3oT2D,KAAK,GoC2oTzD,SAAS,OAAT,CpC3oToE,KAAX,I
AAf,C;;QoC6oTrD,OAAO,G;O;KAbX,C;mFAgBA,yB;MAAA,wB;MAAA,sC;QAUoB,Q;QADhB,UAAgB,W;QA
CA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MpC3pTiD,SoC2pTjD,GpC3pT2D,KAAK,GoC2pTzD,SA
AS,OAAT,CpC3pToE,KAAX,IAAf,C;;QoC6pTrD,OAAO,G;O;KAbX,C;8FAgBA,+B;MAUoB,Q;MADhB,UAAk
B,G;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;+FA
GX,+B;MAUoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,
OAAT,C;;MAEX,OAAO,G;K;+FAGX,+B;MAUoB,Q;MADhB,UAAkB,G;MACF,2B;MAAhB,OAAgB,cAAhB,C
;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;+FAGX,+B;MAUoB,Q;MADhB,UAAkB,G;MA

CF,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;kFAGX,+B;
MAYoB,Q;MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C
;;MAEX,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,
yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAoB,C;MACJ,2B;M
AAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,OAAO,G;K;mFAGX,+B;MAYoB,
Q;MADhB,UAAoB,C;MACJ,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,OAAO,SAAS,OAAT,C;;MAEX,
OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ
,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB
,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,+B;MAYoB,Q
;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,YAAO,SAAS,OAAT,CAAP,I;;MA
EJ,OAAO,G;K;mFAGX,+B;MAYoB,Q;MADhB,UAAe,C;MACC,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QA
CZ,YAAO,SAAS,OAAT,CAAP,I;;MAEJ,OAAO,G;K;mFAGX,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;Q
ADhB,Y;QACgB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO
,G;O;KAfX,C;mFAkBA,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;QADhB,Y;QACgB,2B;QAAhB,OAAgB,
cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAfX,C;mFAkBA,yB;MAAA,S
AWoB,gB;MAXpB,sC;QAYoB,Q;QADhB,Y;QACgB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAO,S
AAS,OAAT,CAAP,C;;QAEJ,OAAO,G;O;KAfX,C;mFAkBA,yB;MAAA,SAWoB,gB;MAXpB,sC;QAYoB,Q;QA
DhB,Y;QACgB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,cAAO,SAAS,OAAT,CAAP,C;;QAEJ,OAAO,
G;O;KAfX,C;mFAkBA,yB;MpCtnTA,6B;MoCsnTA,sC;QAaoB,Q;QADhB,UpCxnTmC,coCwnTnB,CpCxnTmB,
C;QoCynTnB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MpCt8TiD,coCs8TjD,GpCt8T2D,KAAK,GoCs8
TzD,SAAS,OAAT,CpCt8ToE,KAAX,IAAf,C;;QoCw8TrD,OAAO,G;O;KAhBX,C;mFAmBA,yB;MpCzoTA,6B;M
oCyoTA,sC;QAaoB,Q;QADhB,UpC3oTmC,coC2oTnB,CpC3oTmB,C;QoC4oTnB,2B;QAAhB,OAAgB,cAAhB,C;
UAAgB,yB;UACZ,MpCz9TiD,coCy9TjD,GpCz9T2D,KAAK,GoCy9TzD,SAAS,OAAT,CpCz9ToE,KAAX,IAAf,
C;;QoC29TrD,OAAO,G;O;KAhBX,C;mFAmBA,yB;MpC5pTA,6B;MoC4pTA,sC;QAaoB,Q;QADhB,UpC9pTmC,
coC8pTnB,CpC9pTmB,C;QoC+pTnB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MpC5+TiD,coC4+TjD,
GpC5+T2D,KAAK,GoC4+TzD,SAAS,OAAT,CpC5+ToE,KAAX,IAAf,C;;QoC8+TrD,OAAO,G;O;KAhBX,C;mF
AmBA,yB;MpC/qTA,6B;MoC+qTA,sC;QAaoB,Q;QADhB,UpCjrTmC,coCirTnB,CpCjrTmB,C;QoCkrTnB,2B;QA
AhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MpC//TiD,coC+/TjD,GpC//T2D,KAAK,GoC+/TzD,SAAS,OAAT,Cp
C//ToE,KAAX,IAAf,C;;QoCigUrD,OAAO,G;O;KAhBX,C;mFAmBA,yB;MnB/rTA,+B;MmB+rTA,sC;QAaoB,Q;
QADhB,UnBhsTqC,eAAW,oBmBgsT/B,CnBhsT+B,CAAX,C;QmBisTrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,
yB;UACZ,MnB/gUmD,emB+gUnD,GnB/gU8D,KAAK,KmB+gU5D,SAAS,OAAT,CnB/gUuE,KAAX,CAAhB,C;;
QmBihUvD,OAAO,G;O;KAhBX,C;mFAmBA,yB;MnBltTA,+B;MmBktTA,sC;QAaoB,Q;QADhB,UnBntTqC,eA
AW,oBmBmtT/B,CnBntT+B,CAAX,C;QmBotTrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnBliUmD
,emBkiUnD,GnBliU8D,KAAK,KmBkiU5D,SAAS,OAAT,CnBliUuE,KAAX,CAAhB,C;;QmBoiUvD,OAAO,G;O;
KAhBX,C;mFAmBA,yB;MnBruTA,+B;MmBquTA,sC;QAaoB,Q;QADhB,UnBtuTqC,eAAW,oBmBsuT/B,CnBtu
T+B,CAAX,C;QmBuuTrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnBrjUmD,emBqjUnD,GnBrjU8D,
KAAK,KmBqjU5D,SAAS,OAAT,CnBrjUuE,KAAX,CAAhB,C;;QmBujUvD,OAAO,G;O;KAhBX,C;mFAmBA,y
B;MnBxvTA,+B;MmBwvTA,sC;QAaoB,Q;QADhB,UnBzvTqC,eAAW,oBmByvT/B,CnBzvT+B,CAAX,C;QmB0
vTrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,MnBxkUmD,emBwkUnD,GnBxkU8D,KAAK,KmBwkU
5D,SAAS,OAAT,CnBxkUuE,KAAX,CAAhB,C;;QmB0kUvD,OAAO,G;O;KAhBX,C;IAmBA,kC;MA2DI,WpBv9
TO,MAAO,KoBu9TG,cpBv9TH,EoBq6TH,KAkDkB,OpBv9Tf,C;MoBw9Td,WAAW,iBAAa,IAAb,C;MACX,aA
AU,CAAV,MAAkB,IAAlB,M;QACI,IAAK,WArDqB,GAqDP,sBAAK,CAAL,CArDO,EAAnB,KAqDqB,CAAM,
CAAN,CArDF,CAqDrB,C;;MArDT,OAuDO,I;K;IApDX,kC;MAkEI,WpB1+TO,MAAO,KoB0+TG,cpB1+TH,Eo
Bi7TH,KAyDkB,OpB1+Tf,C;MoB2+Td,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,
IAAK,WA5DqB,GA4DP,sBAAK,CAAL,CA5DO,EAAnB,KA4DqB,CAAM,CAAN,CA5DF,CA4DrB,C;;MA5DT,
OA8DO,I;K;IA3DX,kC;MAyEI,WpB7/TO,MAAO,KoB6/TG,cpB7/TH,EoB67TH,KAgEkB,OpB7/Tf,C;MoB8/Td,
WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,IAAK,WAnEqB,GAmEP,sBAAK,CAAL
,CAnEO,EAAnB,KAmEqB,CAAM,CAAN,CAnEF,CAmErB,C;;MAnET,OAqEO,I;K;IAlEX,kC;MAgFI,WpBhhU

O,MAAO,KoBghUG,cpBhhUH,EoBy8TH,KAuEkB,OpBhhUf,C;MoBihUd,WAAW,iBAAa,IAAb,C;MACX,aAA
U,CAAV,MAAkB,IAAlB,M;QACI,IAAK,WA1EqB,GA0EP,sBAAK,CAAL,CA1EO,EAAnB,KA0EqB,CAAM,C
AAN,CA1EF,CA0ErB,C;;MA1ET,OA4EO,I;K;+EAzEX,yB;MAAA,gE;MpBl9TA,iB;MoBk9TA,8C;QAWI,WpBv
9TO,MAAO,KoBu9TG,cpBv9TH,EoBu9TS,KAAM,OpBv9Tf,C;QoBw9Td,WAAW,eAAa,IAAb,C;QACX,aAAU,
CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,C
AAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBr+TA,iB;MoBq+TA,8C;QAWI,WpB1+TO,M
AAO,KoB0+TG,cpB1+TH,EoB0+TS,KAAM,OpB1+Tf,C;QoB2+Td,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV
,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;
;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBx/TA,iB;MoBw/TA,8C;QAWI,WpB7/TO,MAAO,Ko
B6/TG,cpB7/TH,EoB6/TS,KAAM,OpB7/Tf,C;QoB8/Td,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IA
AlB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OA
AO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpB3gUA,iB;MoB2gUA,8C;QAWI,WpBhhUO,MAAO,KoBghUG,c
pBhhUH,EoBghUS,KAAM,OpBhhUf,C;QoBihUd,WAAW,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAlB,
M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,MAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;
O;KAhBX,C;IAmBA,kC;MA8DoB,gB;MAHhB,gBAAgB,c;MAChB,WAAW,iBpBplUJ,MAAO,KoBolUsB,wBAn
DzB,KAmDyB,EAAwB,EAAxB,CpBplUtB,EoBolUmD,SpBplUnD,CoBolUH,C;MACX,QAAQ,C;MACQ,OArD
L,KAqDK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,W
AvDqB,GAuDP,uBAAK,UAAL,EAAK,kBAAL,UAvDO,EAuDI,OAvDJ,CAuDrB,C;;MAvDT,OAyDO,I;K;IAtD
X,kC;MAuEoB,gB;MAHhB,gBAAgB,c;MAChB,WAAW,iBpBzmUJ,MAAO,KoBymUsB,wBA5DzB,KA4DyB,E
AAwB,EAAxB,CpBzmUtB,EoBymUmD,SpBzmUnD,CoBymUH,C;MACX,QAAQ,C;MACQ,OA9DL,KA8DK,W
;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAhEqB,GAgE
P,uBAAK,UAAL,EAAK,kBAAL,UAhEO,EAgEI,OAhEJ,CAgErB,C;;MAhET,OAkEO,I;K;IA/DX,kC;MAgFoB,g
B;MAHhB,gBAAgB,c;MAChB,WAAW,iBpB9nUJ,MAAO,KoB8nUsB,wBArEzB,KAqEyB,EAAwB,EAAxB,Cp
B9nUtB,EoB8nUmD,SpB9nUnD,CoB8nUH,C;MACX,QAAQ,C;MACQ,OAvEL,KAuEK,W;MAAhB,OAAgB,cA
AhB,C;QAAgB,yB;QACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAzEqB,GAyEP,uBAAK,UAAL,EA
AK,kBAAL,UAzEO,EAyEI,OAzEJ,CAyErB,C;;MAzET,OA2EO,I;K;IAxEX,kC;MAyFoB,gB;MAHhB,gBAAgB,c
;MAChB,WAAW,iBpBnpUJ,MAAO,KoBmpUsB,wBA9EzB,KA8EyB,EAAwB,EAAxB,CpBnpUtB,EoBmpUmD,
SpBnpUnD,CoBmpUH,C;MACX,QAAQ,C;MACQ,OAhFL,KAgFK,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;Q
ACZ,IAAI,KAAK,SAAT,C;UAAoB,K;QACpB,IAAK,WAlFqB,GAkFP,uBAAK,UAAL,EAAK,kBAAL,UAlFO,E
AkFI,OAlFJ,CAkFrB,C;;MAlFT,OAoFO,I;K;+EAjFX,yB;MAAA,kF;MAAA,gE;MpB9kUA,iB;MoB8kUA,8C;QA
coB,UAEY,M;QAL5B,gBAAgB,c;QAChB,WAAW,epBplUJ,MAAO,KoBolUsB,wBAAN,KAAM,EAAAwB,EAAx
B,CpBplUtB,EoBolUmD,SpBplUnD,CoBolUH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAg
B,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,uBAAK,UAAL,EAAK,kBAAL,UA
AV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KAlBX,C;+EAqBA,yB;MAAA,kF;MAAA,gE;MpBnmUA,iB;M
oBmmUA,8C;QAcoB,UAEY,M;QAL5B,gBAAgB,c;QAChB,WAAW,epBzmUJ,MAAO,KoBymUsB,wBAAN,KA
AM,EAAwB,EAAxB,CpBzmUtB,EoBymUmD,SpBzmUnD,CoBymUH,C;QACX,QAAQ,C;QACQ,uB;QAAhB,O
AAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,UAAU,uBAAK,UA
AL,EAAK,kBAAL,UAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KAlBX,C;+EAqBA,yB;MAAA,kF;MAA
A,gE;MpBxnUA,iB;MoBwnUA,8C;QAcoB,UAEY,M;QAL5B,gBAAgB,c;QAChB,WAAW,epB9nUJ,MAAO,KoB
8nUsB,wBAAN,KAAM,EAAAwB,EAAAxB,CpB9nUtB,EoB8nUmD,SpB9nUnD,CoB8nUH,C;QACX,QAAQ,C;QA
CQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UACpB,IAAK,WAAI,U
AAU,uBAAK,UAAL,EAAK,kBAAL,UAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KAlBX,C;8EAqBA,yB
;MAAA,kF;MAAA,gE;MpB7oUA,iB;MoB6oUA,8C;QAcoB,UAEY,M;QAL5B,gBAAgB,c;QAChB,WAAW,epBn
pUJ,MAAO,KoBmpUsB,wBAAN,KAAM,EAAwB,EAAxB,CpBnpUtB,EoBmpUmD,SpBnpUnD,CoBmpUH,C;Q
ACX,QAAQ,C;QACQ,uB;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAAI,KAAK,SAAT,C;YAAoB,K;UAC
pB,IAAK,WAAI,UAAU,uBAAK,UAAL,EAAK,kBAAL,UAAV,EAAqB,OAArB,CAAJ,C;;QAET,OAAO,I;O;KAl
BX,C;IAqBA,kC;MA2DI,WpBvtUO,MAAO,KoButUG,cpBvtUH,EoBqqUH,KAkDkB,KpBvtUf,C;MoBwtUd,WA
AW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,IAAK,WArDqB,GAqDP,sBAAK,CAAL,CAr

DO,EAAnB,KAqDqB,aAAM,CAAN,CArDF,CAqDrB,C;;MArDT,OAuDO,I;K;IApDX,kC;MAkEI,WpB1uUO,M
AAO,KoB0uUG,cpB1uUH,EoBirUH,KAyDkB,KpB1uUf,C;MoB2uUd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CA
AV,MAAkB,IAAlB,M;QACI,IAAK,WA5DqB,GA4DP,sBAAK,CAAL,CA5DO,EAAnB,KA4DqB,aAAM,CAAN,
CA5DF,CA4DrB,C;;MA5DT,OA8DO,I;K;IA3DX,kC;MAyEI,WpB7vUO,MAAO,KoB6vUG,cpB7vUH,EoB6rUH,
KAgEkB,KpB7vUf,C;MoB8vUd,WAAW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,IAAK,
WAnEqB,GAmEP,sBAAK,CAAL,CAnEO,EAAnB,KAmEqB,aAAM,CAAN,CAnEF,CAmErB,C;;MAnET,OAqE
O,I;K;IAlEX,kC;MAgFI,WpBhxUO,MAAO,KoBgxUG,cpBhxUH,EoBysUH,KAuEkB,KpBhxUf,C;MoBixUd,WA
AW,iBAAa,IAAb,C;MACX,aAAU,CAAV,MAAkB,IAAlB,M;QACI,IAAK,WA1EqB,GA0EP,sBAAK,CAAL,CA
1EO,EAAnB,KA0EqB,aAAM,CAAN,CA1EF,CA0ErB,C;;MA1ET,OA4EO,I;K;+EAzEX,yB;MAAA,gE;MpBltUA
,iB;MoBktUA,8C;QAWI,WpBvtUO,MAAO,KoButUG,cpBvtUH,EoButUS,KAAM,KpBvtUf,C;QoBwtUd,WAAW
,eAAa,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAA
mB,kBAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBruUA,iB;MoB
quUA,8C;QAWI,WpB1uUO,MAAO,KoB0uUG,cpB1uUH,EoB0uUS,KAAM,KpB1uUf,C;QoB2uUd,WAAW,eAA
a,IAAb,C;QACX,aAAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,k
BAAM,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpBxvUA,iB;MoBwvU
A,8C;QAWI,WpB7vUO,MAAO,KoB6vUG,cpB7vUH,EoB6vUS,KAAM,KpB7vUf,C;QoB8vUd,WAAW,eAAa,IA
Ab,C;QACX,aAAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,kBAA
M,CAAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;+EAmBA,yB;MAAA,gE;MpB3wUA,iB;MoB2wUA,8C
;QAWI,WpBhxUO,MAAO,KoBgxUG,cpBhxUH,EoBgxUS,KAAM,KpBhxUf,C;QoBixUd,WAAW,eAAa,IAAb,C;
QACX,aAAU,CAAV,MAAkB,IAAlB,M;UACI,IAAK,WAAI,UAAU,sBAAK,CAAL,CAAV,EAAmB,kBAAM,C
AAN,CAAnB,CAAJ,C;;QAET,OAAO,I;O;KAhBX,C;IAmBA,2B;MAQoB,Q;MADhB,UAAgB,W;MAChB,wBA
AgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,MpCr9UiD,SoCq9UjD,GpCr9U2D,KAAK,GoCq9UzD,OpCr9
UoE,KAAX,IAAf,C;;MoCu9UrD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAiB,2B;MACjB,wBAAgB,SAAh
B,gB;QAAgB,cAAA,SAAhB,M;QACI,MnBh+UmD,UmBg+UnD,GnBh+U8D,KAAK,KmBg+U5D,OnBh+UuE,K
AAX,CAAhB,C;;MmBk+UvD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB,W;MAChB,wBAAgB,SAAhB,
gB;QAAgB,cAAA,SAAhB,M;QACI,MpCj/UiD,SoCi/UjD,GpCj/U2D,KAAK,GAAW,CDqP5C,SqC4vUxB,OrC5v
UkC,KAAL,GAAiB,GAAtB,CCrP4C,MAAX,IAAf,C;;MoCm/UrD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,U
AAgB,W;MAChB,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QACI,MpC//UiD,SoC+/UjD,GpC//U2D,KAA
K,GAAW,CCsP5C,SmCywUxB,OnCzwUkC,KAAL,GAAiB,KAAtB,CDtP4C,MAAX,IAAf,C;;MoCigVrD,OAAO
,G;K;+EAGX,yB;MAAA,0C;MpClsUA,6B;MoCksUA,4B;QAOI,OpC/rUmC,coC+rUpB,IAAR,iBAAQ,CpC/rUoB
,C;O;KoCwrUvC,C;+EAUA,yB;MAAA,0C;MnB7rUA,+B;MmB6rUA,4B;QAOI,OnB1rUsC,emB0rUvB,IAAR,iB
AAQ,CnB1rUuB,C;O;KmBmrU1C,C;+EAUA,yB;MAAA,sC;MpCttUA,6B;MoCstUA,iBAOiB,yB;QrCnzUb,6B;e
qCmzUa,c;UAAE,OrC1yUoB,cqC0yUpB,ErC1yU8B,KAAL,GAAiB,GAAtB,C;S;OqC0yUtB,C;MAPjB,4B;QA7i
BoB,Q;QADhB,UpC9pTmC,coC8pTnB,CpC9pTmB,C;QoC+pTnB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;U
ACZ,MpC5+TiD,coC4+TjD,GpC5+T2D,KAAK,GAAW,CDqP5C,cqCuvTf,OrCvvTyB,KAAL,GAAiB,GAAtB,C
CrP4C,MAAX,IAAf,C;;QoC+hVrD,OAjjBO,G;O;KA0iBX,C;+EAUA,yB;MAAA,sC;MpChuUA,6B;MoCguUA,i
BAOiB,yB;QnC5zUb,6B;emC4zUa,c;UAAE,OnCnzUoB,cmCmzUpB,EnCnzU8B,KAAL,GAAiB,KAAtB,C;S;Om
CmzUtB,C;MAPjB,4B;QApiBoB,Q;QADhB,UpCjrTmC,coCirTnB,CpCjrTmB,C;QoCkrTnB,2B;QAAhB,OAAgB,
cAAhB,C;UAAgB,yB;UACZ,MpC//TiD,coC+/TjD,GpC//T2D,KAAK,GAAW,CCsP5C,cmCywTf,OnCzwTyB,KA
AL,GAAiB,KAAtB,CDtP4C,MAAX,IAAf,C;;QoCyiVrD,OAxiBO,G;O;KAiiBX,C;IC/lVA,mC;MAQoB,UACL,M
;MAHX,aAAa,gBAAW,cAAX,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,oBA
AO,cAAP,EAAO,sBAAP,WAAkB,OAAlB,C;;MACJ,OAAO,M;K;IAGX,kC;MAQoB,UACL,M;MAHX,aAAa,eA
AU,cAAV,C;MACb,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,oBAAO,cAAP,EAAO,s
BAAP,WAAkB,OAAlB,C;;MACJ,OAAO,M;K;IAGX,mC;MAQoB,UACL,M;MAHX,aAAa,gBAAW,cAAX,C;M
ACb,YAAY,C;MACI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,WAAkB,
OAAlB,C;;MACJ,OAAO,M;K;IAGX,oC;MAQoB,UACL,M;MAHX,aAAa,iBAAY,cAAZ,C;MACb,YAAY,C;MA
CI,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,oBAAO,cAAP,EAAO,sBAAP,WAAkB,OAAlB,C;;MACJ,
OAAO,M;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB,W;MACA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAC

Z,MrCAiD,SqCAjD,GrCA2D,KAAK,GqCAzD,OrCAoE,KAAX,IAAf,C;;MqCErD,OAAO,G;K;IAGX,2B;MAQo
B,Q;MADhB,UAAiB,2B;MACD,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,MpBXmD,UoBWnD,GpBX8
D,KAAK,KoBW5D,OpBXuE,KAAX,CAAhB,C;;MoBavD,OAAO,G;K;IAGX,2B;MAQoB,Q;MADhB,UAAgB,W
;MACA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,MrC5BiD,SqC4BjD,GrC5B2D,KAAK,GAAW,CDqP
5C,SsCzNxB,OtCyNkC,KAAL,GAAiB,GAAtB,CCrP4C,MAAX,IAAf,C;;MqC8BrD,OAAO,G;K;IAGX,2B;MAQ
oB,Q;MADhB,UAAgB,W;MACA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,MrC1CiD,SqC0CjD,GrC1C
2D,KAAK,GAAW,CCsP5C,SoC5MxB,OpC4MkC,KAAL,GAAiB,KAAtB,CDtP4C,MAAX,IAAf,C;;MqC4CrD,O
AAO,G;K;IC3GX,wB;MAMI,OtCuCkE,YsCvCvD,CtCuCwE,KAAjB,EsCvClD,CtCuC+E,KAA7B,CsCvCvD,KA
AJ,GAAY,CAAZ,GAAmB,C;K;IAG9B,wB;MAMI,OrBsCmE,aqBtCxD,CrBsC0E,KAAlB,EqBtCnD,CrBsCiF,KA
A9B,CqBtCxD,KAAJ,GAAY,CAAZ,GAAmB,C;K;IAG9B,wB;MAMI,OvCKgF,0BuCLrE,CvC0P2B,KAAL,GAA
iB,GArP8B,EuCLhE,CvC0PsB,KAAL,GAAiB,GArP8B,CuCLrE,KAAJ,GAAY,CAAZ,GAAmB,C;K;IAG9B,wB;
MAMI,OrCIiF,0BqCJtE,CrCkP2B,KAAL,GAAiB,KA9O+B,EqCJjE,CrCkPsB,KAAL,GAAiB,KA9O+B,CqCJtE,K
AAJ,GAAY,CAAZ,GAAmB,C;K;mFAG9B,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAA
M,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,E
AAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,OAAO,MAA
M,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;QAOI,O
AAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;IAUA,4B;MAOc,Q;MADV,UAA
U,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,O
AAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,
SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,
OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;M
AOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,
CAAX,C;;MACvB,OAAO,G;K;IAGX,wB;MAMI,OtCjFkE,YsCiFvD,CtCjFwE,KAAjB,EsCiFlD,CtCjF+E,KAA7
B,CsCiFvD,KAAJ,GAAY,CAAZ,GAAmB,C;K;IAG9B,wB;MAMI,OrBlFmE,aqBkFxD,CrBlF0E,KAAlB,EqBkFn
D,CrBlFiF,KAA9B,CqBkFxD,KAAJ,GAAY,CAAZ,GAAmB,C;K;IAG9B,wB;MAMI,OvCnHgF,0BuCmHrE,CvC
kI2B,KAAL,GAAiB,GArP8B,EuCmHhE,CvCkIsB,KAAL,GAAiB,GArP8B,CuCmHrE,KAAJ,GAAY,CAAZ,GAA
mB,C;K;IAG9B,wB;MAMI,OrCpHiF,0BqCoHtE,CrC0H2B,KAAL,GAAiB,KA9O+B,EqCoHjE,CrC0HsB,KAAL,
GAAiB,KA9O+B,CqCoHtE,KAAJ,GAAY,CAAZ,GAAmB,C;K;mFAG9B,yB;MAAA,8C;MAAA,0B;QAOI,OAA
O,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;MAAA,0B;
QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;MAAA,8C;
MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;mFAUA,yB;
MAAA,8C;MAAA,0B;QAOI,OAAO,MAAM,CAAN,EAAS,MAAM,CAAN,EAAS,CAAT,CAAT,C;O;KAPX,C;I
AUA,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GA
AN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAA
V,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MAD
V,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,MAAM,SAAM,GAAN,EAAW,CAAX,C;;M
ACvB,OAAO,G;K;IAGX,4B;MAOc,Q;MADV,UAAU,C;MACA,uB;MAAV,OAAU,cAAV,C;QAAU,mB;QAAO,
MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IC7OX,6B;MAOI,IAAI,mBAAJ,C;QACI,MAAM,
2BAAuB,iBAAc,SAAd,eAAvB,C;MACV,OAAO,SAAK,M;K;IAGhB,6B;MAOI,IAAI,mBAAJ,C;QACI,MAAM,2
BAAuB,iBAAc,SAAd,eAAvB,C;MACV,OAAO,SAAK,M;K;IAGhB,mC;MAKI,OAAW,mBAAJ,GAAe,IAAf,GA
AyB,SAAK,M;K;IAGzC,mC;MAKI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,SAAK,M;K;IAGzC,4B;MASI,IAAI,m
BAAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAd,eAAvB,C;MACV,OAAO,SAAK,K;K;IAGhB,4B;MASI,IAAI,mB
AAJ,C;QACI,MAAM,2BAAuB,iBAAc,SAAd,eAAvB,C;MACV,OAAO,SAAK,K;K;IAGhB,kC;MAOI,OAAW,m
BAAJ,GAAe,IAAf,GAAyB,SAAK,K;K;IAGzC,kC;MAOI,OAAW,mBAAJ,GAAe,IAAf,GAAyB,SAAK,K;K;gFA
GzC,yB;MAAA,mC;MAAA,2C;MAAA,4B;QASI,OAAO,kBAAO,cAAP,C;O;KATX,C;gFAYA,yB;MAAA,mC;
MAAA,2C;MAAA,4B;QASI,OAAO,kBAAO,cAAP,C;O;KATX,C;IAYA,sC;;QASQ,OAAc,WAAP,MAAO,EAAS
,SAAT,C;;QAChB,+C;UACE,MAAM,2BAAuB,CAAE,QAAzB,C;;UAHV,O;;K;IAOJ,sC;;QASQ,OAAc,YAAP,M
AAO,EAAU,SAAV,C;;QAChB,+C;UACE,MAAM,2BAAuB,CAAE,QAAzB,C;;UAHV,O;;K;4FAOJ,yB;MAAA,

mC;MAAA,uD;MAAA,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;4FAUA,yB;MAAA,mC;MAAA,uD;MAA
A,4B;QAOI,OAAO,wBAAa,cAAb,C;O;KAPX,C;IAUA,4C;MAMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAc
,WAAP,MAAO,EAAS,SAAT,C;K;IAGlB,4C;MAMI,IAAI,mBAAJ,C;QACI,OAAO,I;MACX,OAAc,YAAP,MAA
O,EAAU,SAAV,C;K;oFAGlB,8B;MASI,OAAO,WAAW,IAAX,IAAmB,2BAAS,OAAT,C;K;oFAG9B,8B;MASI,
OAAO,WAAW,IAAX,IAAmB,2BAAS,OAAT,C;K;IAG9B,uC;MAMI,OAAO,2BxC8F4B,SwC9FnB,KxC8F6B,K
AAL,GAAiB,GAAtB,CwC9F5B,C;K;IAGX,uC;MAMI,OAAO,2BxC+F8B,UAAW,oBwC/FhC,KxC+F2B,KAAK,
CAAL,UAAN,CwC/F9B,C;K;IAGX,uC;MAMI,OAAO,2BvC0G8B,UAAW,oBuC1GhC,KvC0G2B,KAAK,CAAL,
iBAAN,CuC1G9B,C;K;IAGX,uC;MAMY,Q;MAAD,cAAC,OtBO4C,UsBP5C,KtBOkD,yBsBPxC,EtBOwC,CAA
N,CsBP7C,wBAA8B,2BAA9B,Q;MAAA,W;QAAqC,oCvCsKR,SuCtKiB,KtB+FlB,KjBuEW,QAAV,CuCtKQ,C;;
MAA5C,a;K;IAGJ,uC;MAMI,OAAO,2BtC2D4B,SsC3DnB,KtC2D6B,KAAL,GAAiB,KAAtB,CsC3D5B,C;K;IAG
X,uC;MAMI,OAAO,2BtC4D8B,UAAW,oBsC5DhC,KtC4D2B,KAAK,CAAL,YAAN,CsC5D9B,C;K;IAGX,kC;M
ASI,OAAO,uCAAgB,yBxCqCY,SwCrCI,SxCqCM,KAAL,GAAiB,GAAtB,CwCrCZ,ExCqCY,SwCrCmB,ExCqC
T,KAAL,GAAiB,GAAtB,CwCrCZ,EAA4C,EAA5C,C;K;IAG3B,kC;MASI,OAAO,uCAAgB,yBAAgB,SAAhB,EA
AsB,EAAtB,EAA0B,EAA1B,C;K;IAG3B,kC;MASI,OAAO,wCAAiB,yBAAgB,SAAhB,EAAsB,EAAtB,M;K;IAG
5B,kC;MASI,OAAO,uCAAgB,yBtCEY,SsCFI,StCEM,KAAL,GAAiB,KAAtB,CsCFZ,EtCEY,SsCFmB,EtCET,K
AAL,GAAiB,KAAtB,CsCFZ,EAA4C,EAA5C,C;K;IAG3B,gC;MAMI,OAAO,uCAAgB,yBAAgB,cAAhB,EAAsB,
eAAtB,EAA6B,CAAC,cAAD,IAA7B,C;K;IAG3B,gC;MAMI,OAAO,wCAAiB,yBAAgB,cAAhB,EAAsB,eAAtB,
EAA8B,cAAD,aAA7B,C;K;IAG5B,iC;MAMI,oBAAoB,OAAO,CAA3B,EAA8B,IAA9B,C;MACA,OAAO,uCAA
gB,yBAAgB,eAAhB,EAAuB,cAAvB,EAAiC,SAAK,KAAL,GAAY,CAAhB,GAAmB,IAAnB,GAA6B,CAAC,IA
AD,IAA1D,C;K;IAG3B,iC;MAMI,oBAAoB,kBAAO,CAA3B,EAA8B,IAA9B,C;MACA,OAAO,wCAAiB,yBAAg
B,eAAhB,EAAuB,cAAvB,EAAiC,SAAK,KAAL,cAAY,CAAhB,GAAmB,IAAnB,GAA8B,IAAD,aAA1D,C;K;IA
G5B,iC;MAQI,IxCvUgF,0BwCuU5E,ExClFkC,KAAL,GAAiB,GArP8B,EwCuUtE,6BAAM,UxClFsB,KAAL,GA
AiB,GArP8B,CwCuU5E,KAAJ,C;QAA2B,OAAO,iCAAU,M;MAChC,WxCjDuB,SwCiD5B,SxCjDsC,KAAL,GA
AiB,GAAtB,C;MwCiDV,YAAK,W;MAA9B,OvCzI6D,oBAhJP,SAAU,CDwO7B,SwCiDV,ExCjDoB,KAAL,GA
AiB,GAAtB,CCxO6B,MAAK,GDAK,KCAO,KAAZ,IAAf,CAgJO,C;K;IuC4IjE,iC;MAQI,IvCnUkE,YuCmU9D,E
vCnU+E,KAAjB,EuCmUxD,4BAAK,UvCnUgF,KAA7B,CuCmU9D,KAAJ,C;QAA0B,OAAO,iCAAU,M;MAC3C
,OvCrJ6D,cuCqJtD,SvCrJsD,EAhJP,SuCqStC,EvCrSgD,KAAK,GAAY,CuCqS5D,WvCrS4D,MAAZ,IAAf,CAgJO
,C;K;IuCwJjE,iC;MAQI,ItBvUmE,asBuU/D,EtBvUiF,KAAlB,EsBuUzD,6BAAM,UtBvUiF,KAA9B,CsBuU/D,KA
AJ,C;QAA2B,OAAO,kCAAW,M;MAC7C,OtBjK+D,iBsBiKxD,StBjKwD,EA7IP,UsB8SxC,EtB9SmD,KAAK,UA
AY,CjBmQ/C,UAAW,oBAAL,CuC2CtB,WvC3CsB,MAAK,CAAL,iBAAN,CiBnQ+C,MAAZ,CAAhB,CA6IO,C;
K;IsBoKnE,iC;MAQI,ItCnWiF,0BsCmW7E,EtCrHkC,KAAL,GAAiB,KA9O+B,EsCmWvE,8BAAO,UtCrHqB,KA
AL,GAAiB,KA9O+B,CsCmW7E,KAAJ,C;QAA4B,OAAO,iCAAU,M;MACjC,WtCpFuB,SsCoF5B,StCpFsC,KA
AL,GAAiB,KAAtB,C;MsCoFV,YAAK,W;MAA9B,OvC7K6D,oBAhJP,SAAU,CCyO7B,SsCoFV,EtCpFoB,KAA
L,GAAiB,KAAtB,CDzO6B,MAAK,GCAK,KDAO,KAAZ,IAAf,CAgJO,C;K;IuCgLjE,kD;MAUI,OvCzWkE,YuC
yWvD,SvCzWwE,KAAjB,EuCyWhD,YvCzW6E,KAA7B,CuCyWvD,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,
kD;MAUI,OtB9WmE,asB8WxD,StB9W0E,KAAlB,EsB8WjD,YtB9W+E,KAA9B,CsB8WxD,IAAJ,GAAyB,YAA
zB,GAA2C,S;K;IAGtD,kD;MAUI,OxCnZgF,0BwCmZrE,SxC9J2B,KAAL,GAAiB,GArP8B,EwCmZ9D,YxC9JoB
,KAAL,GAAiB,GArP8B,CwCmZrE,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,kD;MAUI,OtCxZiF,0BsCwZtE,S
tC1K2B,KAAL,GAAiB,KA9O+B,EsCwZ/D,YtC1KoB,KAAL,GAAiB,KA9O+B,CsCwZtE,IAAJ,GAAyB,YAAzB
,GAA2C,S;K;IAGtD,iD;MAUI,OvC7ZkE,YuC6ZvD,SvC7ZwE,KAAjB,EuC6ZhD,YvC7Z6E,KAA7B,CuC6ZvD,I
AAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAUI,OtBlamE,asBkaxD,StBla0E,KAAlB,EsBkajD,YtBla+E,KAA
9B,CsBkaxD,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD;MAUI,OxCvcgF,0BwCucrE,SxClN2B,KAAL,GAAi
B,GArP8B,EwCuc9D,YxClNoB,KAAL,GAAiB,GArP8B,CwCucrE,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,iD
;MAUI,OtC5ciF,0BsC4ctE,StC9N2B,KAAL,GAAiB,KA9O+B,EsC4c/D,YtC9NoB,KAAL,GAAiB,KA9O+B,CsC4
ctE,IAAJ,GAAyB,YAAzB,GAA2C,S;K;IAGtD,4D;MAUI,IvCjdkE,YuCid9D,YvCjd+E,KAAjB,EuCid/C,YvCjd4E
,KAA7B,CuCid9D,IAAJ,C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,IvC
ldkE,YuCkd9D,SvCld+E,KAAjB,EuCkdvD,YvCldoF,KAA7B,CuCkd9D,IAAJ,C;QAAyB,OAAO,Y;MAChC,IvCn
dkE,YuCmd9D,SvCnd+E,KAAjB,EuCmdvD,YvCndoF,KAA7B,CuCmd9D,IAAJ,C;QAAyB,OAAO,Y;MAChC,O

AAO,S;K;IAGX,4D;MAUI,ItBzdmE,asByd/D,YtBzdiF,KAAlB,EsBydhD,YtBzd8E,KAA9B,CsByd/D,IAAJ,C;QA
AiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,ItB1dmE,asB0d/D,StB1diF,KAAl
B,EsB0dxD,YtB1dsF,KAA9B,CsB0d/D,IAAJ,C;QAAyB,OAAO,Y;MAChC,ItB3dmE,asB2d/D,StB3diF,KAAlB,E
sB2dxD,YtB3dsF,KAA9B,CsB2d/D,IAAJ,C;QAAyB,OAAO,Y;MAChC,OAAO,S;K;IAGX,4D;MAUI,IxCjgBgF,0
BwCigB5E,YxC5QkC,KAAL,GAAiB,GArP8B,EwCigB7D,YxC5QmB,KAAL,GAAiB,GArP8B,CwCigB5E,IAAJ,
C;QAAiC,MAAM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,IxClgBgF,0BwCkgB5E,SxC
7QkC,KAAL,GAAiB,GArP8B,EwCkgBrE,YxC7Q2B,KAAL,GAAiB,GArP8B,CwCkgB5E,IAAJ,C;QAAyB,OAA
O,Y;MAChC,IxCngBgF,0BwCmgB5E,SxC9QkC,KAAL,GAAiB,GArP8B,EwCmgBrE,YxC9Q2B,KAAL,GAAiB,
GArP8B,CwCmgB5E,IAAJ,C;QAAyB,OAAO,Y;MAChC,OAAO,S;K;IAGX,4D;MAUI,ItCzgBiF,0BsCygB7E,Yt
C3RkC,KAAL,GAAiB,KA9O+B,EsCygB9D,YtC3RmB,KAAL,GAAiB,KA9O+B,CsCygB7E,IAAJ,C;QAAiC,MA
AM,gCAAyB,oDAAiD,YAAjD,8BAAoF,YAApF,MAAzB,C;MACvC,ItC1gBiF,0BsC0gB7E,StC5RkC,KAAL,GA
AiB,KA9O+B,EsC0gBtE,YtC5R2B,KAAL,GAAiB,KA9O+B,CsC0gB7E,IAAJ,C;QAAyB,OAAO,Y;MAChC,ItC3
gBiF,0BsC2gB7E,StC7RkC,KAAL,GAAiB,KA9O+B,EsC2gBtE,YtC7R2B,KAAL,GAAiB,KA9O+B,CsC2gB7E,I
AAJ,C;QAAyB,OAAO,Y;MAChC,OAAO,S;K;IAGX,uC;MAcW,Q;MAJP,IAAI,8CAAJ,C;QACI,OAAY,WAAL,
SAAK,EAAe,KAAf,C;;MAEhB,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,4CAAyC,KAAzC,MAAzB,C;
MAEvB,IvCthB8D,YuCshB9D,SvCthB+E,KAAjB,EuCshBvD,KAAM,MvCthB8E,KAA7B,CuCshB9D,K;QAA4B
,OAAN,KAAM,M;;QAC5B,IvCvhB8D,YuCuhB9D,SvCvhB+E,KAAjB,EuCuhBvD,KAAM,avCvhB8E,KAA7B,C
uCuhB9D,K;UAAmC,OAAN,KAAM,a;;UAC3B,gB;;MAHZ,W;K;IAOJ,uC;MAcW,Q;MAJP,IAAI,8CAAJ,C;QA
CI,OAAY,WAAL,SAAK,EAAgB,KAAhB,C;;MAEhB,IAAI,KAAM,UAAV,C;QAAqB,MAAM,gCAAyB,4CAAy
C,KAAzC,MAAzB,C;MAEvB,ItBniB+D,asBmiB/D,StBniBiF,KAAlB,EsBmiBxD,KAAM,MtBniBgF,KAA9B,Cs
BmiB/D,K;QAA4B,OAAN,KAAM,M;;QAC5B,ItBpiB+D,asBoiB/D,StBpiBiF,KAAlB,EsBoiBxD,KAAM,atBpiBg
F,KAA9B,CsBoiB/D,K;UAAmC,OAAN,KAAM,a;;UAC3B,gB;;MAHZ,W;K;ICvlBJ,2B;MAUoB,Q;MADhB,UA
AgB,W;MACA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,MxCoDiD,SwCpDjD,GxCoD2D,KAAK,GwC
pDzD,OxCoDoE,KAAX,IAAf,C;;MwClDrD,OAAO,G;K;IAGX,2B;MAUoB,Q;MADhB,UAAiB,2B;MACD,2B;M
AAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,MvBuCmD,UuBvCnD,GvBuC8D,KAAK,KuBvC5D,OvBuCuE,KA
AX,CAAhB,C;;MuBrCvD,OAAO,G;K;IAGX,2B;MAUoB,Q;MADhB,UAAgB,W;MACA,2B;MAAhB,OAAgB,c
AAhB,C;QAAgB,yB;QACZ,MxCoBiD,SwCpBjD,GxCoB2D,KAAK,GAAW,CDqP5C,SyCzQxB,OzCyQkC,KAA
L,GAAiB,GAAtB,CCrP4C,MAAX,IAAf,C;;MwClBrD,OAAO,G;K;IAGX,2B;MAUoB,Q;MADhB,UAAgB,W;M
ACA,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACZ,MxCIiD,SwCJjD,GxCI2D,KAAK,GAAW,CCsP5C,SuC1
PxB,OvC0PkC,KAAL,GAAiB,KAAtB,CDtP4C,MAAX,IAAf,C;;MwCFrD,OAAO,G;K;;;;;ICqCP,iD;MAAA,qE;
MAAgB,4B;MANpB,uC;MAMI,Y;K;IACA,4D;MAAA,qE;MAAgC,wBAAM,OAAN,Q;MAPpC,uC;MAOI,Y;K;I
ACA,mE;MAAA,qE;MAAmD,6BAAM,OAAN,EAAe,KAAf,C;MARvD,uC;MAQI,Y;K;IACA,0D;MAAA,qE;MA
AiC,wBAAM,KAAN,Q;MATrC,uC;MASI,Y;K;ICtGJ,gC;K;;;;ICuBoC,wC;8BAAsC,O;;K;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;y
CC0RtE,6B;MASI,MAAM,yB;K;;;;;;;0CAyDV,sB;MASI,OAAO,I;K;;;;;;;;;;;;;;;;;;;;;IC1Xf,gB;MAAA,oB;K;8BAII,
Y;MAA0B,oB;K;;;IAJ9B,4B;MAAA,2B;QAAA,U;;MAAA,oB;K;ICEA,yC;MAAA,e;MAAA,iB;MAAA,uB;K;IA
AA,uC;MAAA,0C;O;MAII,kE;MAEA,wF;MAEA,oF;MAEA,wE;MAEA,kE;MAEA,oF;MAEA,sF;MAEA,8E;MA
EA,wE;MAEA,sF;MAEA,uF;MAEA,iE;MAEA,6E;MAEA,iE;MAEA,2E;K;;IA5BA,8C;MAAA,6B;MAAA,sC;K;;
IAEA,yD;MAAA,6B;MAAA,iD;K;;IAEA,uD;MAAA,6B;MAAA,+C;K;;IAEA,iD;MAAA,6B;MAAA,yC;K;;IAE
A,8C;MAAA,6B;MAAA,sC;K;;IAEA,uD;MAAA,6B;MAAA,+C;K;;IAEA,wD;MAAA,6B;MAAA,gD;K;;IAEA,o
D;MAAA,6B;MAAA,4C;K;;IAEA,iD;MAAA,6B;MAAA,yC;K;;IAEA,wD;MAAA,6B;MAAA,gD;K;;IAEA,wD;
MAAA,6B;MAAA,gD;K;;IAEA,6C;MAAA,6B;MAAA,qC;K;;IAEA,mD;MAAA,6B;MAAA,2C;K;;IAEA,6C;MA
AA,6B;MAAA,qC;K;;IAEA,kD;MAAA,6B;MAAA,0C;K;;IAhCJ,mC;MAAA,+oB;K;;IAAA,wC;MAAA,a;aAAA,
O;UAAA,2C;aAAA,kB;UAAA,sD;aAAA,gB;UAAA,oD;aAAA,U;UAAA,8C;aAAA,O;UAAA,2C;aAAA,gB;UAA
A,oD;aAAA,iB;UAAA,qD;aAAA,a;UAAA,iD;aAAA,U;UAAA,8C;aAAA,iB;UAAA,qD;aAAA,iB;UAAA,qD;aA
AA,M;UAAA,0C;aAAA,Y;UAAA,gD;aAAA,M;UAAA,0C;aAAA,W;UAAA,+C;;UAAA,uE;;K;;IAqCA,4C;MAA
A,e;MAAA,iB;MAAA,uB;K;IAAA,0C;MAAA,6C;O;MAMI,0E;MAEA,0E;MAEA,4E;K;;IAJA,kD;MAAA,gC;M
AAA,0C;K;;IAEA,kD;MAAA,gC;MAAA,0C;K;;IAEA,mD;MAAA,gC;MAAA,2C;K;;IAVJ,sC;MAAA,sI;K;;IAA
A,2C;MAAA,a;aAAA,Q;UAAA,+C;aAAA,Q;UAAA,+C;aAAA,S;UAAA,gD;;UAAA,0E;;K;;IAwB8B,gC;MAAC,

oC;K;;IAQE,0B;MAAC,qB;QAAA,iD;MAAA,kB;K;;IAElC,sB;K;;IAMA,4B;K;;ICxFQ,kD;MAAA,8B;MACI,aA
AY,C;K;oDACZ,Y;MAAyB,oBAAQ,gBAAI,O;K;iDACrC,Y;MAAgD,Q;MAA1B,IAAI,aAAQ,gBAAI,OAAhB,C;
QAAA,OAAsB,iBAAI,iBAAJ,EAAI,yBAAJ,O;;QAAkB,MAAM,2BAAyB,UAAF,WAAvB,C;K;;IAPhF,oC;MAEI
,IAD8D,IAC9D,S;QACI,UAA0B,K;QAF0B,2C;;QAAA,QAAM,IAAN,C;eASxD,c;YATwD,OAStC,qBAAqB,KA
ArB,C;eAClB,W;YAVwD,OAUzC,kBAAkB,KAAlB,C;eACf,Y;YAXwD,OAWxC,mBAAmB,KAAnB,C;eAChB,
W;YAZwD,OAYzC,kBAAkB,KAAlB,C;eACf,U;YAbwD,OAa1C,iBAAiB,KAAjB,C;eACd,W;YAdwD,OAczC,k
BAAkB,KAAlB,C;eACf,Y;YAfwD,OAexC,mBAAmB,KAAnB,C;eAChB,a;YAhBwD,OAgBvC,oBAAoB,KAAp
B,C;;YACT,MAAM,6BAAsB,2DAA+C,IAA/C,CAAtB,C;;K;IAIuC,2D;MAAA,kC;MAAS,0B;MAC9D,aAAY,C;
K;2DACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;+DACvC,Y;MAA2D,Q;MAA9B,IAAI,aAAQ,kBAAM,OAAlB,C;Q
AAA,OAAwB,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAyB,UAAF,WAAvB,C;K;;IAJnF,qC;
MACyD,oD;K;IAON,wD;MAAA,kC;MAAS,uB;MACxD,aAAY,C;K;wDACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;
yDACvC,Y;MAAwD,Q;MAA9B,IAAI,aAAQ,kBAAM,OAAlB,C;QAAA,OAAwB,mBAAM,iBAAN,EAAM,yBA
AN,O;;QAAoB,MAAM,2BAAyB,UAAF,WAAvB,C;K;;IAJhF,kC;MACmD,iD;K;IAOE,yD;MAAA,kC;MAAS,w
B;MAC1D,aAAY,C;K;yDACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;2DACvC,Y;MAAyD,Q;MAA9B,IAAI,aAAQ,k
BAAM,OAAlB,C;QAAA,OAAwB,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAyB,UAAF,WAA
vB,C;K;;IAJjF,mC;MACqD,kD;K;IAOF,wD;MAAA,kC;MAAS,uB;MACxD,aAAY,C;K;wDACZ,Y;MAAyB,oBA
AQ,kBAAM,O;K;yDACvC,Y;MAAwD,Q;MAA9B,IAAI,aAAQ,kBAAM,OAAlB,C;QAAA,OAAwB,mBAAM,iB
AAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAyB,UAAF,WAAvB,C;K;;IAJhF,kC;MACmD,iD;K;IAOF,uD;MA
AA,kC;MAAS,sB;MACtD,aAAY,C;K;uDACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;uDACvC,Y;MAAuD,Q;MAA9
B,IAAI,aAAQ,kBAAM,OAAlB,C;QAAA,OAAwB,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAy
B,UAAF,WAAvB,C;K;;IAJ/E,iC;MACiD,gD;K;IAOI,yD;MAAA,kC;MAAS,wB;MAC1D,aAAY,C;K;yDACZ,Y;
MAAyB,oBAAQ,kBAAM,O;K;2DACvC,Y;MAAyD,Q;MAA9B,IAAI,aAAQ,kBAAM,OAAlB,C;QAAA,OAAwB
,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAyB,UAAF,WAAvB,C;K;;IAJjF,mC;MACqD,kD;K;I
AOE,0D;MAAA,kC;MAAS,yB;MAC5D,aAAY,C;K;0DACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;6DACvC,Y;MAA
0D,Q;MAA9B,IAAI,aAAQ,kBAAM,OAAlB,C;QAAA,OAAwB,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,M
AAM,2BAAyB,UAAF,WAAvB,C;K;;IAJlF,oC;MACuD,mD;K;IAOJ,wD;MAAA,kC;MAAS,uB;MACxD,aAAY,
C;K;wDACZ,Y;MAAyB,oBAAQ,kBAAM,O;K;yDACvC,Y;MAAwD,Q;MAA9B,IAAI,aAAQ,kBAAM,OAAlB,C
;QAAA,OAAwB,mBAAM,iBAAN,EAAM,yBAAN,O;;QAAoB,MAAM,2BAAyB,UAAF,WAAvB,C;K;;IAJhF,kC
;MACmD,iD;K;IAOpB,gC;MAAC,wB;K;;IAEhC,+B;MAC8C,MAAM,mC;K;IAEpD,8C;MAEI,IAAI,qBAAJ,C;Q
ACI,OAAO,CtBkKiF,WsBlKrE,UtBkKqE,EsBlKzD,QtBkKyD,C;;QsBhKxF,OAAS,CAAY,qBAAsB,UAAtB,EA
AkC,QAAlC,C;;K;IAI7B,2C;MAEI,IAAI,KAAY,kBAAhB,C;QAGI,KAAY,mBAAkB,QAAlB,C;;QAEH,QAAT,S
AA+C,CAAlB,IAAO,KAAP,EAAkB,O;;K;IAIvD,sC;MAGwB,Q;MADpB,gBAAgB,IAAO,KAAP,E;MACI,IAAI,
OCnGkB,ODmGT,OAAT,EAAqB,WAArB,CAAJ,C;QAChB,OAAI,aAAJ,GAAmB,KAAM,WAAzB,GAAyC,I;;Q
AEzC,c;;MAHJ,wB;MAKA,kBAAkB,K;MAClB,iBAAiB,W;MACjB,OAAO,S;K;IAIa,sB;MAAC,U;K;iCACrB,iB
;MACI,OAAO,mCAAsB,WAAK,KAAM,E;K;mCAG5C,Y;MACI,OAAO,M;K;mCAGX,Y;MACI,OAAuC,oBAA
nB,UAAV,IAAH,EAAa,CAAmB,C;K;0CAG3C,iB;MACI,OAAU,IAAI,EAAP,GAAY,K;K;kCAGvB,Y;MAEI,OA
AO,M;K;;+DAIf,gB;MAEI,YAAY,MAAY,IAAK,OAAjB,C;MACZ,sBAAU,IAAV,a;QACI,UAAU,KAAK,CAAL
,C;QACV,IAAI,oBAAJ,C;UACI,MAAM,CAAN,IAAW,EAAS,MAAM,MAAK,GAAL,C;;UAE1B,MAAM,CAAN
,IAAW,G;;;MAGnB,OAAO,EAAS,OAAO,OAAM,EAAN,EAAgB,KAAhB,C;K;IAG3B,2B;MAMW,WAAO,S;M
AlBd,YAAY,MAAY,IAAK,OAAjB,C;MACZ,sBAAU,IAAV,a;QACI,UAAU,KAAK,CAAL,C;QACV,IAAI,oBA
AJ,C;UACI,MAAM,CAAN,IAAW,EAAS,MAAM,MAAK,GAAL,C;;UAE1B,MAAM,CAAN,IAAW,G;;;MAYnB,
OATO,EAAS,OAAO,OAAM,EAAN,EAAgB,KAAhB,C;K;IAY3B,oC;MAWI,WAAqB,S;MACrB,IAAI,qBAAmB
,CAAY,OAAd,KAA2B,SAAhD,C;QAjCA,YAAY,MAkCM,IAlCW,OAAjB,C;QACZ,sBAiCkB,IAjClB,a;UACI,U
AgCc,IAhCJ,CAAK,CAAL,C;UACV,IAAI,oBAAJ,C;YACI,MAAM,CAAN,IAAW,EAAS,MAAM,MAAK,GAAL
,C;;YAE1B,MAAM,CAAN,IAAW,G;;;QA4Bf,OAzBG,EAAS,OAAO,OAAM,EAAN,EAAgB,KAAhB,C;;QA2Bn
B,WAAW,C;QACX,0BAAU,IAAV,e;UACY,IAAoB,I;UAA5B,eAAQ,QAAoB,OAApB,IAAQ,CAAH,GAAG,CA
AY,OAApB,oCAAR,K;;QAEJ,aAAa,IAAO,CAAC,YAAR,CAAqB,IAArB,C;QE3FjB,IF4FyB,CE5FhB,OAAL,K
AAkB,SAAtB,C;UF4F4B,ME3FxB,UF2FqB,CE3FF,O;;QF4FnB,OAAO,C;QACP,0BAAU,IAAV,e;UAE0B,YAC

X,M;UAFX,YAAU,IAAQ,CAAH,GAAG,C;UACI,SAAJ,KAAI,O;UAAtB,aAAU,CAAV,kB;YACI,OAAO,aAAP,
EAAO,qBAAP,YAAiB,MAAI,CAAJ,C;;;QAGzB,OAAO,M;;K;IAIf,0B;MACgC,WAAS,c;MAAT,YAA4B,EAAE
,MAAM,KAAX,CAAiB,SAAjB,C;MAWrD,eAAiB,I;MAXW,OAYrB,K;K;IAVX,uB;MAC6B,WAAS,W;MAAT,
YAAsB,IAAO,WAAP,CAAmB,SAAnB,C;MAQ/C,eAAiB,I;MARQ,OASlB,K;K;IAPX,uB;MAC6B,WAAS,W;M
AAT,YAAyB,EAAE,MAAM,KAAX,CAAiB,SAAjB,C;MAK/C,eAAiB,I;MALQ,OAMlB,K;K;2DAJX,uB;MAGI,
eAAiB,I;MACjB,OAAO,K;K;kEG9MX,yB;MAAA,0B;MAAA,uB;QASI,OAAoB,OAAb,IjDoR+B,KAAL,GAAiB
,KiDpR9B,C;O;KATxB,C;ICIqC,2C;MAAC,8C;MAClC,eAAsB,C;MACtB,wBAA+B,C;MAC/B,gBAA6B,I;MAC
7B,mBAAsC,I;MACtC,qBAAyC,I;MAEzC,yBAAgD,yBAAmB,Q;MAEnE,sBAAgD,I;K;wFAFhD,Y;MAAA,6B;
K;0CAIA,Y;MAEY,kBADR,M;MAAA,U;MAAA,2C;QAAA,e;;QAES,gBADD,2CAAQ,yCAAR,gDAAwD,IAAx
D,6BAAiE,I;QACzD,sBpCwEd,S;QoC1EF,SpC2EG,S;;MoC3EH,a;K;iDAIJ,kB;MACI,kBAAc,IAAd,C;MACiC,o
B;MCuBrB,Q;MADR,IDtBsB,MCsBtB,W;QADJ,mBACiB,I;;QADjB,mBAEY,QDvBc,MCuBd,+D;;MDvBZ,yC;
MACA,2BAAmC,MAAO,kBAA1C,C;MAGA,OAAO,IAAP,C;QpCoCY,gBoCnCH,S;;QACD,iBAAiB,8B;QAGjB
,IAAI,0BAAJ,C;UACI,qBAAc,e;;UAEd,oBAAQ,0B;UACR,wBAAY,kB;;;UAIZ,cAAc,oB;UACd,IAAI,YAAY,yB
AAhB,C;YAAqC,M;UACrC,kBAAgB,O;UAChB,qBAAmB,I;;UAEnB,kBAAgB,I;UAChB,qBAAmB,S;;QAGvB,
gC;QAEA,IAAI,wCAAJ,C;UAEI,YAAU,U;;UAGV,U;UAAA,0C;YEThB,8BDgDQ,WAAO,qBAAP,CChDR,C;Y
FSgB,a;;YAAA,a;UAAA,mB;YAEK,UEpBrB,oBDgDQ,WD5B+B,eC4B/B,CChDR,C;;UFqBgB,M;;;K;mDAMhB,
Y;MACI,kBAAkB,mB;MAClB,IAAI,uBAAuB,gBAAgB,IAA3C,C;QACI,uCAAQ,yCAAR,EAAmC,wCAA+B,W
AA/B,C;;MAEvC,sBAAoB,mC;K;;IAM5B,iC;MAAA,qC;K;gGAEQ,Y;MvC0DyC,MAAM,6BuC1DjC,uCvC0D+
D,WAA9B,C;K;yDuCxDnD,kB;MvCwD6C,MAAM,6BuCvDzC,uCvCuDuE,WAA9B,C;K;+CuCpDnD,Y;MAAk
C,8C;K;;;IARtC,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;IGyDA,mG;IAAA,yH;IAAA,6F;MAKW,kC;MAAS,4C;
K;IALpB,sEAMQ,Y;MACI,Q;MAAA,sC;QAAiB,U;;MACjB,OAAO,oB;K;IARnB,6G;sJAjIA,iC;MAgBU,OAAK,
SAAL,CAAiB,UAAjB,EAA6B,KAA7B,C;K;wJAEV,2C;MAiBU,OAAK,SAAL,CAAiB,QAAjB,EAA2B,UAA3B,
EAAuC,KAAvC,C;K;wJAEV,kD;MAKU,OAAK,SAAL,CAAiB,QAAjB,EAA2B,KAA3B,EAAkC,UAAlC,EAA8
C,KAA9C,C;K;IAgC6C,oG;MAAA,mB;QAC3C,OAAK,iCAAL,CAAiB,kBAAjB,C;O;K;IA/BZ,6D;MA0BI,IAAS
,SAAY,OAAjB,IAA2B,CAA/B,C;QAAA,OAES,SAAL,CAAiB,UAAjB,EAA6B,IAA7B,C;;QA8D0B,Q;QAhE9B,
4DAImD,0DAJnD,EAgE8B,qBA5DS,UA4DT,qCAhE9B,C;;K;IAwCmD,wH;MAAA,mB;QAC3C,OAAK,iCAAL,
CAAiB,gBAAjB,EAA2B,kBAA3B,C;O;K;IAhCZ,yE;MA2BI,IAAS,SAAY,OAAjB,IAA2B,CAA/B,C;QAAA,OA
ES,SAAL,CAAiB,QAAjB,EAA2B,UAA3B,EAAuC,IAAvC,C;;QA0B0B,Q;QA5B9B,4DAImD,sEAJnD,EA4B8B,
qBAxBS,UAwBT,qCA5B9B,C;;K;IASJ,gC;MAWK,kBAAD,M;MAAA,kBAAC,qEAAD,4DAA2C,S;K;6CAG/C,
yB;MAAA,mG;MAAA,yH;MAAA,6F;QAKW,kC;QAAS,4C;O;MALpB,sEAMQ,Y;QACI,Q;QAAA,sC;UAAiB,U
;;QACjB,OAAO,oB;O;MARnB,6G;MAAA,oC;QAKkC,Q;QAA9B,mEAA8B,oEAA9B,C;O;KALJ,C;iFC7HA,a;M
AC6C,OAAA,MAAa,YAAW,CAAX,C;K;ICM3B,iC;;MAA6E,Q;MAAA,+BAAS,I;sCAAlB,O,2DAAA,O;;;;K;;;;;;
;;IAC/F,2B;MAAA,iD;MAAuB,oBAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,sC;MAAA,iD;MAAuC,oBA
AK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;IACA,oC;MAAA,iD;MAAwC,oBAAK,SAAL,EAAgB,KAAhB,C;MAAx
C,Y;K;IAI+B,mC;;MAA6E,Q;MAAA,+BAAS,I;sCAAlB,O,2DAAA,O;;;;K;;;;;;;;IACnG,+B;MAAA,mD;MAAuB,
sBAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,0C;MAAA,mD;MAAuC,sBAAK,OAAL,EAAc,IAAd,C;MA
AvC,Y;K;IACA,wC;MAAA,mD;MAAwC,sBAAK,SAAL,EAAgB,KAAhB,C;MAAxC,Y;K;IAGsC,0C;MAA0D,q
BAAU,OAAV,EAAmB,KAAnB,C;;K;;IAChG,sC;MAAA,0D;MAAuB,6BAAK,IAAL,EAAW,IAAX,C;MAAvB,Y
;K;IACA,iD;MAAA,0D;MAAuC,6BAAK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;IACA,+C;MAAA,0D;MAAwC,6B
AAK,SAAL,EAAgB,KAAhB,C;MAAxC,Y;K;IAG8C,kD;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IACx
G,8C;MAAA,kE;MAAuB,qCAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,yD;MAAA,kE;MAAuC,qCAAK,O
AAL,EAAc,IAAd,C;MAAvC,Y;K;IACA,uD;MAAA,kE;MAAwC,qCAAK,SAAL,EAAgB,KAAhB,C;MAAxC,Y;
K;IAG2C,+C;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IACrG,2C;MAAA,+D;MAAuB,kCAAK,IAAL,E
AAW,IAAX,C;MAAvB,Y;K;IACA,sD;MAAA,+D;MAAuC,kCAAK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;IACA,o
D;MAAA,+D;MAAwC,kCAAK,SAAL,EAAgB,KAAhB,C;MAAxC,Y;K;IAG+C,4C;8BAAwD,O;;K;;IACvG,+C;
MAAA,mE;MAAuB,sCAAK,IAAL,C;MAAvB,Y;K;IAGqD,yD;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;
;IAC/G,qD;MAAA,yE;MAAuB,4CAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,gE;MAAA,yE;MAAuC,4CA
AK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;IACA,8D;MAAA,yE;MAAwC,4CAAK,SAAL,EAAgB,KAAhB,C;MAA

xC,Y;K;IAGmD,uD;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IAC7G,mD;MAAA,uE;MAAuB,0CAAK,I
AAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,8D;MAAA,uE;MAAuC,0CAAK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;I
ACA,4D;MAAA,uE;MAAwC,0CAAK,SAAL,EAAgB,KAAhB,C;MAAxC,Y;K;IAI2C,wC;sCAAgE,O;;K;;IAC3G
,2C;MAAA,+D;MAAuB,kCAAK,IAAL,C;MAAvB,Y;K;IAI0C,uC;8BAAwD,O;;K;;IAClG,0C;MAAA,8D;MAAu
B,iCAAK,IAAL,C;MAAvB,Y;K;IAGwC,qC;8BAAwD,O;;K;;IAChG,wC;MAAA,4D;MAAuB,+BAAK,IAAL,C;
MAAvB,Y;K;IAIJ,wC;MACmD,mBAAM,OAAN,EAAe,KAAf,C;;K;;IAC/C,oC;MAAA,wD;MAAuB,sBAAK,IA
AL,Q;MAAvB,Y;K;IACA,+C;MAAA,wD;MAAgC,2BAAK,OAAL,EAAc,IAAd,C;MAAhC,Y;K;IACA,+C;MAA
A,wD;MAAiD,IAAY,I;MAAzB,2BAAa,SAAR,OAAQ,CAAb,EAAyB,sDAAzB,C;MAApC,Y;K;IAG4C,yC;8BA
AwD,O;;K;;IACpG,4C;MAAA,gE;MAAuB,mCAAK,IAAL,C;MAAvB,Y;K;IAIyC,sC;8BAAwD,O;;K;;IACjG,yC;
MAAA,6D;MAAuB,gCAAK,IAAL,C;MAAvB,Y;K;IAGkD,sD;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;
IAC5G,kD;MAAA,sE;MAAuB,yCAAK,IAAL,EAAW,IAAX,C;MAAvB,Y;K;IACA,6D;MAAA,sE;MAAuC,yCA
AK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;IACA,2D;MAAA,sE;MAAwC,yCAAK,SAAL,EAAgB,KAAhB,C;MAAx
C,Y;K;IAG0D,8D;MAA0D,4BAAiB,OAAjB,EAA0B,KAA1B,C;;K;;IACpH,0D;MAAA,8E;MAAuB,iDAAK,IAA
L,EAAW,IAAX,C;MAAvB,Y;K;IACA,qE;MAAA,8E;MAAuC,iDAAK,OAAL,EAAc,IAAd,C;MAAvC,Y;K;IAC
A,mE;MAAA,8E;MAAwC,iDAAK,SAAL,EAAgB,KAAhB,C;MAAxC,Y;K;6FClGJ,yB;MAEI,OAAU,GAAG,CA
AC,QAAD,C;K;mFAGjB,oB;MAEI,OAAG,GAAH,GAAS,G;K;6ETVb,a;MAK8C,cAAU,C;K;6ECHxD,Y;MAG+
C,S;K;IA6B/C,2B;MAG4D,0BAAe,WAAf,C;K;IAE5D,mC;MAIwF,0BAAe,WAAf,C;K;IAExF,mC;MAKwE,0BA
Ae,WAAf,C;K;IAGxE,4B;MAI8D,Q;MAH1D,aAAkB,GAAI,O;MACtB,aAAkB,GAAI,O;MACtB,YAAiB,C;MAC
jB,OAAO,QAAQ,MAAR,IAAkB,QAAQ,MAAjC,C;QAAyC,IAAI,KAAJ,IAAa,IAAI,YAAJ,EAAI,oBAAJ,O;;MA
CtD,OAAO,G;K;IAIX,wD;MAMuC,Q;MALnC,aAAa,MAAO,OAAM,CAAN,EAAS,OAAT,C;MA0BpB,IAzBc,M
AyBL,OAAL,KAAkB,SAAtB,C;QAzBsB,MA0BlB,UA1BU,MA0BS,O;;MAzBvB,YAAiB,MAAO,O;MACxB,IA
AI,UAAU,KAAd,C;QACI,gBAAgB,O;QAChB,OAAO,QAAQ,OAAf,C;UAAwB,OAAO,YAAP,EAAO,oBAAP,U
AAkB,Y;;;MAE9C,OAAO,M;K;IAGX,gD;MAKoB,UAAmB,M;MAJnC,aAAa,KAAM,Q;MACnB,MAAO,OAAP,
IAAiB,UAAW,K;MAc5B,IAbc,KAaL,OAAL,KAAkB,SAAtB,C;QAbqB,MAcjB,UAdU,KAcS,O;;MAbvB,YAAiB
,KAAM,O;MACP,4B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QAAY,OAAO,cAAP,EAAO,sBAAP,YAAkB,O;;M
AC9C,OAAO,M;K;IAGX,yD;MAEoB,UAAgB,M;MADhC,YAAY,U;MACI,4B;MAAhB,OAAgB,cAAhB,C;QAA
gB,yB;QAAY,IAAI,cAAJ,EAAI,sBAAJ,YAAe,O;;MAC3C,OAAO,G;K;oFAGX,oB;MACI,IAAI,IAAK,OAAL,K
AAkB,SAAtB,C;QACI,YAAc,IAAK,O;;K;0EAI3B,wB;MAA+D,OAAA,MAAa,QAAO,GAAP,EAAY,OAAZ,C;K
;IS/F5E,mC;MAOI,kBAAkB,MAAa,eAAc,SAAd,C;MAC/B,iBAAiB,MAAa,eAAc,IAAd,C;MAC9B,OAAW,gBA
Ae,UAAnB,GAA+B,SAA/B,GAAyC,CAAC,S;K;0ECUrD,2B;MAKyE,OAAA,MAAa,gBAAe,IAAf,C;K;4EAyBt
F,2B;MAKsE,OAAA,MAAa,eAAc,IAAd,C;K;kEAGnF,qB;MACgD,OAAA,MAAa,KAAK,UAAS,GAAT,EAAc,I
AAd,C;K;wEAChC,qB;MAAQ,OAAK,SAAY,a;K;0EACxB,qB;MAAQ,OAAK,SAAY,c;K;IC3D5D,0D;MAGI,O
AAO,I;K;ICHX,sC;MAMsD,OAAA,SAAY,UAAS,WAAW,KAAX,CAAT,C;K;IhDKlE,uC;MhBynBW,Q;MAAA,
IgBnnBgB,KhBmnBZ,IAAS,CAAT,IgBnnBY,KhBmnBE,IAAS,wBAA3B,C;QAAA,OAAsC,UgBnnBtB,KhBmnB
sB,C;;QgBnnBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MhB0nBW,Q;MAAA,I
gBpnBgB,KhBonBZ,IAAS,CAAT,IgBpnBY,KhBonBE,IAAS,0BAA3B,C;QAAA,OAAsC,UgBpnBtB,KhBonBsB,
C;;QgBpnBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MhB2nBW,Q;MAAA,IgBr
nBgB,KhBqnBZ,IAAS,CAAT,IgBrnBY,KhBqnBE,IAAS,0BAA3B,C;QAAA,OAAsC,UgBrnBtB,KhBqnBsB,C;;Q
gBrnBb,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MhB4nBW,Q;MAAA,IgBtnBgB
,KhBsnBZ,IAAS,CAAT,IgBtnBY,KhBsnBE,IAAS,0BAA3B,C;QAAA,OAAsC,UgBtnBtB,KhBsnBsB,C;;QgBtnB
b,MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MhB6nBW,Q;MAAA,IgBvnBgB,KhB
unBZ,IAAS,CAAT,IgBvnBY,KhBunBE,IAAS,0BAA3B,C;QAAA,OAAsC,UgBvnBtB,KhBunBsB,C;;QgBvnBb,
MAAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MhB8nBW,Q;MAAA,IgBxnBgB,KhBw
nBZ,IAAS,CAAT,IgBxnBY,KhBwnBE,IAAS,0BAA3B,C;QAAA,OAAsC,UgBxnBtB,KhBwnBsB,C;;QgBxnBb,M
AAM,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MhB+nBW,Q;MAAA,IgBznBgB,KhBynB
Z,IAAS,CAAT,IgBznBY,KhBynBE,IAAS,0BAA3B,C;QAAA,OAAsC,UgBznBtB,KhBynBsB,C;;QgBznBb,MAA
M,8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,uC;MhBgoBW,Q;MAAA,IgB1nBgB,KhB0nBZ,I
AAS,CAAT,IgB1nBY,KhB0nBE,IAAS,0BAA3B,C;QAAA,OAAsC,UgB1nBtB,KhB0nBsB,C;;QgB1nBb,MAAM,

8BAA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,wC;MhBioBW,Q;MAAA,IgB3nBgB,KhB2nBZ,IAA
S,CAAT,IgB3nBY,KhB2nBE,IAAS,0BAA3B,C;QAAA,OAAsC,UgB3nBtB,KhB2nBsB,C;;QgB3nBb,MAAM,8B
AA0B,iCAAuB,gBAAvB,MAA1B,C;;MAAtC,W;K;IAGJ,2B;MAII,OAAO,cAAa,SAAb,C;K;oFAGX,yB;MAAA,
gD;MAAA,4B;QAKI,OAAsC,OAA/B,SAA+B,C;O;KAL1C,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAuC,
OAAhC,SAAgC,C;O;KAL3C,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAqC,OAA9B,SAA8B,C;O;KALzC,
C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAsC,OAA/B,SAA+B,C;O;KAL1C,C;oFAQA,yB;MAAA,gD;MA
AA,4B;QAKI,OAAuC,OAAhC,SAAgC,C;O;KAL3C,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAwC,OAAj
C,SAAiC,C;O;KAL5C,C;oFAQA,yB;MAAA,gD;MAAA,4B;QAKI,OAAyC,OAAlC,SAAkC,C;O;KAL7C,C;IAY
W,2C;MAAA,8B;MAAS,uB;K;4FACW,Y;MAAQ,OAAA,gBAAY,O;K;6CAC3C,Y;MAAkC,OAAA,gBhB8nP/B,
YAAQ,C;K;oDgB7nPX,mB;MAAgD,OAAY,WAAZ,gBAAY,EAAS,OAAT,C;K;iDAC5D,iB;MACI,oCAAa,2BA
AkB,KAAlB,EAAyB,SAAzB,C;MACb,OAAO,6BAAY,KAAZ,E;K;mDAEX,mB;MAES,Q;MAAL,IAAI,eAAC,u
FAAD,CAAJ,C;QAAgC,OAAO,E;MACvC,OAAmB,UAAZ,gBAAY,EAAQ,OAAR,C;K;uDAEvB,mB;MAES,Q;
MAAL,IAAI,eAAC,uFAAD,CAAJ,C;QAAgC,OAAO,E;MACvC,OAAmB,cAAZ,gBAAY,EAAY,OAAZ,C;K;;IA
pB/B,6B;MAII,0C;K;IAqBJ,+C;MAaI,OAAY,kBAAL,SAAK,EAAkB,KAAlB,C;K;IAqBhB,0C;MASI,OAAY,oB
AAL,SAAK,C;K;IAehB,0C;MAYI,OAAY,oBAAL,SAAK,C;K;IAkBhB,2C;MAWI,OAAY,cAAL,SAAK,EAAc,K
AAd,C;K;IAGhB,2C;MAWI,OAAY,cAAL,SAAK,EAAc,KAAd,C;K;IAGhB,4C;MAWI,OAAY,cAAL,SAAK,EA
Ac,KAAd,C;K;IAGhB,4C;MAWI,OAAY,cAAL,SAAK,EAAc,KAAd,C;K;IAGhB,4C;MAWI,OAAY,cAAL,SAA
K,EAAc,KAAd,C;K;IAGhB,4C;MAWI,OAAY,cAAL,SAAK,EAAc,KAAd,C;K;IAGhB,4C;MAWI,OAAY,cAAL,
SAAK,EAAc,KAAd,C;K;IAGhB,4C;MAWI,OAAY,cAAL,SAAK,EAAc,KAAd,C;K;IAGhB,4C;MAWI,OAAY,c
AAL,SAAK,EAAc,KAAd,C;K;IAwHhB,sC;MAOI,OAAY,gBAAL,SAAK,C;K;IAGhB,sC;MAOI,OAAY,gBAAL,
SAAK,C;K;IAGhB,uC;MAOI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OAAY,gBAAL,SAAK,C;K;IAGhB,u
C;MAOI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OAAY,gB
AAL,SAAK,C;K;IAGhB,uC;MAOI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MAOI,OAAY,gBAAL,SAAK,C;K;IA
oFhB,sC;MASI,OAAY,gBAAL,SAAK,C;K;IAGhB,sC;MASI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAA
Y,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAY,gBAAL,SAAK,C;
K;IAGhB,uC;MASI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MASI,
OAAY,gBAAL,SAAK,C;K;IAGhB,uC;MASI,OAAY,gBAAL,SAAK,C;K;wFAsGhB,yB;MAAA,8C;MAAA,kF;Q
AmB0E,iC;UAAA,oBAAyB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACvI,UAAU,SAA
V,EAAgB,WAAhB,EAA6B,iBAA7B,EAAgD,UAAhD,EAA4D,QAA5D,C;QACA,OAAO,W;O;KArBX,C;wFAw
BA,yB;MAAA,8C;MAAA,kF;QAmBoE,iC;UAAA,oBAAyB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,
WAAgB,gB;QACjI,UAAU,SAAV,EAA0C,WAA1C,EAAiF,iBAAjF,EAAoG,UAApG,EAAgH,QAAhH,C;QACA,
OAAO,W;O;KArBX,C;wFAwBA,yB;MAAA,8C;MAAA,kF;QAmBsE,iC;UAAA,oBAAyB,C;QAAG,0B;UAAA,a
AAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACnI,UAAU,SAAV,EAA2C,WAA3C,EAAmF,iBAAnF,EAAsG,UAAt
G,EAAkH,QAAlH,C;QACA,OAAO,W;O;KArBX,C;wFAwBA,yB;MAAA,8C;MAAA,kF;QAmBkE,iC;UAAA,oB
AAyB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QAC/H,UAAU,SAAV,EAAyC,WAAzC,E
AA+E,iBAA/E,EAAkG,UAAlG,EAA8G,QAA9G,C;QACA,OAAO,W;O;KArBX,C;wFAwBA,yB;MAAA,8C;MA
AA,kF;QAmBoE,iC;UAAA,oBAAyB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACjI,UA
AU,SAAV,EAA0C,WAA1C,EAAiF,iBAAjF,EAAoG,UAApG,EAAgH,QAAhH,C;QACA,OAAO,W;O;KArBX,C;
wFAwBA,yB;MAAA,8C;MAAA,kF;QAmBsE,iC;UAAA,oBAAyB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;U
AAA,WAAgB,gB;QACnI,UAAU,SAAV,EAA2C,WAA3C,EAAmF,iBAAnF,EAAsG,UAAtG,EAAkH,QAAlH,C;
QACA,OAAO,W;O;KArBX,C;uFAwBA,yB;MAAA,8C;MAAA,kF;QAmBwE,iC;UAAA,oBAAyB,C;QAAG,0B;
UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACrI,UAAU,SAAV,EAA4C,WAA5C,EAAqF,iBAArF,EAAw
G,UAAxG,EAAoH,QAApH,C;QACA,OAAO,W;O;KArBX,C;yFAwBA,yB;MAAA,8C;MAAA,kF;QAmB0E,iC;U
AAA,oBAAyB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QACvI,UAAU,SAAV,EAA6C,W
AA7C,EAAuF,iBAAvF,EAA0G,UAA1G,EAAsH,QAAtH,C;QACA,OAAO,W;O;KArBX,C;yFAwBA,yB;MAAA,
8C;MAAA,kF;QAmBoE,iC;UAAA,oBAAyB,C;QAAG,0B;UAAA,aAAkB,C;QAAG,wB;UAAA,WAAgB,gB;QA
CjI,UAAU,SAAV,EAA0C,WAA1C,EAAiF,iBAAjF,EAAoG,UAApG,EAAgH,QAAhH,C;QACA,OAAO,W;O;KA
rBX,C;oFAwBA,qB;MAOI,OAAY,SAAY,Q;K;oFAG5B,qB;MAOI,OAAY,SAAY,Q;K;oFAG5B,qB;MAOI,OAA

Y,SAAY,Q;K;qFAG5B,qB;MAOI,OAAY,SAAY,Q;K;IAG5B,8B;MAMW,WAAS,W;MAAT,YAA2B,SAAY,Q;M
kCl7B9C,eAAiB,I;MlCk7BjB,OkCj7BO,K;K;qFlCo7BX,qB;MAOI,OAAY,SAAY,Q;K;qFAG5B,qB;MAOI,OAA
Y,SAAY,Q;K;IAG5B,8B;MAMW,WAAS,c;MAAT,YAA8B,SAAY,Q;MkC/8BjD,eAAiB,I;MlC+8BjB,OkC98BO,
K;K;IlCi9BX,8B;MAMW,WAAS,W;MAAT,YAA2B,SAAY,Q;MkCx9B9C,eAAiB,I;MlCw9BjB,OkCv9BO,K;K;I
lC09BX,uC;MD5oCI,IAAI,ECspCI,WAAW,CDtpCf,CAAJ,C;QACI,cCqpCoB,0C;QDppCpB,MAAM,gCAAyB,O
AAQ,WAAjC,C;;MCqpCV,OAAO,SAAS,SAAT,EAAe,cAAU,OAAV,CAAf,C;K;IAGX,uC;MD1pCI,IAAI,ECoq
CI,WAAW,CDpqCf,CAAJ,C;QACI,cCmqCoB,0C;QDlqCpB,MAAM,gCAAyB,OAAQ,WAAjC,C;;MCmqCV,OA
AO,SAAS,SAAT,EAAe,eAAW,OAAX,CAAf,C;K;IAGX,uC;MDxqCI,IAAI,ECkrCI,WAAW,CDlrCf,CAAJ,C;QA
CI,cCirCoB,0C;QDhrCpB,MAAM,gCAAyB,OAAQ,WAAjC,C;;MCirCV,OAAO,SAAS,SAAT,EAAe,eAAS,OAA
T,CAAf,C;K;IAGX,uC;MDtrCI,IAAI,ECgsCI,WAAW,CDhsCf,CAAJ,C;QACI,cC+rCoB,0C;QD9rCpB,MAAM,g
CAAyB,OAAQ,WAAjC,C;;MC+rCH,WAAS,W;MAAT,YAAsB,gBAAgB,SAAhB,EAAsB,OAAtB,K;MkChhC7B
,eAAiB,I;MlCghCjB,OkC/gCO,K;K;IlCkhCX,uC;MDpsCI,IAAI,EC8sCI,WAAW,CD9sCf,CAAJ,C;QACI,cC6sCo
B,0C;QD5sCpB,MAAM,gCAAyB,OAAQ,WAAjC,C;;MC6sCV,OAAO,SAAS,SAAT,EAAe,iBAAW,OAAX,CAA
f,C;K;IAGX,uC;MDltCI,IAAI,EC4tCI,WAAW,CD5tCf,CAAJ,C;QACI,cC2tCoB,0C;QD1tCpB,MAAM,gCAAyB,
OAAQ,WAAjC,C;;MC2tCV,OAAO,SAAS,SAAT,EAAe,iBAAY,OAAZ,CAAf,C;K;IAGX,uC;MDhuCI,IAAI,EC0
uCI,WAAW,CD1uCf,CAAJ,C;QACI,cCyuCoB,0C;QDxuCpB,MAAM,gCAAyB,OAAQ,WAAjC,C;;MCyuCH,W
AAS,c;MAAT,YAAyB,gBAAgB,SAAhB,EAAsB,OAAtB,EAA+B,KAA/B,C;MkC1jChC,eAAiB,I;MlC0jCjB,OkC
zjCO,K;K;IlC4jCX,uC;MD9uCI,IAAI,ECwvCI,WAAW,CDxvCf,CAAJ,C;QACI,cCuvCoB,0C;QDtvCpB,MAAM,
gCAAyB,OAAQ,WAAjC,C;;MCuvCH,WAAS,W;MAAT,YAAsB,SAAS,SAAT,EAAe,iBAAU,OAAV,CAAf,C;M
kCxkC7B,eAAiB,I;MlCwkCjB,OkCvkCO,K;K;IlC0kCX,uC;MD5vCI,IAAI,ECuwCI,WAAW,CDvwCf,CAAJ,C;Q
ACI,cCswCoB,0C;QDrwCpB,MAAM,gCAAyB,OAAQ,WAAjC,C;;MCswCV,OAAO,gBAAgB,SAAhB,EAAsB,O
AAtB,EAA+B,IAA/B,C;K;IAGX,sD;MAWI,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MAC
b,OAAY,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;K;IAG5B,sD;MAUI,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,
EAAsC,gBAAtC,C;MACb,OAAY,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;K;IAG5B,sD;MAUI,oCAAa,2BAAkB
,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,OAAY,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;K;IAG5B,s
D;MAUI,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,OAAY,SAAY,OAAM,SAAN,EA
AiB,OAAjB,C;K;IAG5B,sD;MAUI,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACN,WAAS
,W;MAAT,YAA2B,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;MkC9pC9C,eAAiB,I;MlC8pCjB,OkC7pCO,K;K;IlC
gqCX,sD;MAUI,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,OAAY,SAAY,OAAM,SA
AN,EAAiB,OAAjB,C;K;IAG5B,sD;MAUI,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,
OAAY,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;K;IAG5B,uD;MAUI,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,E
AAsC,gBAAtC,C;MACN,WAAS,c;MAAT,YAA8B,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;MkCxsCjD,eAAiB,I
;MlCwsCjB,OkCvsCO,K;K;IlC0sCX,uD;MAUI,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;M
ACN,WAAS,W;MAAT,YAA2B,SAAY,OAAM,SAAN,EAAiB,OAAjB,C;MkCttC9C,eAAiB,I;MlCstCjB,OkCrtC
O,K;K;IlCwtCX,wD;MAWgD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC/E,oCAAa,2BAAkB,SAAl
B,EAA6B,OAA7B,EAAsC,gBAAtC,C;MiDz3CD,ejD03CD,OiD13CC,EjD03CQ,SiD13CR,EjD03CmB,OiD13Cn
B,C;K;IjD63ChB,wD;MAWgD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC/E,oCAAa,2BAAkB,SA
AlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MiDz4CD,ejD04CD,OiD14CC,EjD04CQ,SiD14CR,EjD04CmB,OiD14C
nB,C;K;IjD64ChB,wD;MAWkD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACjF,oCAAa,2BAAkB,SA
AlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MiDz5CD,ejD05CD,OiD15CC,EjD05CQ,SiD15CR,EjD05CmB,OiD15C
nB,C;K;IjD65ChB,wD;MAW8C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC7E,oCAAa,2BAAkB,S
AAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MiDz6CD,ejD06CD,OiD16CC,EjD06CQ,SiD16CR,EjD06CmB,OiD16
CnB,C;K;IjD66ChB,wD;MAWgD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC/E,oCAAa,2BAAkB,S
AAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MiDz7CD,ejD07CD,OiD17CC,EjD07CQ,SiD17CR,EjD07CmB,OiD17
CnB,C;K;IjD67ChB,wD;MAWkD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACjF,oCAAa,2BAAkB,S
AAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MiDz8CD,ejD08CD,OiD18CC,EjD08CQ,SiD18CR,EjD08CmB,OiD18
CnB,C;K;IjD68ChB,wD;MAWoD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACnF,oCAAa,2BAAkB,S
AAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MiDz9CD,ejD09CD,OiD19CC,EjD09CQ,SiD19CR,EjD09CmB,OiD19

CnB,C;K;IjD69ChB,yD;MAWsD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACrF,oCAAa,2BAAkB,S
AAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MiDz+CD,ejD0+CD,OiD1+CC,EjD0+CQ,SiD1+CR,EjD0+CmB,OiD1
+CnB,C;K;IjD6+ChB,yD;MAWgD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAC/E,oCAAa,2BAAkB,
SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MiDz/CD,ejD0/CD,oBiD1/CC,EjD0/CQ,SiD1/CR,EjD0/CmB,OiD1/
CnB,C;K;iFjD6/ChB,8B;MAKI,OAAY,SAAY,QAAO,CAAQ,OAAR,CAAP,C;K;iFAG5B,yB;MAwIA,iD;MAxIA
,qC;QAKI,OAwIO,gCAxIK,eAAY,OAAZ,EAwIL,C;O;KA7IX,C;iFAQA,yB;MAwIA,iD;MAxIA,qC;QAKI,OAwI
O,gCAxIK,gBAAa,OAAb,EAwIL,C;O;KA7IX,C;iFAQA,yB;MAwIA,iD;MAxIA,qC;QAKI,OAwIO,gCAxIK,gBA
AW,OAAX,EAwIL,C;O;KA7IX,C;iFAQA,yB;MAwIA,iD;MAxIA,qC;QAKI,OAwIO,gCAxIK,mBAAY,OAAZ,C
AwIL,C;O;KA7IX,C;iFAQA,yB;MAwIA,iD;MAxIA,qC;QAKI,OAwIO,gCAxIK,kBAAa,OAAb,EAwIL,C;O;KA7
IX,C;gFAQA,yB;MAwIA,iD;MAxIA,qC;QAKI,OAwIO,gCAxIK,kBAAc,OAAd,EAwIL,C;O;KA7IX,C;iFAQA,y
B;MAwIA,iD;MAxIA,qC;QAKI,OAwIO,gCAxIK,sBAAe,OAAf,CAwIL,C;O;KA7IX,C;iFAQA,yB;MAwIA,iD;M
AxIA,qC;QAKI,OAwIO,gCAxIK,mBAAY,OAAZ,CAwIL,C;O;KA7IX,C;IAQA,sC;MAKI,OAAO,oBAAoB,SAA
pB,EAA0B,QAA1B,C;K;IAGX,sC;MAII,OAAO,mBAAwB,UAAL,SAAK,EAAO,mBAAO,QAAS,KAAhB,IAAP,
CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,C;K;IAGX,sC;MAII,OAAO,mBAAwB,UAAL,SAAK,EAAO,mB
AAO,QAAS,KAAhB,IAAP,CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,C;K;IAGX,sC;MAII,OAAO,mBAA
wB,UAAL,SAAK,EAAO,mBAAO,QAAS,KAAhB,IAAP,CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,C;K;IA
GX,sC;MAII,OAAO,oBAAoB,SAApB,EAA0B,QAA1B,C;K;IAGX,sC;MAII,OAAO,mBAAwB,UAAL,SAAK,EA
AO,mBAAO,QAAS,KAAhB,IAAP,CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,C;K;IAGX,sC;MAII,OAAO,
mBAAwB,UAAL,SAAK,EAAO,mBAAO,QAAS,KAAhB,IAAP,CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,
C;K;IAGX,sC;MAII,OAAO,oBAAoB,SAApB,EAA0B,QAA1B,C;K;IAGX,sC;MAII,OAAO,mBAAwB,UAAL,SA
AK,EAAO,mBAAO,QAAS,KAAhB,IAAP,CAAxB,EAAsD,SAAK,OAA3D,EAAiE,QAAjE,C;K;iFAGX,+B;MA
KI,OAAY,SAAY,QAAO,QAAP,C;K;iFAG5B,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,
QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;K
ALX,C;iFAQA,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,
yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;
MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;MAAA,sC;QA
KI,OAAO,qBAAqB,SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBA
AqB,SAArB,EAA2B,QAA3B,C;O;KALX,C;iFAQA,yB;MAAA,iD;MAAA,sC;QAKI,OAAO,qBAAqB,SAArB,EA
A2B,QAA3B,C;O;KALX,C;8FAQA,8B;MAKI,OAAY,SAAY,QAAO,CAAQ,OAAR,CAAP,C;K;IAoBL,2B;MAA
sB,OAAA,CAAE,iBAAU,CAAV,C;K;IAP/C,2B;MAOI,IAAI,mBAAO,CAAX,C;QkDvwDY,elDuwDO,WkDvwD
P,C;;K;IlD4zDhB,2B;MAQI,IAAI,mBAAO,CAAX,C;QAAc,UAAU,SAAV,C;K;IAGlB,wC;MAQI,IAAI,mBAAO,
CAAX,C;QAAc,cAAc,SAAd,EAAoB,UAApB,C;K;IAGlB,gD;MAewD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,
UAAe,gB;MACvF,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,gBAAc,SAAd,EAAoB,S
AApB,EAA+B,OAA/B,EAAwC,cAAxC,C;K;IAGJ,gD;MAaiC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB
;MAChE,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EA
AoB,OAApB,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAakC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB
;MACjE,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAA
oB,OAApB,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAagC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;
MAC/D,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAA
oB,OAApB,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAaiC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;
MAChE,oCAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,gBAAc,SAAd,EAA8C,SAA9C,EA
AyD,OAAzD,EAAkE,cAAlE,C;K;IAGJ,gD;MAakC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACjE,o
CAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAAp
B,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAamC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAClE,o
CAAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAAp
B,C;MACvB,KAAT,QAAS,C;K;IAGb,gD;MAaiC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MAChE,oC
AAa,2BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,eAAoB,SAAY,UAAS,SAAT,EAAoB,OAApB,
C;MACvB,KAAT,QAAS,C;K;iFAGb,iC;MkD1+DgB,elDi/DD,UkDj/DC,C;K;iFlDo/DhB,iC;MkDp/DgB,elD2/DD

,UkD3/DC,C;K;iFlD8/DhB,iC;MkD9/DgB,elDqgED,UkDrgEC,C;K;iFlDwgEhB,iC;MkDxgEgB,elD+gED,UkD/gE
C,C;K;iFlDkhEhB,iC;MkDlhEgB,elDyhED,UkDzhEC,C;K;iFlD4hEhB,iC;MkD5hEgB,elDmiED,UkDniEC,C;K;iF
lDsiEhB,iC;MkDtiEgB,elD6iED,UkD7iEC,C;K;IlDgjEhB,yC;MAMI,IAAI,mBAAO,CAAX,C;QAAc,gBAAc,SAA
d,EAAoB,UAApB,C;K;IAGlB,+D;MAa0E,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,gB;MACzG,oCAAa,2
BAAkB,SAAlB,EAA6B,OAA7B,EAAsC,gBAAtC,C;MACb,gBAAc,SAAd,EAAoB,SAApB,EAA+B,OAA/B,EA
AwC,UAAxC,C;K;IAGJ,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,EAAS,M
AAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,
EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII
,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAG1B,mC;MAII,OAAO,EAAS,MAAM,MAAK,SAAL,C;K;IAOH,k
D;MAAA,wB;QAAW,qCAAK,KAAL,E;O;K;IAJlC,oC;MAII,OAAO,iBAAM,gBAAN,EAAY,gCAAZ,C;K;ImDn
pEX,oB;MAAA,wB;MAEI,6B;MACA,gC;MAKuB,UAAT,MAAS,EAAT,MAAS,EAAT,M;MAFV,eAAe,kE;MA
Cf,iBAAiB,eAAS,GAAT,C;MACE,sBAAT,QAAS,C;MAAT,mB;MAAA,kB;MAAA,kB;MAAV,8C;QACI,WAA
W,oBAAS,CAAT,CrC4BuB,IqC5BlC,IAA+B,C;;MAInC,qBAAqB,48C;MACrB,WAAW,mBAAmB,cAAnB,EAA
mC,UAAnC,EAA+C,IAA/C,C;MACX,YAAY,eAAS,IAAK,OAAL,GAAY,CAAZ,IAAT,C;MACZ,0BAAU,IAAV,
e;QACI,MAAM,MAAI,CAAJ,IAAN,IAAe,MAAM,GAAN,IAAW,KAAK,GAAL,CAAX,I;;MAEnB,yBAAoB,K;
MAGpB,oBAAoB,m/D;MACpB,4BAAuB,mBAAmB,aAAnB,EAAkC,UAAlC,EAA8C,IAA9C,C;K;;;IAvB/B,gC;
MAAA,+B;QAAA,c;;MAAA,wB;K;IA2BA,qC;MAKkB,IAJP,I;MACH,WAAO,EAAP,C;QAAe,W;WACf,WAAO
,IAAP,C;QAAgB,OAAI,CAAC,KAAO,CAAR,MAAc,CAAlB,GAAqB,QAAS,CAA9B,GAAqC,OAAS,E;;QAE1D
,QAAM,KAAK,CAAL,IAAN,C;eACI,C;YAAK,eAAS,E;YAAd,K;eACA,C;YAAK,OAAC,QAAS,CAAV,GAAiB,
E;YAAtB,K;;YACQ,cAAS,E;YAHrB,K;;;MAJR,W;K;IAYJ,qC;MAII,SAAS,SrCPiC,I;MqCS1C,YAAY,kBAAkB,
sBAAS,kBAA3B,EAA8C,EAA9C,C;MACZ,YAAY,sBAAS,kBAAT,CAA2B,KAA3B,C;MACZ,WAAW,sBAAS,
qBAAT,CAA8B,KAA9B,C;MACX,YAAY,kBAAkB,IAAlB,EAAwB,KAAK,KAAL,IAAxB,C;MAEZ,OAAW,U
AAS,EAAb,GAAyC,mDAAzC,GAAoD,K;K;IAG/D,8D;MAKiB,UAIE,M;MARf,aAAa,eAAS,YAAT,C;MACb,Y
AAY,C;MACZ,UAAU,C;MACV,YAAY,C;MACC,yB;MAAb,OAAa,cAAb,C;QAAa,iC;QACT,aAAa,WAAW,IrC
vBc,IqCuBzB,C;QACb,MAAM,MAAQ,CAAC,SAAW,EAAZ,KAAsB,K;QACpC,IAAI,SAAS,EAAb,C;UACI,OA
AO,cAAP,EAAO,sBAAP,YAAkB,G;UAClB,MAAM,C;UACN,QAAQ,C;;UAER,gBAAS,CAAT,I;;;MAGR,OAA
O,M;K;IClEX,+B;MAII,eAAe,CAAC,iBAAO,CAAP,IAAD,IAAa,CAAb,I;MACf,IAAI,WAAW,CAAf,C;QAAkB,
M;MAClB,mBAAmB,2B;MACnB,iBAAc,CAAd,WAAiB,QAAjB,U;QACI,UAAU,sBAAK,KAAL,C;QACV,sBA
AK,KAAL,EAAc,sBAAK,YAAL,CAAd,C;QACA,sBAAK,YAAL,EAAqB,GAArB,C;QACA,mC;;K;IjDbR,wB;M
AOI,OAAW,oBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAmB,C;K;mFAG9B,yB;MAkBA,iB;MAlBA,uB;QAMI,OA
kBO,MAAO,KAlBC,CAkBD,EAlBY,CAkBZ,C;O;KAxBlB,C;mFASA,yB;MASA,iB;MATA,uB;QAMI,OASO,M
AAO,KATC,CASD,EATY,CASZ,C;O;KAflB,C;mFASA,yB;MAAA,iB;MAAA,uB;QAMI,OAAO,MAAO,KAAI,
CAAJ,EAAO,CAAP,C;O;KANlB,C;mFASA,gB;MAMI,OAAW,kBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAmB,C
;K;mFAG9B,yB;MAAA,iB;MAAA,uB;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARlB,C;mFAWA
,yB;MAAA,iB;MAAA,uB;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KARlB,C;IAWA,2B;MAOI,OA
AO,SAAM,CAAN,EAAS,SAAM,CAAN,EAAS,CAAT,CAAT,C;K;mFAGX,yB;MAAA,iB;MAAA,0B;QAMI,OA
AO,MAAO,KAAM,CAAN,EAAiB,CAAjB,EAA4B,CAA5B,C;O;KANlB,C;mFASA,yB;MAAA,iB;MAAA,0B;Q
AMI,OAAO,MAAO,KAAM,CAAN,EAAiB,CAAjB,EAA4B,CAA5B,C;O;KANlB,C;mFASA,yB;MAAA,iB;MAA
A,0B;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,C;O;KANlB,C;mFASA,mB;MAMW,UAA
e,CApEX,iBAoEc,CApEd,MAAJ,GAoEe,CApEf,GAoEkB,C;MAAzB,OAAa,CApEF,iBAAK,GAAL,MAAJ,GAo
EM,CApEN,GAAmB,G;K;mFAuE9B,yB;MAAA,iB;MAAA,0B;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP
,EAAU,CAAV,C;O;KARlB,C;mFAWA,yB;MAAA,iB;MAAA,0B;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CA
AP,EAAU,CAAV,C;O;KARlB,C;IAWA,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAA
A,KAAV,M;QAAiB,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,
UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAxHV,MAAO,KAwHe,GAxHf,EAwH
oB,CAxHpB,C;;MAyHd,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,Q
AAA,KAAV,M;QAAiB,MAlIV,MAAO,KAkIe,GAlIf,EAkIoB,CAlIpB,C;;MAmId,OAAO,G;K;IAGX,4B;MAMc,
Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA5IV,MAAO,KA4Ie,GA5If,

EA4IoB,CA5IpB,C;;MA6Id,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAA
U,QAAA,KAAV,M;QAAuB,UAAM,G;QAAZ,MA7IN,oBA6IuB,CA7IvB,MAAJ,GAAY,GAAZ,GA6I2B,C;;MA
ClC,OAAO,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QA
AiB,MA9IV,MAAO,KA8Ie,GA9If,EA8IoB,CA9IpB,C;;MA+Id,OAAO,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C
;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA/IV,MAAO,KA+Ie,GA/If,EA+IoB,CA/IpB,C;;
MAgJd,OAAO,G;K;IAGX,wB;MAOI,OAAW,oBAAK,CAAL,MAAJ,GAAY,CAAZ,GAAmB,C;K;mFAG9B,yB;
MAkBA,iB;MAlBA,uB;QAMI,OAkBO,MAAO,KAlBC,CAkBD,EAlBY,CAkBZ,C;O;KAxBlB,C;mFASA,yB;MA
SA,iB;MATA,uB;QAMI,OASO,MAAO,KATC,CASD,EATY,CASZ,C;O;KAflB,C;mFASA,yB;MAAA,iB;MAAA
,uB;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;O;KANlB,C;mFASA,gB;MAMI,OAAW,kBAAK,CAAL,
MAAJ,GAAY,CAAZ,GAAmB,C;K;mFAG9B,yB;MAAA,iB;MAAA,uB;QAQI,OAAO,MAAO,KAAI,CAAJ,EAA
O,CAAP,C;O;KARlB,C;mFAWA,yB;MAAA,iB;MAAA,uB;QAQI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,C;
O;KARlB,C;IAWA,2B;MAOI,OAAO,SAAM,CAAN,EAAS,SAAM,CAAN,EAAS,CAAT,CAAT,C;K;mFAGX,yB
;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAM,CAAN,EAAiB,CAAjB,EAA4B,CAA5B,C;O;KANlB,C;mF
ASA,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAM,CAAN,EAAiB,CAAjB,EAA4B,CAA5B,C;O;KANl
B,C;mFASA,yB;MAAA,iB;MAAA,0B;QAMI,OAAO,MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,C;O;KA
NlB,C;mFASA,mB;MAMW,UAAe,CApEX,iBAoEc,CApEd,MAAJ,GAoEe,CApEf,GAoEkB,C;MAAzB,OAAa,C
ApEF,iBAAK,GAAL,MAAJ,GAoEM,CApEN,GAAmB,G;K;mFAuE9B,yB;MAAA,iB;MAAA,0B;QAQI,OAAO,
MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,C;O;KARlB,C;mFAWA,yB;MAAA,iB;MAAA,0B;QAQI,OAA
O,MAAO,KAAI,CAAJ,EAAO,CAAP,EAAU,CAAV,C;O;KARlB,C;IAWA,4B;MAQc,Q;MADV,UAAU,C;MAC
V,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAAM,SAAM,GAAN,EAAW,CAAX,C;;MACvB,OAA
O,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAx
HV,MAAO,KAwHe,GAxHf,EAwHoB,CAxHpB,C;;MAyHd,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;M
ACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MAlIV,MAAO,KAkIe,GAlIf,EAkIoB,CAlIpB,C;;MA
mId,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QA
AiB,MA5IV,MAAO,KA4Ie,GA5If,EA4IoB,CA5IpB,C;;MA6Id,OAAO,G;K;IAGX,4B;MAMc,Q;MADV,UAAU,C
;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAuB,UAAM,G;QAAZ,MA7IN,oBA6IuB,CA7IvB,MA
AJ,GAAY,GAAZ,GA6I2B,C;;MAClC,OAAO,G;K;IAGX,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV
,gB;QAAU,QAAA,KAAV,M;QAAiB,MA9IV,MAAO,KA8Ie,GA9If,EA8IoB,CA9IpB,C;;MA+Id,OAAO,G;K;IA
GX,4B;MAQc,Q;MADV,UAAU,C;MACV,wBAAU,KAAV,gB;QAAU,QAAA,KAAV,M;QAAiB,MA/IV,MAAO,
KA+Ie,GA/If,EA+IoB,CA/IpB,C;;MAgJd,OAAO,G;K;IkDvaX,iB;MAAA,qB;MAEI,0BAA0B,gBACtB,EADsB,E
ACd,IADc,EACN,IADM,EACE,IADF,EACU,IADV,EACkB,IADlB,EAC0B,IAD1B,EACkC,IADlC,EAC0C,IAD1
C,EACkD,IADlD,EAC0D,IAD1D,EACkE,IADlE,EAC0E,IAD1E,EACkF,IADlF,EAC0F,IAD1F,EACkG,IADlG,E
AC0G,IAD1G,EACkH,IADlH,EAC0H,IAD1H,EACkI,IADlI,EAEtB,IAFsB,EAEd,IAFc,EAEN,IAFM,EAEE,IAFF
,EAEU,IAFV,EAEkB,IAFlB,EAE0B,IAF1B,EAEkC,IAFlC,EAE0C,IAF1C,EAEkD,KAFlD,EAE0D,KAF1D,EAE
kE,KAFlE,EAE0E,KAF1E,EAEkF,KAFlF,EAE0F,KAF1F,EAEkG,KAFlG,EAE0G,KAF1G,E;K;;;IAF9B,6B;MA
AA,4B;QAAA,W;;MAAAA,qB;K;IAQA,0C;MAKI,aAAa,C;MACb,UAAU,KAAM,OAAN,GAAa,CAAb,I;MACV,a
AAa,E;MACb,YAAY,C;MACZ,OAAO,UAAU,GAAjB,C;QACI,SAAS,CAAC,SAAS,GAAT,IAAD,IAAiB,CAAj
B,I;QACT,QAAQ,MAAM,MAAN,C;QACR,IAAI,SAAS,KAAb,C;UACI,SAAS,SAAS,CAAT,I;aACR,IAAI,WA
AU,KAAd,C;UACD,OAAO,M;;UAEP,MAAM,SAAS,CAAT,I;;MAEd,OAAO,UAAc,SAAS,KAAb,GAAoB,CAA
pB,GAA2B,CAArC,K;K;IAGX,mC;MAKI,SAAS,SvCEiC,I;MuCD1C,YAAY,kBAAkB,mBAAM,mBAAxB,EAA
oC,EAApC,C;MACZ,WAAW,KAAK,mBAAM,mBAAN,CAAiB,KAAjB,CAAL,I;MACX,OAAW,OAAO,EAAX,
GAAe,IAAf,GAAyB,E;K;IAGpC,gC;MAII,OAAO,6BAAoB,C;K;IC7C/B,kB;MAAA,sB;MAEI,6B;MACA,8B;M
ACA,gC;MAKuB,UAAT,MAAS,EAAT,MAAS,EAAT,M;MAFV,eAAe,kE;MACf,iBAAiB,eAAS,GAAT,C;MAC
E,sBAAT,QAAS,C;MAAT,mB;MAAA,kB;MAAA,kB;MAAV,8C;QACI,WAAW,oBAAS,CAAT,CxC2BuB,IwC3
BlC,IAA+B,C;;MAInC,qBAAqB,sW;MACrB,WAAW,mBAAmB,cAAnB,EAAmC,UAAnC,EAA+C,GAA/C,C;M
ACX,YAAY,eAAS,IAAK,OAAd,C;MACZ,0BAAU,IAAV,e;QACI,IAAI,QAAK,CAAT,C;UAAY,MAAM,GAAN
,IAAW,KAAK,GAAL,C;;UACVlB,MAAM,GAAN,IAAW,MAAM,MAAI,CAAJ,IAAN,IAAe,KAAK,GAAL,CAAf,
I;;MAEpB,yBAAoB,K;MAGpB,kBAAkB,0U;MAClB,0BAAqB,mBAAmB,WAAnB,EAAgC,UAAhC,EAA4C,GA

A5C,C;MAGrB,oBAAoB,i8B;MACpB,4BAAuB,mBAAmB,aAAnB,EAAkC,UAAlC,EAA8C,GAA9C,C;K;;;IA7B
/B,8B;MAAA,6B;QAAA,Y;;MAAA,sB;K;IAiCA,iC;MAII,OAAO,6BAAmB,C;K;IAG9B,oC;MAIW,wCAAmB,C
;MAAnB,U;QAA6B,wBxCPM,awCON,C;;MAApC,W;K;IAGJ,oC;MAIW,wCAAmB,C;MAAnB,U;QAA6B,wBx
CdM,awCcN,C;;MAApC,W;K;IAGJ,kC;MAQI,SAAS,SxCzBiC,I;MwC0B1C,YAAY,kBAAkB,oBAAO,kBAAzB,
EAA4C,EAA5C,C;MAEZ,iBAAiB,oBAAO,kBAAP,CAAyB,KAAzB,C;MACjB,eAAe,aAAa,oBAAO,mBAAP,C
AA0B,KAA1B,CAAb,GAAgD,CAAhD,I;MACf,WAAW,oBAAO,qBAAP,CAA4B,KAA5B,C;MAEX,IAAI,KAA
K,QAAT,C;QACI,OAAO,C;;MAGX,kBAAkB,OAAS,C;MAE3B,IAAI,gBAAe,CAAnB,C;QACI,YAAY,C;QACZ,
gBAAgB,U;QAChB,aAAU,CAAV,OAAa,CAAb,M;UACI,yBAAc,QAAS,KAAV,GAAqB,GAAlC,K;UACA,IAAI
,YAAY,EAAhB,C;YACI,OAAO,C;;UAEX,gBAAS,CAAT,I;UACA,yBAAc,QAAS,KAAV,GAAqB,GAAlC,K;UA
CA,IAAI,YAAY,EAAhB,C;YACI,OAAO,C;;UAEX,gBAAS,CAAT,I;;QAEJ,OAAO,C;;MAGX,IAAI,QAAQ,CA
AZ,C;QACI,OAAO,W;;MAGX,eAAgB,KAAK,UAAL,I;MAChB,cAAgB,QAAQ,EAAZ,GAAkB,WAAW,CAAX,
IAAlB,GAAoC,Q;MAChD,OAAQ,SAAU,IAAI,OAAJ,IAAV,CAAD,GAA2B,C;K;ICnGtC,0B;MAAA,8B;MACI,
+BAA+B,gBAC3B,GAD2B,EACnB,GADmB,EACX,GADW,EACH,GADG,EACK,GADL,EACa,GADb,EACqB,
GADrB,EAC6B,IAD7B,EACqC,IADrC,EAC6C,IAD7C,EACqD,IADrD,EAC6D,IAD7D,EACqE,IADrE,EAC6E,I
AD7E,EACqF,IADrF,EAC6F,KAD7F,EACqG,KADrG,EAC6G,KAD7G,EACqH,KADrH,EAC6H,KAD7H,E;MA
G/B,gCAAgC,gBAC5B,CAD4B,EACzB,CADyB,EACtB,CADsB,EACnB,CADmB,EAChB,CADgB,EACb,CADa,
EACV,CADU,EACP,EADO,EACH,CADG,EACA,EADA,EACI,CADJ,EACO,CADP,EACU,EADV,EACc,EADd,
EACkB,EADlB,EACsB,CADtB,EACyB,CADzB,EAC4B,CAD5B,EAC+B,CAD/B,EACkC,CADlC,E;K;;;IAJpC,s
C;MAAA,qC;QAAA,oB;;MAAA,8B;K;IASA,qC;MACI,YAAY,kBAAkB,4BAAe,wBAAjC,EAAkD,SAAlD,C;M
ACZ,OAAO,SAAS,CAAT,IAAc,aAAO,4BAAe,wBAAf,CAA+B,KAA/B,IAAwC,4BAAe,yBAAf,CAAgC,KAAh
C,CAAxC,IAAP,C;K;ICXzB,qC;MACI,OAAe,IAAR,8BAAgB,IAAhB,KACY,IAAR,8BAAgB,IADpB,C;K;ICCX,
wC;MxCiBW,Q;MAAA,IwCXgB,KxCWZ,IAAS,CAAT,IwCXY,KxCWE,IAAS,2BAA3B,C;QAAA,OAAsC,qBw
CXtB,KxCWsB,C;;QwCXb,MAAM,8BAA0B,mCAAyB,gBAAzB,MAA1B,C;;MAAtC,W;K;ICRJ,sC;MAEI,WA
AW,S5CmC+B,I;M4CjC1C,IAAY,GAAR,oBAAgB,GAAhB,KAAkC,GAAR,oBAAgB,GAA1C,CAAJ,C;QACI,O
AA8B,OAAtB,KAAK,CAAC,OAAO,CAAP,IAAD,IAAa,CAAb,IAAL,KAAsB,C;;MAGlC,IAAY,IAAR,oBAAgB
,IAAhB,KAAkC,IAAR,oBAAgB,IAA1C,CAAJ,C;QACI,OAAO,S;;MAEX,OAAO,wB;K;ICPX,wC;MpCqTe,WoC
7SY,KpC6SZ,IAAS,C;MAAT,S;QAAc,OoC7SF,KpC6SE,IAygHT,gBAAR,iBAAQ,C;;MAzgHT,U;MAAA,S;QA
AA,SAAsC,sBoC7StB,KpC6SsB,C;;QoC7Sb,MAAM,8BAA0B,iCAAuB,cAAvB,MAA1B,C;;MAAtC,a;K;IAGJ,w
C;MpCsTe,WoC9SY,KpC8SZ,IAAS,C;MAAT,S;QAAc,OoC9SF,KpC8SE,IAqgHT,gBAAR,iBAAQ,C;;MArgHT,
U;MAAA,S;QAAA,SAAsC,sBoC9StB,KpC8SsB,C;;QoC9Sb,MAAM,8BAA0B,iCAAuB,cAAvB,MAA1B,C;;MA
AtC,a;K;IAGJ,wC;MpCuTe,WoC/SY,KpC+SZ,IAAS,C;MAAT,S;QAAc,OoC/SF,KpC+SE,IAigHT,gBAAR,iBAA
Q,C;;MAjgHT,U;MAAA,S;QAAA,SAAsC,sBoC/StB,KpC+SsB,C;;QoC/Sb,MAAM,8BAA0B,iCAAuB,cAAvB,M
AA1B,C;;MAAtC,a;K;IAGJ,wC;MpCwTe,WoChTY,KpCgTZ,IAAS,C;MAAT,S;QAAc,OoChTF,KpCgTE,IA6/G
T,gBAAR,iBAAQ,C;;MA7/GT,U;MAAA,S;QAAA,SAAsC,sBoChTtB,KpCgTsB,C;;QoChTb,MAAM,8BAA0B,iC
AAuB,cAAvB,MAA1B,C;;MAAtC,a;K;IASO,6C;MAAA,8B;MAAS,uB;K;8FACW,Y;MAAQ,OAAA,gBAAY,K;
K;+CAC3C,Y;MAAkC,OAAA,gBAAY,U;K;sDAC9C,mB;MAAgD,OAAA,gBAAY,gBAAS,OAAT,C;K;mDAC5
D,iB;MACI,oCAAa,2BAAkB,KAAlB,EAAyB,SAAzB,C;MACb,OAAO,6BAAY,KAAZ,C;K;qDAEX,mB;MAES,
Q;MAAL,IAAI,eAAC,0EAAD,OAAJ,C;QAAgC,OAAO,E;MACvC,OpC0rBO,UoC1rBA,gBpC0rBR,QAAQ,EoC
1rBoB,OxE0OF,KoCgdlB,C;K;yDoCxrBX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,OAAJ,C;QAAgC,OAAO,E;
MACvC,OpC66BO,coC76BA,gBpC66BR,QAAQ,EoC76BwB,OxEqON,KoCwsBlB,C;K;;IoCn8BnB,6B;MAMI,4
C;K;IA2BO,6C;MAAA,8B;MAAS,uB;K;8FACW,Y;MAAQ,OAAA,gBAAY,K;K;+CAC3C,Y;MAAkC,OAAA,gB
AAY,U;K;sDAC9C,mB;MAAiD,OAAA,gBAAY,gBAAS,OAAT,C;K;mDAC7D,iB;MACI,oCAAa,2BAAkB,KAA
lB,EAAyB,SAAzB,C;MACb,OAAO,6BAAY,KAAZ,C;K;qDAEX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,QA
AJ,C;QAAiC,OAAO,E;MACxC,OpCyqBO,UoCzqBA,gBpCyqBR,QAAQ,EoCzqBoB,OvD0NA,KmB+cpB,C;K;y
DoCvqBX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,E;MACxC,OpC45BO,coC55BA,g
BpC45BR,QAAQ,EoC55BwB,OvDqNJ,KmBusBpB,C;K;;IoCl7BnB,6B;MAMI,4C;K;IA2BO,6C;MAAA,8B;MA
AS,uB;K;8FACW,Y;MAAQ,OAAA,gBAAY,K;K;+CAC3C,Y;MAAkC,OAAA,gBAAY,U;K;sDAC9C,mB;MAAi
D,OAAA,gBAAY,gBAAS,OAAT,C;K;mDAC7D,iB;MACI,oCAAa,2BAAkB,KAAlB,EAAyB,SAAzB,C;MACb,O

AAO,6BAAY,KAAZ,C;K;qDAEX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,E;MACxC
,OpCwpBO,UoCxpBA,gBpCwpBR,QAAQ,EoCxpBoB,OzE4IA,KqC4gBpB,C;K;yDoCtpBX,mB;MAES,Q;MAAL
,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,E;MACxC,OpC24BO,coC34BA,gBpC24BR,QAAQ,EoC34BwB,O
zEuIJ,KqCowBpB,C;K;;IoCj6BnB,8B;MAMI,4C;K;IA2BO,6C;MAAA,8B;MAAS,uB;K;8FACW,Y;MAAQ,OAA
A,gBAAY,K;K;+CAC3C,Y;MAAkC,OAAA,gBAAY,U;K;sDAC9C,mB;MAAkD,OAAA,gBAAY,gBAAS,OAAT,
C;K;mDAC9D,iB;MACI,oCAAa,2BAAkB,KAAlB,EAAyB,SAAzB,C;MACb,OAAO,6BAAY,KAAZ,C;K;qDAE
X,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,SAAJ,C;QAAkC,OAAO,E;MACzC,OpCuoBO,UoCvoBA,gBpCuoB
R,QAAQ,EoCvoBoB,OvE4HE,KmC2gBtB,C;K;yDoCroBX,mB;MAES,Q;MAAL,IAAI,eAAC,0EAAD,SAAJ,C;Q
AAkC,OAAO,E;MACzC,OpC03BO,coC13BA,gBpC03BR,QAAQ,EoC13BwB,OvEuHF,KmCmwBtB,C;K;;IoCh5
BnB,8B;MAMI,4C;K;ICtIJ,qC;MAII,SAAS,S9CgCiC,I;M8C/B1C,OAAa,CAAN,gBAAc,EAAd,KACU,EAAN,gB
AAc,EADlB,KAEI,OAAM,GAFV,KAGI,KAAK,IAAL,KACC,OAAM,IAAN,KACS,IAAN,gBAAc,IADjB,KAEG
,OAAM,IAFT,IAGG,OAAM,IAHT,IAIG,OAAM,IAJT,IAKG,OAAM,IALT,IAMG,OAAM,KAPV,CAHJ,C;K;;;;m
CCTP,gB;;K;;ICAJ,wB;K;;IAIA,wB;K;;IAIA,wB;K;;IAKiC,uB;MAAC,oB;QAAA,OAA0B,E;MAA1B,gB;K;;IAE
lC,kB;K;;IAqCqC,sB;MAAC,gB;K;;IAgCN,4B;MAAC,sB;K;;IAEjC,uB;K;;IA8DmC,4B;MAAC,kB;K;;IAEpC,oB
;K;;IAmCA,+B;K;;ICvLA,oB;K;;IAIA,wB;K;;oFzDJA,qB;MAKqE,uC8BJtB,E;K;iG9BM/C,yB;MAAA,kD;MAA
A,4B;QAQsE,mBAAY,SAAZ,C;O;KARtE,C;IAUA,iC;MAGI,OAAsB,UAAY,QAAvB,KAAmC,SAA9C,GACe,U
AAY,UAD3B,GAGI,gBAAgB,UAAhB,C;K;IAGR,qC;MAEI,Y8B3B2C,E;M9B4B3C,eAAe,UAAW,W;MAC1B,
OAAO,QAAS,UAAhB,C;QACU,KAAY,MAAK,QAAS,OAAd,C;MACtB,OAAO,K;K;IAGX,8C;MAQc,Q;MAN
V,IAAI,KAAM,OAAN,GAAa,UAAW,KAA5B,C;QACI,OAAO,gBAAgB,UAAhB,C;;MAEX,eAAe,UAAW,W;M
AC1B,YAAY,C;MACZ,OAAO,QAAS,UAAhB,C;QACI,MAAM,YAAN,EAAM,oBAAN,UAAiB,QAAS,O;;MAE
9B,IAAI,QAAQ,KAAM,OAAlB,C;QACI,MAAM,KAAN,IAAe,I;;MAEnB,OAAO,K;K;IAIX,yB;MAG6C,sBAAY
,OAAZ,E;K;wGAE7C,yB;MAAA,+D;MAAA,gC;QAI0B,gBAAf,gB;QAAqB,aJU5B,W;QIVA,OJWO,SIXoC,Q;O
;KAJ/C,C;yGAOA,yB;MAAA,4E;MAAA,gE;MAAA,0C;QAII,qBAAqB,QAArB,C;QAC8B,gBAAvB,eAAa,QAA
b,C;QAA6B,aJEpC,W;QIFA,OJGO,SIH4C,Q;O;KALvD,C;IASA,wB;MAG2C,oBAAU,OAAV,E;K;sGAE3C,yB;
MAAA,uE;MAAA,gC;QAI8B,gBAAnB,oB;QAAyB,aJXhC,W;QIWA,OJVO,SIUwC,Q;O;KAJnD,C;wGAOA,yB;
MAAA,wE;MAAA,0C;QAIsC,gBAA3B,mBAAiB,QAAjB,C;QAAiC,aJlBxC,W;QIkBA,OJjBO,SIiBgD,Q;O;KAJ
3D,C;IAQA,qB;MAIuD,oBAAU,IAAV,E;K;sGAEvD,yB;MAAA,wE;MAAA,gC;QAIiC,gBAAtB,oB;QAA4B,aJh
CnC,W;QIgCA,OJ/BO,SI+B2C,Q;O;KAJtD,C;uGAOA,yB;MAAA,uE;MAAA,0C;QAIyC,gBAA9B,mBAAoB,QA
ApB,C;QAAoC,aJvC3C,W;QIuCA,OJtCO,SIsCmD,Q;O;KAJ9D,C;IAQA,mC;MAOqB,Q;MAAA,kC;MAAjB,iBA
Ac,CAAd,yB;QACI,sBAAK,KAAL,EAAc,KAAd,C;;K;IAIR,+B;MAMuD,sBAAQ,4BAAR,C;K;IAEvD,6B;MAIw
E,kBAAhB,0B;MAAwB,uB;MAAxB,OJlE7C,W;K;IIoEX,4B;MAQI,gBAAgB,SAAhB,EAAsB,cAAtB,C;K;IAGJ,
2C;MAQI,gBAAgB,SAAhB,EAAsB,UAAtB,C;K;IAGJ,2C;MACI,IAAI,IAAK,KAAL,IAAa,CAAjB,C;QAAoB,M
;MAEpB,YAAY,YAAY,IAAZ,C;MACZ,gBAAc,KAAd,EAAqB,UAArB,C;MAEA,aAAU,CAAV,MAAkB,KAA
M,OAAxB,M;QACI,iBAAK,CAAL,EAAU,MAAM,CAAN,CAAV,C;;K;IAIR,uC;MACI,OAAO,gBAAkB,IAAlB,
O;K;IAGX,iF;MAII,oCAAa,2BAAkB,UAAlB,EAA8B,QAA9B,EAAwC,MAAO,OAA/C,C;MACb,gBAAgB,WA
AW,UAAX,I;MAChB,oCAAa,2BAAkB,iBAAlB,EAAqC,oBAAoB,SAApB,IAArC,EAAoE,WAAY,OAAhF,C;M
AEb,IAAI,kBAAkB,WAAlB,KAAkC,kBAAkB,MAAlB,CAAtC,C;QACI,eAAsB,MAAY,UAAS,UAAT,EAAqB,
QAArB,C;QACtB,WAAY,KAAI,QAAJ,EAAc,iBAAd,C;;QAExB,IAAI,WAAW,WAAX,IAA0B,qBAAqB,UAAn
D,C;UACI,iBAAc,CAAd,UAAsB,SAAtB,U;YACI,YAAY,oBAAoB,KAApB,IAAZ,IAAyC,OAAO,aAAa,KAAb,I
AAP,C;;;UAG7C,mBAAc,YAAY,CAAZ,IAAd,aAAmC,CAAnC,Y;YACI,YAAY,oBAAoB,OAApB,IAAZ,IAAyC
,OAAO,aAAa,OAAb,IAAP,C;;;;K;8GAMzD,qB;MAEgF,gB;K;kGAEhF,yB;MAAA,4D;MAAA,4B;QAC8E,OAA
K,aAAL,SAAK,C;O;KADnF,C;sGAIA,gC;MAEI,OAAI,SAAJ,GAEI,SAFJ,GAII,SN63BoB,Q;K;IMz3B5B,mC;M
AEI,IAAI,QAAQ,CAAZ,C;QACI,oB;;MAEJ,OAAO,K;K;IAGX,mC;MAEI,IAAI,QAAQ,CAAZ,C;QACI,oB;;MA
EJ,OAAO,K;K;IAIX,mC;MAIqD,mB;K;IAErD,wC;MP1NI,IAAI,EOiOI,YAAY,CPjOhB,CAAJ,C;QACI,cOgOqB,
gC;QP/NrB,MAAM,gCAAyB,OAAQ,WAAjC,C;;K;IiEzB4C,qC;MAAiC,6B;K;uDAIvF,mB;MACI,qB;MACA,eA
Ae,e;MACf,OAAO,QAAS,UAAhB,C;QACI,IAAI,OAAA,QAAS,OAAT,EAAmB,OAAnB,CAAJ,C;UACI,QAAS,
S;UACT,OAAO,I;;;MAGf,OAAO,K;K;yDAGX,oB;MAGoB,Q;MAFhB,qB;MACA,eAAe,K;MACC,0B;MAAhB,
OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,eAAI,OAAJ,CAAJ,C;UAAkB,WAAW,I;;MAEjC,OAAO,Q;K;IAKuC

,sE;MAAA,qB;QAAE,OAAM,gBAAN,mB;O;K;4DAFpD,oB;MAEY,Q;MADR,qB;MACA,OAAoC,YAA5B,iEA
A4B,EAAU,oDAAV,C;K;IAKU,sE;MAAA,qB;QAAE,QAAO,gBAAP,mB;O;K;4DAFpD,oB;MAEY,Q;MADR,q
B;MACA,OAAoC,YAA5B,iEAA4B,EAAU,oDAAV,C;K;gDAGxC,Y;MACI,qB;MACA,eAAe,IAAK,W;MACpB,
OAAO,QAAS,UAAhB,C;QACI,QAAS,O;QACT,QAAS,S;;K;iDAIjB,Y;MAE8B,OAAA,IAAK,U;K;yDAGnC,Y;
K;;IC3CgD,+B;MAAiC,oC;MACjF,gBAA8B,C;K;8CAM9B,mB;MAMI,qB;MACA,iBAAI,SAAJ,EAAU,OAAV,
C;MACA,OAAO,I;K;mDAGX,2B;MAMc,UACF,M;MANR,oCAAa,4BAAmB,KAAnB,EAA0B,SAA1B,C;MAEb
,qB;MACA,aAAa,K;MACb,cAAc,K;MACJ,0B;MAAV,OAAU,cAAV,C;QAAU,mB;QACN,kBAAI,eAAJ,EAAI,u
BAAJ,WAAc,CAAd,C;QACA,UAAU,I;;MAEd,OAAO,O;K;0CAGX,Y;MACI,qB;MACA,yBAAY,CAAZ,EAAe,
SAAf,C;K;IAKiB,gE;MAAA,qB;QAAE,OAAM,gBAAN,mB;O;K;sDAFvB,oB;MACI,qB;MACA,OAAO,kBAAU
,8CAAV,C;K;IAKU,gE;MAAA,qB;QAAE,QAAO,gBAAP,mB;O;K;sDAFvB,oB;MACI,qB;MACA,OAAO,kBAA
U,8CAAV,C;K;6CAIX,Y;MAAqD,iD;K;mDAErD,mB;MAAoD,0BAAQ,OAAR,KAAoB,C;K;kDAExE,mB;MAC
qB,Q;MAAA,6B;MAAjB,iBAAc,CAAd,yB;QACI,IAAI,wBAAI,KAAJ,GAAc,OAAd,CAAJ,C;UACI,OAAO,K;;;
MAGf,OAAO,E;K;sDAGX,mB;MACI,iBAAc,sBAAd,WAA+B,CAA/B,U;QACI,IAAI,wBAAI,KAAJ,GAAc,OA
Ad,CAAJ,C;UACI,OAAO,K;;;MAGf,OAAO,E;K;iDAGX,Y;MAA6D,iCAAa,CAAb,C;K;yDAC7D,iB;MAAuE,sD
AAiB,KAAjB,C;K;oDAGvE,8B;MAA4E,uCAAQ,IAAR,EAAc,SAAd,EAAyB,OAAzB,C;K;wDAE5E,8B;MAII,e
AAe,0BAAa,SAAb,C;MACf,YAAO,UAAU,SAAV,I;M/DuDX,iBAAc,CAAd,UAAsB,KAAtB,U;Q+DtDiB,e;QAC
A,iB;;K;2CAIjB,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,2BAAJ,C;QAAuB,OAAO,K;M
AE9B,OAAO,oCAAa,uBAAc,IAAd,EAAoB,KAApB,C;K;6CAGxB,Y;MAG+B,OAAA,oCAAa,yBAAgB,IAAhB,
C;K;IAG5C,kD;MAAA,oB;MACI,eACsB,C;MACtB,cAIqB,E;K;yDAErB,Y;MAAkC,sBAAQ,gB;K;sDAE1C,Y;
MAEW,Q;MADP,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MACtB,eAAO,mBAAP,EAAO,2BAAP,O;MACA,O
AAO,wBAAI,WAAJ,C;K;wDAGX,Y;MlE5CJ,IAAI,EkE6CU,gBAAQ,ElE7ClB,CAAJ,C;QACI,ckE4CwB,sE;QlE
3CxB,MAAM,6BAAsB,OAAQ,WAA9B,C;;MkE6CF,6BAAS,WAAT,C;MACA,eAAQ,W;MACR,cAAO,E;K;;IA
OqB,6D;MAHpC,oB;MAGmD,wD;MAG3C,oCAAa,4BAAmB,KAAnB,EAA0B,WAAyB,KAAnD,C;MACb,eAA
a,K;K;iEAGjB,Y;MAAsC,sBAAQ,C;K;+DAE9C,Y;MAAgC,mB;K;8DAEhC,Y;MACI,IAAI,CAAC,kBAAL,C;Q
AAoB,MAAM,6B;MAE1B,eAAO,mCAAP,EAAO,YAAP,C;MACA,OAAO,wBAAI,WAAJ,C;K;mEAGX,Y;MA
AoC,sBAAQ,CAAR,I;K;+DAEpC,mB;MACI,wBAAI,YAAJ,EAAW,OAAX,C;MACA,mC;MACA,cAAO,E;K;+D
AGX,mB;MlElFJ,IAAI,EkEmFU,gBAAQ,ElEnFlB,CAAJ,C;QACI,ckEkFwB,4E;QlEjFxB,MAAM,6BAAsB,OAA
Q,WAA9B,C;;MkEkFF,wBAAI,WAAJ,EAAU,OAAV,C;K;;IAIgB,+D;MAAuF,8B;MAAtF,kB;MAA0C,4B;MAC/
D,eAAyB,C;MAGrB,oCAAa,2BAAkB,gBAAlB,EAA6B,OAA7B,EAAsC,WAAK,KAA3C,C;MACb,eAAa,UAAU
,gBAAV,I;K;wDAGjB,0B;MACI,oCAAa,4BAAmB,KAAnB,EAA0B,YAA1B,C;MAEb,WAAK,aAAI,mBAAY,K
AAZ,IAAJ,EAAuB,OAAvB,C;MACL,mC;K;wDAGJ,iB;MACI,oCAAa,2BAAkB,KAAlB,EAAyB,YAAzB,C;MA
Eb,OAAO,wBAAK,mBAAY,KAAZ,IAAL,C;K;6DAGX,iB;MACI,oCAAa,2BAAkB,KAAlB,EAAyB,YAAzB,C;
MAEb,aAAa,WAAK,kBAAS,mBAAY,KAAZ,IAAT,C;MAClB,mC;MACA,OAAO,M;K;wDAGX,0B;MACI,oCA
Aa,2BAAkB,KAAlB,EAAyB,YAAzB,C;MAEb,OAAO,WAAK,aAAI,mBAAY,KAAZ,IAAJ,EAAuB,OAAvB,C;K
;mGAGO,Y;MAAQ,mB;K;2DAE/B,Y;MAA+C,WAAK,iB;K;;;ICxMN,8B;MAAiC,sB;MAwCnF,uBAAoC,I;MA+
CpC,yBAA6C,I;K;IAlFR,oD;MAAC,wB;MAGlC,gBAAqB,K;K;iFAHa,Y;MAAA,yB;K;uGAKZ,Y;MAAQ,oB;K;
8DAE9B,oB;MAKI,eAAe,IAAK,S;MACpB,gBAAc,Q;MACd,OAAO,Q;K;wDAGX,Y;MAA+B,iEAAc,IAAd,C;K
;wDAC/B,Y;MAAkC,iEAAc,IAAd,C;K;sDAClC,iB;MAA4C,+DAAY,IAAZ,EAAkB,KAAlB,C;K;;IAlB5C,8E;M
AAA,wE;MAAsC,2CAAK,KAAM,IAAX,EAAgB,KAAM,MAAtB,C;MAAtC,Y;K;IAsBJ,+C;MACsE,6B;K;mEA
ClE,mB;MAAmD,kCAAc,OAAd,C;K;iEAEnD,mB;MAAiD,gCAAY,OAAZ,C;K;;yCAIrD,Y;MACI,YAAQ,Q;K;I
AOQ,+F;MAAA,sD;MAAS,6B;K;uFACb,mB;MAAwC,MAAM,qCAA8B,8BAA9B,C;K;mFAC9C,Y;MACI,4BA
AwB,Q;K;4FAG5B,mB;MAAsD,sDAY,OAAZ,C;K;IAI3C,oH;MAAA,kD;K;4GACH,Y;MAAkC,OAAA,0BAA
c,U;K;yGAChD,Y;MAAyB,OAAA,0BAAc,OAAO,I;K;2GAC9C,Y;MAAwB,0BAAc,S;K;;sFAL9C,Y;MACI,oBA
AoB,oCAAQ,W;MAC5B,6G;K;0FAOJ,mB;MACI,qB;MACA,IAAI,+CAAY,OAAZ,CAAJ,C;QACI,4BAAwB,cA
AO,OAAP,C;QACxB,OAAO,I;;MAEX,OAAO,K;K;oIAGY,Y;MAAQ,OAAA,4BAAwB,K;K;4FAEvD,Y;MAAsC
,4BAAwB,iB;K;;0FA9B1E,Y;MACI,IAAI,4BAAJ,C;QACI,6F;;MA+BJ,OAAO,mC;K;kDAKf,gB;MAEyB,Q;MA
DrB,qB;MACqB,OAAA,I5EkR2D,QAAQ,W;M4ElRxF,OAAqB,cAArB,C;QAAqB,wB;QAAf,U5EmMsD,U;Q4E
nMjD,Y5EgNiD,Y;Q4E/MxD,iBAAI,GAAJ,EAAS,KAAT,C;;K;IAQc,iG;MAAA,sD;MAAS,oC;K;yFACf,mB;MA

AwC,MAAM,qCAA8B,gCAA9B,C;K;qFAC9C,Y;MAAuB,4BAAwB,Q;K;8FAE/C,mB;MAAsD,wDAAc,OAAd,
C;K;IAI3C,sH;MAAA,kD;K;8GACH,Y;MAAkC,OAAA,0BAAc,U;K;2GAChD,Y;MAAyB,OAAA,0BAAc,OAA
O,M;K;6GAC9C,Y;MAAwB,0BAAc,S;K;;wFAL9C,Y;MACI,oBAAoB,oCAAQ,W;MAC5B,+G;K;sIAOmB,Y;M
AAQ,OAAA,4BAAwB,K;K;8FAEvD,Y;MAAsC,4BAAwB,iB;K;;4FAnB1E,Y;MACI,IAAI,8BAAJ,C;QACI,iG;;
MAoBJ,OAAO,qC;K;gDAGf,e;MACI,qB;MACA,WAAW,YAAQ,W;MACnB,OAAO,IAAK,UAAZ,C;QACI,YA
AY,IAAK,O;QACjB,QAAQ,KAAM,I;QACd,IAAI,YAAO,CAAP,CAAJ,C;UACI,YAAY,KAAM,M;UAClB,IAA
K,S;UACL,OAAO,K;;;MAGf,OAAO,I;K;kDAIX,Y;K;;IC3I+C,8B;MAAiC,oC;K;0CAEhF,iB;MAMI,IAAI,UAAU
,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,0BAAJ,C;QAAsB,OAAO,K;MAC7B,OAAO,mCAAY,mBAAU,IAAV,
EAAgB,KAAhB,C;K;4CAGvB,Y;MAG+B,OAAA,mCAAY,2BAAkB,IAAlB,C;K;;ICbT,0B;MAAuD,8B;MAAlC,
4B;MACvD,4BAAkC,K;K;gCAkBlC,Y;MAEI,qB;MACA,4BAAa,I;MACb,OAAO,I;K;qCAGX,Y;K;iDAGA,uB;
K;iFAG8B,Y;MAAQ,OAAA,oBAAM,O;K;sCAC5C,iB;MACyC,Q;MAAA,oCAAM,0BAAW,KAAX,CAAN,4D;
K;sCACzC,0B;MAIW,IAAa,I;MAHpB,qB;MACA,0BAAW,KAAX,C;MAEoB,gBAAb,qBAAM,KAAN,C;MAAq
B,qC;MAA5B,OAAO,CAAa,OlE8BjB,SkE9BI,2D;K;oCAGX,mB;MACI,qB;MACM,oBAAY,MAAK,OAAL,C;M
AClB,qC;MACA,OAAO,I;K;sCAGX,0B;MACI,qB;MACM,oBAAY,QAAO,mCAAoB,KAApB,CAAP,EAAmC,C
AAnC,EAAsC,OAAtC,C;MAClB,qC;K;yCAGJ,oB;MACI,qB;MACA,IAAI,QAAS,UAAb,C;QAAwB,OAAO,K;M
AE/B,uBAAA,oBpEioDoB,QMhrD0C,Y8D+CrD,Q9D/CqD,CNgrD1C,C;MoEhoDpB,qC;MACA,OAAO,I;K;yCA
GX,2B;MACI,qB;MACA,mCAAoB,KAApB,C;MAEA,IAAI,UAAS,SAAb,C;QAAmB,OAAO,oBAAO,QAAP,C;
MAC1B,IAAI,QAAS,UAAb,C;QAAwB,OAAO,K;MAE3B,IADE,KACF,e;QAAQ,OAAO,oBAAO,QAAP,C;WA
Cf,IAFE,KAEF,O;QAAK,uB9D5DqD,Y8D4D7C,Q9D5D6C,CNgrD1C,QoEpnD6B,oBpEonD7B,C;;QoEnnDR,uB
AAoC,cAA5B,oBAA4B,EAAV,CAAU,EAAP,KAAO,CAAY,Q9D7DE,Y8D6DK,Q9D7DL,C8D6DF,EAA4C,cA
AN,oBAAM,EAAY,KAAZ,EAAmB,SAAnB,CAA5C,C;;MAG5D,qC;MACA,OAAO,I;K;2CAGX,iB;MACI,qB;M
ACA,0BAAW,KAAX,C;MACA,qC;MACA,OAAW,UAAS,sBAAb,GACG,oBAAY,MADf,GAGG,oBAAY,QAA
O,KAAP,EAAc,CAAd,CAAlB,CAAmC,CAAnC,C;K;uCAGR,mB;MAEkB,Q;MADd,qB;MACc,2B;MAAd,mD;Q
ACI,IAAI,4BAAM,KAAN,GAAgB,OAAhB,CAAJ,C;UACU,oBAAY,QAAO,KAAP,EAAc,CAAd,C;UAClB,qC;
UACA,OAAO,I;;;MAGf,OAAO,K;K;8CAGX,8B;MACI,qB;MACA,qC;MACM,oBAAY,QAAO,SAAP,EAAkB,U
AAU,SAAV,IAAlB,C;K;gCAGtB,Y;MACI,qB;MACA,uBhChHuC,E;MgCiHvC,qC;K;wCAIJ,mB;MAA+C,OAA
M,QAAN,oBAAM,EAAQ,OAAR,C;K;4CAErD,mB;MAAmD,OAAM,YAAN,oBAAM,EAAY,OAAZ,C;K;mCAE
zD,Y;MAA0B,uBAAc,oBAAd,C;K;0CAE1B,iB;MAGe,UAGL,MAHK,EAMO,M;MAPlB,IAAI,KAAM,OAAN,G
AAa,SAAjB,C;QACI,OAAO,2D;;MAGc,gBAAxB,eAAK,SAAL,IAAK,gBAAL,yB;MpEuwBL,UAAU,SAAV,Eo
EvwBsC,KpEuwBtC,EAD+F,CAC/F,EADoH,CACpH,EADuI,gBACvI,C;MoErwBI,IAAI,KAAM,OAAN,GAAa,S
AAjB,C;QACI,MAAM,SAAN,IAAc,6E;;MAGlB,OAAO,K;K;kCAGX,Y;MACI,OAAO,EAAS,MAAM,MAAK,o
BAAL,C;K;yCAI1B,Y;MACI,IAAI,yBAAJ,C;QAAgB,MAAM,oC;K;+CAG1B,iB;MACI,oCAAa,kCAAyB,SAAz
B,C;MADoB,Y;K;wDAIrC,iB;MACI,oCAAa,mCAA0B,SAA1B,C;MAD6B,Y;K;;IAlJ9C,+B;MAAAA,mD;MAG8B
,sBhCRa,EgCQb,C;MAH9B,Y;K;IAKA,kD;MAAAA,mD;MAIkD,sBhCdP,EgCcO,C;MAJlD,Y;K;IAMA,2C;MAA
A,mD;MAGqD,sB9DLa,Y8DKR,Q9DLQ,C8DKb,C;MAHrD,Y;K;ICrBJ,0C;MACI,IAAI,6BAAJ,C;QACU,KAAY
,MAAK,UAAL,C;;QAElB,UAAU,KAAV,EAAwC,CAAxC,EAAiD,cAAN,KAAM,CAAjD,EAA4D,eAAW,UAA
X,CAA5D,C;;K;IAMiB,kD;MAAA,uB;QAAgB,OAAA,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IAFpD,4C;
MACI,IAAI,6BAAJ,C;QACI,iBAAiB,gC;QACX,KAAY,MAAK,UAAL,C;;QAElB,UAAU,KAAV,EAAwC,CAAx
C,EAAiD,cAAN,KAAM,CAAjD,EAA4D,UAA5D,C;;K;IAIR,gE;MACI,IAAI,aAAY,UAAU,CAAV,IAAZ,CAAJ,
C;QACI,UAAU,KAAV,EAAwC,SAAxC,EAAmD,UAAU,CAAV,IAAnD,EAAgE,UAAhE,C;;K;IAMiB,gC;MAA
gB,OAAE,iBAAF,CAAE,EAAU,CAAV,C;K;IAF3C,0B;MACI,IAAI,6BAAJ,C;QACI,iBAAiB,gB;QACX,KAAY,
MAAK,UAAL,C;;QAElB,UAAU,KAAV,EAAwC,CAAxC,EAAiD,cAAN,KAAM,CAAjD,EAA4D,cAA5D,C;;K;;
IAaa,kD;MAAoB,QAAC,IAAM,CAAP,KAAa,IAAM,CAAnB,K;K;IARzC,uC;MACI,sC;QAAiC,OAAjC,yB;;MA
CA,4BAA4B,K;MAE5B,YAAY,E;MAGZ,iBAAc,CAAd,UAAsB,GAAAtB,U;QAAiC,KAAY,MAAK,KAAL,C;MA
C7C,iBAAiB,kC;MACX,KAAY,MAAK,UAAL,C;MAClB,mBAAc,CAAd,YAAsB,KAAM,OAA5B,Y;QACI,QA
AQ,MAAM,UAAQ,CAAR,IAAN,C;QACR,QAAQ,MAAM,OAAAN,C;QACR,IAAI,CAAC,IAAM,CAAP,OAAc,I
AAM,CAApB,KAA0B,KAAK,CAAnC,C;UAAsC,OAAO,K;;MAEjD,4BAA4B,I;MAC5B,OAAO,I;K;IAIX,2D;M
ACI,aAAa,gBAAmB,KAAM,OAAzB,O;MACb,aAAa,YAAU,KAAV,EAAiB,MAAjB,EAAyB,KAAzB,EAAgC,Y

AAhC,EAA8C,UAA9C,C;MACb,IAAI,WAAW,KAAf,C;QACI,aAAU,KAAV,OAAiB,YAAjB,M;UAA+B,MAA
M,CAAN,IAAW,OAAO,CAAP,C;;K;IAIlD,4D;MAEI,IAAI,UAAS,GAAb,C;QACI,OAAO,K;;MAGX,aAAa,CAA
C,QAAQ,GAAR,IAAD,IAAgB,CAAhB,I;MACb,WAAW,YAAU,KAAV,EAAiB,MAAjB,EAAyB,KAAzB,EAAg
C,MAAhC,EAAwC,UAAxC,C;MACX,YAAY,YAAU,KAAV,EAAiB,MAAjB,EAAyB,SAAS,CAAT,IAAzB,EAA
qC,GAArC,EAA0C,UAA1C,C;MAEZ,aAAiB,SAAS,MAAb,GAAqB,KAArB,GAAgC,M;MAG7C,gBAAgB,K;M
AChB,iBAAiB,SAAS,CAAT,I;MACjB,aAAU,KAAV,OAAiB,GAAjB,M;QAEQ,iBAAa,MAAb,IAAuB,cAAc,GA
ArC,C;UACI,gBAAgB,KAAK,SAAL,C;UAChB,iBAAiB,MAAM,UAAN,C;UAEjB,IAAI,UAAW,SAAQ,SAAR,E
AAmB,UAAnB,CAAX,IAA6C,CAAjD,C;YACI,OAAO,CAAP,IAAY,S;YACZ,6B;;YAEA,OAAO,CAAP,IAAY,
U;YACZ,+B;;eAGR,iBAAa,MAAb,C;UACI,OAAO,CAAP,IAAY,KAAK,SAAL,C;UACZ,6B;;UAGA,OAAO,CA
AP,IAAY,MAAM,UAAN,C;UACZ,+B;;;MAMZ,OAAO,M;K;ICrGX,4C;MAMoB,UACM,M;MAHtB,IAAI,iBAA
J,C;QAAkB,OAAO,C;MACzB,aAAa,C;MACb,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAEQ,oB;UAA
mB,U;;UACnB,I5BFiC,MAAa,Y4BEnC,O5BFmC,C4BE9C,C;YAAwD,iCAAhC,OAAgC,C;iBAExD,uC;YAAmC
,2BAAR,OAAQ,C;eACnC,wC;YAAmC,2BAAR,OAAQ,C;eACnC,sC;YAAmC,2BAAR,OAAQ,C;eACnC,uC;YA
AmC,2BAAR,OAAQ,C;;YAEA,kBAAR,OAAQ,C;;QATvC,wB;QAYA,SAAS,MAAK,MAAL,QAAc,WAAd,I;;M
AEb,OAAO,M;K;;;ICTP,uC;MAAA,2C;K;2DACI,0B;MAA2D,sBAAU,MAAV,C;K;gEAE3D,iB;MAA6C,Q;MA
AA,wEAAqB,C;K;;;IAHtE,mD;MAAA,kD;QAAA,iC;;MAAA,2C;K;;;MC0BA,iC;MAKA,8B;MA6CA,0BAAmE,
I;;IAzEnE,kC;MAAA,oB;MAA+B,8C;K;2CAE3B,mB;MAAyD,MAAM,qCAA8B,iCAA9B,C;K;uCAC/D,Y;MAC
I,WAAa,Q;K;uDAGjB,mB;MAAgE,OAAA,WAAa,uBAAc,OAAd,C;K;0CAE7E,Y;MAAwE,OAAA,iCAAY,W;K
;qDAEpF,mB;MACI,IAAI,iBAAS,OAAT,CAAJ,C;QACI,WAAa,cAAO,OAAQ,IAAf,C;QACb,OAAO,I;;MAEX,
OAAO,K;K;wFAGY,Y;MAAQ,OAAA,WAAa,K;K;;8BA6ChD,Y;MACI,0BAAY,Q;K;0CAIhB,e;MAAmD,OAA
A,0BAAY,gBAAS,GAAT,C;K;4CAE/D,iB;MAAmE,gBAAZ,0B;MAAY,c;;QnEqnDnD,Q;QADhB,IAAI,wCAAs
B,mBAA1B,C;UAAqC,aAAO,K;UAAP,e;;QACrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,ImErnDm
D,uBAAS,gBnEqnD9C,OmErnDwD,MAAV,QnEqnD5D,C;YAAwB,aAAO,I;YAAP,e;;;QAC9C,aAAO,K;;;MmEt
nDgD,iB;K;kFAInD,Y;MACI,IAAI,+BAAJ,C;QACI,0BAAW,qB;;MAEf,OAAO,sC;K;uCAGf,Y;MAAgF,iC;K;kC
AEhF,e;MAA+C,OAAA,0BAAY,WAAI,GAAJ,C;K;oCAE3D,sB;MAAgD,OAAA,0BAAY,aAAI,GAAJ,EAAS,K
AAT,C;K;qCAE5D,e;MAAyC,OAAA,0BAAY,cAAO,GAAP,C;K;+EAEvB,Y;MAAQ,OAAA,0BAAY,K;K;;IA5D
lD,0C;MAAA,iD;MAAuD,8B;MAvC3D,mB;MAwCQ,8BAAmB,W;MACnB,2BAAgB,WAAY,S;MAFhC,Y;K;IA
KA,+B;MAAA,iD;MAGuB,aAAK,kEAAL,Q;MAHvB,Y;K;IAKA,4D;MAAA,iD;MAQ8D,qB;MzEpC9D,IAAI,Ey
EsCQ,mBAAmB,CzEtC3B,CAAJ,C;QACI,cyEqCgC,+C;QzEpChC,MAAM,gCAAyB,OAAQ,WAAjC,C;;MAFV,I
AAI,EyEuCQ,cAAc,CzEvCtB,CAAJ,C;QACI,gByEsC2B,yC;QzErC3B,MAAM,gCAAyB,SAAQ,WAAjC,C;;MyE
0BV,Y;K;IAcA,gD;MAAA,iD;MAA2C,eAAK,eAAL,EAAsB,GAAtB,Q;MAA3C,Y;K;IAGA,yC;MAAA,iD;MAG
8C,qB;MAC1C,KAAK,gBAAO,QAAP,C;MAJT,Y;K;IAqCJ,4B;MAK8E,gBAAnE,aAAmB,gEAAnB,C;MAA2E,
wB;MAAlF,OtEvCO,S;K;;MuEjEP,uB;;kCAyCA,mB;MACI,UAAU,gBAAI,aAAI,OAAJ,EAAa,IAAb,C;MACd,O
AAO,W;K;8BAGX,Y;MACI,gBAAI,Q;K;uCAOR,mB;MAA6D,OAAA,gBAAI,mBAAY,OAAZ,C;K;gCAEjE,Y;
MAAyC,OAAA,gBAAI,U;K;iCAE7C,Y;MAAqD,OAAA,gBAAI,KAAK,W;K;qCAE9D,mB;MAAkD,OAAA,gB
AAI,cAAO,OAAP,CAAJ,Q;K;+EAEpB,Y;MAAQ,OAAA,gBAAI,K;K;;IA5D1C,6B;MAAA,iD;MAGoB,8B;MAZ
xB,mB;MAaQ,oBAAM,gB;MAJV,Y;K;IAOA,yC;MAAA,iD;MAG2C,8B;MAnB/C,mB;MAoBQ,oBAAM,eAAgB
,QAAS,KAAzB,C;MACN,qBAAO,QAAP,C;MALJ,Y;K;IAQA,4D;MAAA,iD;MAQ2D,8B;MAhC/D,mB;MAiCQ,
oBAAM,eAAgB,eAAhB,EAAiC,UAAjC,C;MATV,Y;K;IAYA,gD;MAAA,iD;MAA2C,eAAK,eAAL,EAAsB,GA
AtB,Q;MAA3C,Y;K;IAEA,oC;MAAA,iD;MAM0C,8B;MA5C9C,mB;MA6CQ,oBAAW,G;MAPf,Y;K;IAmCJ,+B;
MAKuC,gBAA5B,eAAQ,eAAR,C;MAAoC,6B;MAA3C,OvENO,S;K;IwEzD6B,uC;MAAC,kC;MAErC,oBAAkC,
kB;MAClC,sBAAyB,C;K;2EAHY,Y;MAAAA,8B;K;2FAGrC,Y;MAAAA,0B;K,OAAA,gB;MAAAA,0B;K;gDAGA,sB
;MACI,eAAe,aAAS,qBAAY,GAAZ,C;MACxB,mBAAmB,6BAAsB,QAAAtB,C;MACnB,IAAI,oBAAJ,C;QAEI,kB
AAW,QAAAX,IAAuB,mCAAY,GAAZ,EAAAiB,KAAjB,C;;QAEvB,IAAI,6BAAJ,C;UAEI,YAA+B,Y;UAC/B,IAAI,
aAAS,gBAAO,KAAM,IAAb,EAAAkB,GAAAlB,CAAb,C;YACI,OAAO,KAAM,gBAAS,KAAT,C;;YAEb,kBAAW,
QAAX,IAAuB,CAAQ,KAAR,EAAAe,mCAAY,GAAZ,EAAAiB,KAAjB,CAAAf,C;YACvB,6B;YACA,OAAO,I;;;UAI
X,YAAuC,Y;UACvC,cAAkB,wBAAN,KAAAM,EAAAiB,GAAAjB,C;UAClB,IAAI,eAAJ,C;YACI,OAAO,OAAM,gB
AAS,KAAT,C;;UAEX,KAAY,MAAK,mCAAY,GAAZ,EAAAiB,KAAjB,CAAAL,C;;;MAG1B,6B;MAEA,OAAO,I;

K;iDAGX,e;MAE0D,IAAnC,I;MADnB,eAAe,aAAS,qBAAY,GAAZ,C;MACL,oCAAsB,QAAtB,C;MAAA,iB;QA
AmC,OAAO,I;;MAA7D,mBAAmB,I;MACnB,IAAI,6BAAJ,C;QACI,YAAgC,Y;QAChC,IAAI,aAAS,gBAAO,KA
AM,IAAb,EAAkB,GAAlB,CAAb,C;U9BzDR,O8B0D6B,iB9B1DhB,C8B0D4B,Q9B1D5B,C;U8B2DD,6B;UACA
,OAAO,KAAM,M;;UAEb,OAAO,I;;;QAGX,YAAuC,Y;QACvC,8BAAc,KAAd,iB;UACI,cAAY,MAAM,KAAN,C
;UACZ,IAAI,aAAS,gBAAO,GAAP,EAAY,OAAM,IAAlB,CAAb,C;YACI,IAAI,KAAM,OAAN,KAAc,CAAlB,C;
cACU,KAAN,UAA2B,C;c9BtE/C,O8BwEqC,iB9BxExB,C8BwEoC,Q9BxEpC,C;;c8B2Ea,KAAY,QAAO,KAAP,
EAAc,CAAd,C;;YAEtB,6B;YAEA,OAAO,OAAM,M;;;;MAIzB,OAAO,I;K;0CAGX,Y;MACI,oBAAa,kB;MACb,
YAAO,C;K;mDAGX,e;MAAyC,uBAAS,GAAT,S;K;8CAEzC,e;MAA+B,Q;MAAA,+BAAS,GAAT,8B;K;+CAE/
B,e;MAC2E,IAApD,I;MAAA,oCAAsB,aAAS,qBAAY,GAAZ,CAA/B,C;MAAA,iB;QAAoD,OAAO,I;;MAA9E,m
BAAmB,I;MACnB,IAAI,6BAAJ,C;QACI,YAAgC,Y;QAChC,IAAI,aAAS,gBAAO,KAAM,IAAb,EAAkB,GAAlB,
CAAb,C;UACI,OAAO,K;;UAEP,OAAO,I;;;QAGX,YAAuC,Y;QACvC,OAAa,wBAAN,KAAM,EAAiB,GAAjB,C
;;K;uDAIrB,0B;MACI,sB;;Q1FsoCY,Q;QAAhB,iD;UAAgB,cAAhB,e;UAAsB,I0FtoCK,aAAS,gB1FsoCA,O0FtoC
a,IAAb,M1FsoCd,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;M0FvoCH,yB;K;IAIO,8E;MAAAA,wD;M
ACH,aAAY,E;MAEZ,YAA0B,MAAa,MAAK,qCAAL,C;MACvC,gBAAe,E;MAEf,oBAA4B,I;MAC5B,eAAc,K;
MACd,iBAAgB,E;MAChB,iBAAqC,I;K;yEAErC,Y;MACI,IAAI,6BAAwB,YAA5B,C;QACI,gBAAqB,iBAAqD,
O;QAC1E,IAAI,4DAAc,SAAlB,C;UACI,OAAO,C;;MAGf,IAAI,yDAAa,SAAK,OAAtB,C;QACI,oBAAe,2CAA
W,UAAK,aAAL,CAAX,C;QACf,eAAU,iC;QACV,iBAAY,C;QACZ,OAAO,C;;QAEP,oBAAe,I;QACf,OAAO,C;;
K;mEAIf,Y;MACI,IAAI,eAAS,EAAb,C;QACI,aAAQ,oB;MACZ,OAAO,eAAS,C;K;gEAGpB,Y;MAEoB,Q;MAD
hB,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MACN,IAAI,YAAJ,C;QACZ,yBAAqD,cAArD,C;;QAEa,OAAb,iB;
;MAHJ,oB;MAKA,iBAAiB,S;MACjB,aAAQ,E;MACR,OAAO,S;K;kEAGX,Y;M3E/CR,I2EgDyB,c3EhDrB,QAA
J,C;QACI,cAhByB,0B;QAiBzB,MAAM,6BAAsB,OAAQ,WAA9B,C;;M2E+CE,6BAAyB,cAAO,6BAAY,IAAnB,
C;MACzB,iBAAY,I;MAEZ,uC;K;;6CAtDZ,Y;MAEI,2D;K;4DAyDJ,oB;MACI,mBAAmB,kBAAW,QAAX,C;MA
CnB,OAAW,iBAAiB,SAArB,GAAgC,IAAhC,GAA0C,Y;K;;;;wCCtKrD,Y;MACI,aAAgB,MAAM,OAAT,CAAiB
,IAAjB,C;MAEb,OAAO,KAAP,IAAgB,C;M/BXpB,O+BYqB,M/BZR,C+BYgB,K/BZhB,C;M+BaT,OAAO,M;K;;
ICNuB,qC;MAAC,kC;MAEnC,oBAAkC,kB;MAClC,sBAAyB,C;K;yEAHU,Y;MAAA,8B;K;yFAGnC,Y;MAAA,0
B;K,OAAA,gB;MAAA,0B;K;iDAWA,e;MACI,IAAI,0BAAJ,C;QAAoB,OAAO,K;MAC3B,OAAO,kBAAW,GAA
X,MAAoB,S;K;4CAG/B,e;MACI,IAAI,0BAAJ,C;QAAoB,OAAO,I;MAC3B,YAAY,kBAAW,GAAX,C;MACZ,O
AAW,UAAU,SAArB,GAAgC,KAAhC,GAA2D,I;K;8CAI/D,sB;M7EVA,IAAI,E6EWQ,uB7EXR,CAAJ,C;QACI,c
Ada,qB;QAeb,MAAM,gCAAyB,OAAQ,WAAjC,C;;M6EUN,eAAe,kBAAW,GAAX,C;MACf,kBAAW,GAAX,IA
AkB,K;MAElB,IAAI,aAAa,SAAjB,C;QACI,6B;QAEA,OAAO,I;;QAGP,OAAO,Q;;K;+CAIf,e;MACI,IAAI,0BAA
J,C;QAAoB,OAAO,I;MAC3B,YAAY,kBAAW,GAAX,C;MACZ,IAAI,UAAU,SAAd,C;QhCnDJ,OgCoDyB,iBhC
pDZ,CgCoDwB,GhCpDxB,C;QgCqDL,6B;QAEA,OAAO,K;;QAGP,OAAO,I;;K;wCAKf,Y;MACI,oBAAa,kB;MA
Cb,YAAO,C;K;IAKA,0E;MAAAA,oD;MACH,cAAkC,MAAa,MAAK,mCAAL,C;MAC/C,kBAA4B,qBAAL,WAA
K,C;MAC5B,iBAA+B,I;K;iEAE/B,Y;MAAkC,OAAAA,eAAS,U;K;8DAE3C,Y;MAIuB,gB;MAHnB,UAAU,eAAS,
O;MACnB,iBAAU,G;MAES,+E;MAAnB,OAAO,iD;K;gEAGX,Y;MAEkC,UAA9B,M;MAAAA,oC;MAA8B,YAAa
,c;M7EchD,uB;MAeP,IAfoB,KAehB,QAAJ,C;QACI,cAhByB,0B;QAiBzB,MAAM,6BAAsB,OAAQ,WAA9B,C;;
QAEN,sBAnBgB,K;;M6Ede,oBAAO,sFAAP,C;K;;2CAjBnC,Y;MACI,yD;K;IAqBkD,0F;MAAAA,8B;MAAAA,oD;K
;kHAC9B,Y;MAAQ,uB;K;oHACN,Y;MAAQ,6CAAuB,gBAAvB,C;K;2EAE9B,oB;MAAwC,OAAAA,2BAAuB,aA
AI,gBAAJ,EAAS,QAAT,C;K;qEAE/D,Y;MAA+B,OAAAA,mCAAY,uBAAc,IAAd,C;K;qEAC3C,Y;MAAkC,OAA
A,mCAAY,uBAAc,IAAd,C;K;mEAC9C,iB;MAA4C,OAAAA,mCAAY,qBAAY,IAAZ,EAAkB,KAAlB,C;K;;gDAR
5D,e;MAAsD,iE;K;;;MCItD,sBAOsC,I;MA6CtC,yB;MAOA,4BAAkC,K;;IArIE,sD;MAZpC,oB;MAYyD,0CAAq
C,GAArC,EAA0C,KAA1C,C;MACrD,oBAAuC,I;MACvC,oBAAuC,I;K;wDAEvC,oB;MACI,WAAmB,iB;MACn
B,OAAa,mEAAS,QAAT,C;K;;IAIrB,wC;MAAAA,oB;MAA+B,8C;K;IAE3B,sD;MAAAA,oB;MACI,cACsC,I;MAEt
C,cACsC,I;MAGlC,cAAO,iC;K;6DAIX,Y;MACI,OAAO,gBAAS,I;K;0DAGpB,Y;MAEI,IAAI,CAAC,cAAL,C;Q
AAgB,MAAM,6B;MAEtB,cAAc,0B;MACd,cAAO,O;MACa,gBAAb,OAAQ,a;;MAAf,c3E0DS,S2E1DoB,KAAO,
iC3E0DzC,GAAqB,SAArB,GAA+B,I;M2EzD1B,OAAO,O;K;4DAGX,Y;M9EwBR,IAAI,E8EvBc,eAAQ,I9EuBtB
,CAAJ,C;QACI,cAdW,e;QAeX,MAAM,6BAAsB,OAAQ,WAA9B,C;;M8ExBE,WAAc,iB;MAGP,oCAAP,0BAA
O,C;MACP,gCAAI,cAAO,0BAAO,IAAd,C;MAEJ,cAAO,I;K;;iDAIf,mB;MAAyD,MAAM,qCAA8B,iCAA9B,C;

K;6CAC/D,Y;MACI,WAAmB,Q;K;6DAGvB,mB;MAAgE,OAAA,WAAmB,uBAAc,OAAd,C;K;gDAEnF,Y;MA
AwE,qD;K;2DAExE,mB;MACI,qB;MACA,IAAI,iBAAS,OAAT,CAAJ,C;QACI,WAAmB,cAAO,OAAQ,IAAf,C;
QACnB,OAAO,I;;MAEX,OAAO,K;K;8FAGY,Y;MAAQ,OAAA,WAAmB,K;K;sDAElD,Y;MAAsC,WAAmB,iB;
K;;iDAa7D,qB;M9ErBA,IAAI,E8E0BM,0BAAQ,IAAR,IAAgB,0BAAQ,I9E1B9B,CAAJ,C;QACI,cAdW,e;QAeX,
MAAM,6BAAsB,OAAQ,WAA9B,C;;M8E0BN,YAAY,mB;MACZ,IAAI,SAAS,IAAb,C;QACI,sBAAO,S;QACP,y
BAAO,S;QACP,yBAAO,S;;QAGK,YAAa,KAAM,a;Q9ElBhC,uB;QAeP,IAfoB,KAehB,QAAJ,C;UACI,gBAhByB
,0B;UAiBzB,MAAM,6BAAsB,SAAQ,WAA9B,C;;UAEN,sBAnBgB,K;;Q8EkBZ,+B;QAEA,yBAAO,K;QACP,yB
AAO,K;QAEP,qBAAa,S;QACb,qBAAa,S;;K;+CAIrB,qB;MAII,IAAI,SAAK,aAAL,KAAc,SAAlB,C;QAEI,sBAA
O,I;;QAEP,IAAI,wBAAS,SAAb,C;UAEI,sBAAO,sB;;QAEX,qDAAc,sB;QACd,qDAAc,sB;;MAElB,yBAAO,I;M
ACP,yBAAO,I;K;oCA8CX,Y;MAEI,qB;MACA,4BAAa,I;MACb,OAAO,I;K;oCAGX,Y;MACI,qB;MACA,kBAAI
,Q;MACJ,sBAAO,I;K;gDASX,e;MAAmD,OAAA,kBAAI,mBAAY,GAAZ,C;K;kDAEvD,iB;MACyC,IAAR,I;MA
AA,0B;MAAA,iB;QAAQ,OAAO,K;;MAA5C,WAA6B,I;;QAEzB,IAAI,OAAA,IAAK,MAAL,EAAc,KAAd,CAAJ
,C;UACI,OAAO,I;;QAEX,OAAO,cAAA,IAAK,aAAL,C;;MACF,iBAAS,mBAAT,C;MACT,OAAO,K;K;6CAIX,
Y;MAAoF,uC;K;wCAEpF,e;MAAmD,Q;MAAJ,QAAI,OAAJ,kBAAI,WAAI,GAAJ,CAAJ,6B;K;0CAE/C,sB;MA
CI,qB;MAEA,UAAU,kBAAI,WAAI,GAAJ,C;MACd,IAAI,OAAO,IAAX,C;QACI,eAAe,mCAAW,GAAX,EAAg
B,KAAhB,C;QACf,kBAAI,aAAI,GAAJ,EAAS,QAAT,C;QACK,wBAAT,QAAS,C;QACT,OAAO,I;;QAEP,OAA
O,GAAI,gBAAS,KAAT,C;;K;2CAInB,e;MACI,qB;MAEA,YAAY,kBAAI,cAAO,GAAP,C;MAChB,IAAI,SAAS,I
AAb,C;QACU,sBAAN,KAAM,C;QACN,OAAO,KAAM,M;;MAEjB,OAAO,I;K;qFAGmB,Y;MAAQ,OAAAA,kBA
AI,K;K;6CAE1C,Y;MACI,IAAI,yBAAJ,C;QAAgB,MAAM,oC;K;;IAnG1B,mC;MAAA,uD;MAGuB,qB;MA9J3B,
yB;MA+JQ,sBAAM,gB;MAJV,Y;K;IAOA,iD;MAAA,uD;MAAoD,qB;MAlKxD,yB;MAoKc,Q;MAAN,sBAAM,+
D;MAFV,Y;K;IAKA,kE;MAAA,uD;MAQ8D,eAAM,eAAN,EAAuB,UAAvB,Q;MA/KlE,yB;MAgLQ,sBAAM,gB
;MATV,Y;K;IAYA,sD;MAAA,uD;MAA2C,qBAAK,eAAL,EAAsB,GAAtB,Q;MAA3C,Y;K;IAEA,+C;MAAA,uD
;MAG2C,qB;MAxL/C,yB;MAyLQ,sBAAM,gB;MACN,KAAK,gBAAO,QAAP,C;MALT,Y;K;IA6EJ,kC;MAKwD
,gBAA7C,qBAAyB,eAAzB,C;MAAqD,wB;MAA5D,O3EjMO,S;K;;;oC4EvCP,Y;MAEK,Q;MAA8B,CAA9B,2EA
A8B,S;MAC/B,OAAO,I;K;6CAGX,Y;MAA+C,gBAAI,iB;K;;IAhCnD,wC;MAAA,uD;MAAmD,eAAM,GAAN,Q;
MAPvD,yB;MAOI,Y;K;IAEA,qC;MAAA,uD;MAGuB,eAAM,oBAAN,Q;MAZ3B,yB;MASI,Y;K;IAKA,+C;MAA
A,uD;MAG8C,eAAM,oBAAN,Q;MAjBlD,yB;MAkBQ,qBAAO,QAAP,C;MAJJ,Y;K;IAOA,kE;MAAA,uD;MAQ8
D,eAAM,qBAAsB,eAAtB,EAAuC,UAAvC,CAAN,Q;MA7BlE,yB;MAqBI,Y;K;IAUA,sD;MAAA,uD;MAA2C,qB
AAK,eAAL,EAAsB,GAAtB,Q;MAA3C,Y;K;IAgBJ,qC;MAKmD,gBAAxC,mBAAc,qBAAd,C;MAAgD,6B;MAA
vD,O5EoBO,S;K;;;;kF6EzEX,uB;MAQI,OAAO,O;K;ICXX,sB;K;mCACI,Y;MACI,mBAAM,IAAN,C;K;2CAGJ,
mB;MACI,mBAAM,OAAN,C;MACA,c;K;iCAKJ,Y;K;;IAKuB,oC;MAA8B,qB;MAA7B,gC;K;2CACxB,mB;MA
EI,oBA+DyC,OA/Dd,OA+Dc,C;MA9DzC,iBAAa,OAAM,aAAN,C;K;;IAIrB,8B;MAEoC,qB;K;iDAChC,mB;MA
CI,OAAQ,KAAI,OAAJ,C;K;mDAGZ,mB;MACI,OAAQ,KAAI,OAAJ,C;K;2CAGZ,Y;MACI,OAAQ,KAAI,EAAJ
,C;K;;IAIhB,0B;MAEqC,qB;MACjC,cAAa,E;K;6CAEb,mB;MACI,eAoCyC,OApCxB,OAoCwB,C;K;qCAjC7C,Y
;MACI,cAAS,E;K;;IAIjB,sC;MAE4C,yB;K;yDACxC,mB;MACI,QAwByC,OAxB1B,OAwB0B,C;MAvBzC,QAA
Q,CpEqJoF,aoErJhE,IpEqJgE,EoErJ1D,CpEqJ0D,C;MoEpJ5F,IAAI,KAAK,CAAT,C;QACI,4BAAU,CpEwL0E,W
oExL9D,CpEwL8D,EoExL3D,CpEwL2D,C;QoEvLpF,Y;QACA,IAAI,CpEmLiE,WoEnLrD,IAAI,CAAJ,IpEmLqD
,C;;MoEjLzE,4BAAU,C;K;iDAGd,Y;MACI,OAAQ,KAAI,WAAJ,C;MACR,cAAS,E;K;;;IAWjB,yB;MACiD,cAA
a,KAAb,C;K;IAEjD,mB;MAEI,MAAO,U;K;IAGX,4B;MAEI,MAAO,iBAAQ,OAAR,C;K;IAGX,wB;MAEI,MAA
O,eAAM,OAAN,C;K;IAGX,kB;MACqC,MAAM,qCAA8B,sCAA9B,C;K;IAE3C,wB;MAC4C,MAAM,qCAA8B,4
CAA9B,C;K;IClGlD,mD;MACI,0B;MASA,gBAA2B,a;K;2FAFvB,Y;MAAQ,OAAA,eAAS,Q;K;oDAIrB,kB;MA
CI,UAAU,IAAK,S;MAEX,YAAQ,2CAAR,C;QACI,gBAAc,MAAO,M;WAEzB,YAAQ,yBAAR,C;QACI,gBAAc,
yC;QACd,eAAS,oBAAW,MAAX,C;;QAEL,MAAM,6BAAsB,iBAAtB,C;K;4CAItB,Y;MAOW,Q;MALP,IAAI,kB
AAW,2CAAf,C;QACI,gBAAS,yB;QACT,OAAO,yB;;MAEX,aAAa,IAAK,S;MAEd,eAAW,yCAAX,C;QAAsB,gC
;WACtB,0C;QAA4B,MAAM,MAAO,U;;QACjC,a;MAHZ,W;K;;IA7BJ,gD;MAAA,0D;MACyD,6BAAK,QAAL,E
AAe,2CAAf,C;MADzD,Y;K;;;;;ICRA,2C;MAAA,+D;MAAAuB,iC;MAF3B,iC;MAEI,Y;K;IACA,sD;MAAA,+D;M
AAuC,6BAAM,OAAN,Q;MAH3C,iC;MAGI,Y;K;IACA,6D;MAAA,+D;MAAAmD,kCAAM,OAAN,EAAAe,KAAAf,
C;MAJvD,iC;MAII,Y;K;IACA,oD;MAAAA,+D;MAAAiC,6BAAM,KAAN,Q;MALrC,iC;MAKI,Y;K;I1C4CJ,yE;MA

SI,sC;MAAA,4C;K;IATJ,iGAWY,Y;MAAQ,2B;KAXpB,E;IAAA,0DAaQ,kB;MACI,wBAAW,MAAX,C;K;IAdZ,
sF;I2C5C2E,0C;M5CkKhE,Q;MADP,e4ChKA,M5CgKA,C;MACO,Q4CjKP,M5CiKO,+D;M4ChKX,W;K;;+FCuH
A,gB;MACI,aAAa,IAAO,MAAP,E;MACb,KAAK,MAAL,C;MACA,OAAO,M;K;wFC3HX,yB;MAAA,uD;MAA
A,wC;QAWqG,OAAK,cAAL,SAAK,EAAiB,IAAjB,EAAuB,IAAvB,C;O;KAX1G,C;wFAaA,yB;MAAA,uD;MAA
A,wC;QAWoG,OAAK,cAAL,SAAK,EAAiB,IAAjB,EAAuB,IAAvB,C;O;KAXzG,C;8ECbA,yB;MAAA,6C;MAA
A,sC;QAOyD,OAAK,SAAL,SAAK,EAAY,QAAZ,C;O;KAP9D,C;8EASA,yB;MAAA,6C;MAAA,wC;QAWkE,O
AAK,SAAL,SAAK,EAAa,UAAb,S;O;KAXvE,C;oFAaA,yB;MAAA,mD;MAAA,wC;QAWqE,OAAK,YAAL,SA
AK,EAAgB,UAAhB,S;O;KAX1E,C;kFCZI,yB;MAAA,iD;MAAA,4B;QAAe,OAAK,WAAL,SAAK,C;O;KAApB,
C;wFAYA,yB;MAAA,uD;MAAA,4B;QAAe,OAAK,cAAL,SAAK,C;O;KAApB,C;IC5BJ,gC;MAAoE,gCAAqB,O
AArB,C;K;IAElC,uC;MAAC,wB;K;iDAC/B,iB;MACI,eAAQ,KAAR,C;K;8CAGJ,Y;MAAyC,iCAAuB,cAAvB,M;
K;;ICCO,6C;MAAA,8B;MAAS,uB;K;8FAClC,Y;MAAQ,OAAA,gBAAY,O;K;mDAE3C,iB;MACI,IADoC,KACp
C,IAAG,CAAH,IADoC,KACpC,IAAM,sBAAN,C;QAD8B,OACX,gBAAY,MAAK,KAAL,C;;QACvB,MAAM,8B
AA0B,WAAQ,KAAR,6BAAmC,sBAAnC,MAA1B,C;K;;IARtB,8B;MAGoD,4C;K;wECFpD,yB;MAAA,uC;MAA
A,4B;QAOsC,MAAL,SAAK,C;O;KAPtC,C;kFASA,yB;MAAA,iD;MAAA,kC;QAWuD,OAAK,WAAL,SAAK,EA
Ac,IAAd,C;O;KAX5D,C;+ECfA,qB;MAI8C,gB;K;iFAE9C,qB;MAIsE,OAAK,S;K;kFAE3E,qB;MAMyE,gB;K;IA
EzE,6B;MAiBa,UAPF,M;MAFP,QAAc,S;MAGV,cAAK,UAAL,U;QACI,mBAAK,UAAL,G;;QACJ,IjDzBqC,MA
Aa,YiDyBvC,CjDzBuC,CiDyBlD,C;UAC6B,8BAAzB,CAAyB,C;;UAGN,UAAlB,uDAAkB,Y;;MAP3B,a;K;IC7B
6D,gD;K;;ICDjE,2B;MAEI,MAAM,yBAAqB,OAArB,C;K;IAGV,sB;MAEI,MAAM,uBAAmB,cAAnB,C;K;IAGV
,2B;MAEI,MAAM,6BAAsB,OAAtB,C;K;IAGV,iC;MAEI,MAAM,4CAAqC,uBAAqB,YAArB,8BAArC,C;K;IClB
V,8B;MC8CW,kBzGqBiD,oB;MyGM9C,Q;MAAA,OAAK,0B;MAAf,OAAU,cAAV,C;QAAU,mB;QACN,UAAU,
sBAAM,CAAN,C;QACV,kBAAkB,sBAAY,GAAZ,C;QAkFiD,U;QAjFnE,WzGyKJ,ayGzKgB,GzGyKhB,EwG5O
oB,CCmEkC,uBAAuB,CAAC,WAAY,mBAAY,GAAZ,CAiFhD,GDpJrC,CCoJqC,GAA6B,UAjFjC,WAiFiC,6DD
pJnD,IAAM,CAAN,IxG4OpB,C;;MwG5OA,OCqEO,W;K;IC3EyB,oC;MAAC,oC;K;;;;;iF/CFrC,kD;MAyDI,SAA
Y,MAAK,OAAL,EAAc,SAAd,EAAyB,OAAzB,C;K;iFCzDhB,iC;MAuBmC,0B;QAAA,aAAuD,S;MACtF,SAAY,
MAAK,UAAL,C;K;;;;I+CoBhB,qB;MAK0B,Q;MADtB,UAAmB,E;MACnB,wBAAsB,KAAtB,gB;QAAsB,aAAA,
KAAtB,M;QAAK,IAAC,0BAAD,EAAO,2B;QACR,IAAI,IAAJ,IAAY,K;;MAEhB,OAAO,G;K;IAGX,+B;MAMg
B,Q;MADZ,WAA0B,MAAa,MAAK,KAAL,C;MACvC,wBAAY,IAAZ,gB;QAAY,UAAA,IAAZ,M;QACI,IAAU,
KAAY,gBAAe,GAAf,CAAtB,C;UACI,UAAK,GAAL,IAAY,MAAM,GAAN,C;;;MAGpB,OAAO,S;K;qEC5DX,y
B;MAAA,iB;MAAA,oB;QAOkD,OAAA,MAAW,KAAI,CAAJ,C;O;KAP7D,C;qEASA,yB;MAAA,iB;MAAA,oB;
QAOkD,OAAA,MAAW,KAAI,CAAJ,C;O;KAP7D,C;qEASA,yB;MAAA,iB;MAAA,oB;QAOkD,OAAA,MAAW,
KAAI,CAAJ,C;O;KAP7D,C;uEASA,yB;MAAA,iB;MAAA,oB;QASmD,OAAA,MAAW,MAAK,CAAL,C;O;KAT
9D,C;uEAWA,yB;MAAA,iB;MAAA,oB;QASmD,OAAA,MAAW,MAAK,CAAL,C;O;KAT9D,C;uEAWA,yB;M
AAA,iB;MAAA,oB;QASmD,OAAA,MAAW,MAAK,CAAL,C;O;KAT9D,C;yEAWA,yB;MAAA,iB;MAAA,uB;Q
AkB+D,OAAA,MAAW,OAAM,CAAN,EAAS,CAAT,C;O;KAlB1E,C;uEAoBA,yB;MAAA,0B;MAAA,oB;QAU
mD,kBAAW,CAAX,C;O;KAVnD,C;uEAYA,yB;MAAA,0B;MAAA,oB;QASmD,kBAAW,CAAX,C;O;KATnD,C;
uEAWA,yB;MAAA,0B;MAAA,oB;QAUmD,kBAAW,CAAX,C;O;KAVnD,C;yEAYA,yB;MAAA,4B;MAAA,oB;
QAYoD,mBAAY,CAAZ,C;O;KAZpD,C;yEAcA,yB;MAAA,4B;MAAA,oB;QAYoD,mBAAY,CAAZ,C;O;KAZpD
,C;yEAcA,yB;MAAA,4B;MAAA,oB;QAaoD,mBAAY,CAAZ,C;O;KAbpD,C;yEAeA,yB;MAAA,4B;MAAA,uB;
QAS+D,mBAAY,CAAZ,EAAe,CAAf,C;O;KAT/D,C;uEAWA,yB;MAAA,iB;MAAA,oB;QAQmD,OAAA,MAA
W,MAAK,CAAL,C;O;KAR9D,C;qEAUA,yB;MAAA,iB;MAAA,oB;QAUkD,OAAA,MAAW,KAAI,CAAJ,C;O;K
AV7D,C;yEAYA,yB;MAAA,4B;MAAA,oB;QAcoD,mBAAY,CAAZ,C;O;KAdpD,C;IAgBA,sB;MAcI,IAAI,QAA
Q,GAAR,IAAe,SAAQ,GAA3B,C;QAAgC,OAAO,wCAAO,I;MAC9C,OAAO,IAAW,KAAI,CAAJ,CAAX,GAAo
B,IAAW,KAAI,IAAJ,C;K;mEAG1C,yB;MAAA,iB;MAAA,oB;QAWiD,OAAA,MAAW,KAAI,CAAJ,C;O;KAX5
D,C;yEAaA,yB;MAAA,4B;MAAA,oB;QAOoD,mBAAY,CAAZ,C;O;KAPpD,C;uEASA,yB;MAAA,0B;MAAA,o
B;QAOmD,kBAAW,CAAX,C;O;KAPnD,C;uEASA,yB;MAAA,4B;MAAA,oB;QAgBmD,mBAAY,CAAZ,C;O;K
AhBnD,C;uEAkBA,yB;MAAA,iB;MAAA,oB;QAUmD,OAAA,MAAW,MAAK,CAAL,C;O;KAV9D,C;yEAYA,y
B;MAAA,iB;MAAA,oB;QAUoD,OAAA,MAAW,OAAM,CAAN,C;O;KAV/D,C;+EAYA,yB;MAAA,4B;MAAA,
oB;QAUuD,mBAAY,CAAZ,C;O;KAVvD,C;IAYA,kB;MAQI,IAAI,IAAI,GAAJ,KAAW,GAAf,C;QACI,OAAO,I

AAW,OAAM,CAAN,C;;MAEtB,YAzBgD,MAAW,OAyBzC,CAzByC,C;MA0B3D,OAAW,QAAQ,CAAR,KAAa,
GAAxB,GAA6B,KAA7B,GAtC+C,MAAW,MAsCb,CAtCa,C;K;qEAyC9D,yB;MAAA,iB;MAAA,oB;QAUkD,O
AAA,MAAW,KAAI,CAAJ,C;O;KAV7D,C;uEAYA,yB;MAAA,0B;MAAA,oB;QAWmD,kBAAW,CAAX,C;O;K
AXnD,C;wEAcA,yB;MAAA,iB;MAAA,uB;QAO6D,OAAA,MAAW,KAAI,CAAJ,EAAO,CAAP,C;O;KAPxE,C;w
EASA,yB;MAAA,iB;MAAA,uB;QAO6D,OAAA,MAAW,KAAI,CAAJ,EAAO,CAAP,C;O;KAPxE,C;uEAUA,yB;
MAAA,iB;MAAA,oB;QAamD,OAAA,MAAW,MAAK,CAAL,C;O;KAb9D,C;qEAkBA,yB;MAAA,iB;MAAA,+B
;QAayD,OAAA,MAAW,KAAI,SAAJ,EAAU,CAAV,C;O;KAbpE,C;uEAeA,yB;MAAA,iB;MAAA,+B;QAOsD,O
AAA,MAAW,KAAI,SAAJ,EAAY,CAAZ,C;O;KAPjE,C;iGAmBsD,yB;MAAA,iB;MAAA,4B;QAAQ,OAAA,MA
AW,KAAI,SAAJ,C;O;KAAnB,C;+EAaT,yB;MAAA,0B;MAAA,4B;QAAQ,kBAAW,SAAX,C;O;KAAR,C;iFAE7
C,yB;MAAA,6C;MAAA,kC;QAK8D,OAAK,SAAL,SAAK,EAAc,IAAd,C;O;KALnE,C;IAkBqC,4B;MACjC,gBA
AO,CAAP,C;QADyC,OACrB,QAAP,CAAC,SAAM,C;WACpB,IAAK,QAAL,SAAK,CAAL,IAAgB,cAAQ,wCA
AO,kBAA/B,C;QAFyC,OAEW,S;WACpD,kBAAQ,wCAAO,UAAf,C;QAHyC,OAGb,YAAY,SAAL,SAAK,C;;Q
AHC,OAI5B,OAAL,SAAK,CAAL,GAAgB,S;K;IAG5B,2B;MAKI,IAAK,QAAL,SAAK,CAAL,IAAgB,cAAQ,wC
AAO,kBAA/B,C;QADwC,OACY,S;WACpD,kBAAQ,GAAR,C;QAFwC,OAEzB,wCAAO,U;;QACP,WAAc,UAA
L,SAAK,CAAL,yBAAuB,YAAO,CAAX,GAAc,CAAd,GAAqB,EAAxC,E;QAHgB,OpDjc6B,MAAa,gBAAe,IAA
f,C;;K;IoDuctF,6B;MAKI,IAAK,QAAL,SAAK,CAAL,IAAgB,cAAQ,wCAAO,kBAA/B,C;QAD0C,OACU,S;WA
CpD,kBAAQ,GAAR,C;QAF0C,OAE3B,CAAC,wCAAO,U;;QACR,WAAc,UAAL,SAAK,CAAL,yBAAuB,YAAO
,CAAX,GAAc,EAAd,GAAsB,CAAzC,E;QAHkB,OpD3c2B,MAAa,gBAAe,IAAf,C;;K;IoDkdtF,oC;MAUI,IAAK,
QAAL,SAAK,CAAL,IAAmB,QAAH,EAAG,CAAnB,C;QADuD,OACzB,wCAAO,I;WACrC,WAAM,SAAN,C;Q
AFuD,OAEzC,E;WACd,SAAK,SAAL,C;QAHuD,OAGrC,OAAL,SAAK,C;;QAHqC,OAI1B,SAAL,SAAK,C;K;I
AIjC,+B;MAYI,uB;QAAW,MAAM,gCAAyB,yBAAzB,C;WACjB,gBAAO,UAAP,C;QAFyC,OAEjB,U;WACxB,
gBAAO,WAAP,C;QAHyC,OAGjB,W;;QAHiB,OAIV,YAAvB,IAAW,OAAM,SAAN,CAAY,C;K;IAGnC,gC;MA
YI,uB;QAAW,MAAM,gCAAyB,yBAAzB,C;WACjB,oD;QAF2C,+B;WAG3C,oD;QAH2C,+B;;QAAA,OAIZ,uB
AAvB,IAAW,OAAM,SAAN,CAAY,C;K;uEASnC,yB;MAAA,iB;MAAA,oB;QAOgD,OAAA,MAA6B,KAAZ,CA
AY,C;O;KAP7E,C;uEASA,yB;MAAA,iB;MAAA,oB;QAOgD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAP7E,C;uEA
SA,yB;MAAA,iB;MAAA,oB;QAOgD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAP7E,C;yEASA,yB;MAAA,iB;MA
AA,oB;QASiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAT/E,C;yEAWA,yB;MAAA,iB;MAAA,oB;QASiD,OAAA,
MAA8B,MAAZ,CAAY,C;O;KAT/E,C;yEAWA,yB;MAAA,iB;MAAA,oB;QASiD,OAAA,MAA8B,MAAZ,CAAY
,C;O;KAT/E,C;2EAWA,yB;MAAA,iB;MAAA,uB;QAkB4D,OAAA,MAA6C,OAA1B,CAA0B,EAAZ,CAAY,C;O
;KAlBzG,C;yEAoBA,yB;MAAA,0B;MAAA,oB;QAUiD,OAAAyB,WAAZ,CAAY,C;O;KAV1E,C;yEAYA,yB;MA
AA,0B;MAAA,oB;QASiD,OAAyB,WAAZ,CAAY,C;O;KAT1E,C;yEAWA,yB;MAAA,0B;MAAA,oB;QAUiD,O
AAyB,WAAZ,CAAY,C;O;KAV1E,C;2EAYA,yB;MAAA,4B;MAAA,oB;QAYkD,OAA0B,YAAZ,CAAY,C;O;K
AZ5E,C;2EAcA,yB;MAAA,4B;MAAA,oB;QAYkD,OAA0B,YAAZ,CAAY,C;O;KAZ5E,C;2EAcA,yB;MAAA,4B
;MAAA,oB;QAakD,OAA0B,YAAZ,CAAY,C;O;KAb5E,C;2EAeA,yB;MAAA,4B;MAAA,uB;QAS4D,OAAwC,Y
AA1B,CAA0B,EAAZ,CAAY,C;O;KATpG,C;yEAWA,yB;MAAA,iB;MAAA,oB;QAQiD,OAAA,MAA8B,MAAZ,
CAAY,C;O;KAR/E,C;uEAUA,yB;MAAA,iB;MAAA,oB;QAUgD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAV7E,C;
2EAYA,yB;MAAA,4B;MAAA,oB;QAckD,OAA0B,YAAZ,CAAY,C;O;KAd5E,C;uEAgBA,yB;MAAA,mC;MAA
A,0B;QAc6D,OAAmC,IAA7B,CAA6B,EAAZ,IAAY,C;O;KAdhG,C;qEAgBA,yB;MAAA,iB;MAAA,oB;QAW+C
,OAAA,MAA6B,KAAZ,CAAY,C;O;KAX5E,C;2EAaA,yB;MAAA,4B;MAAA,oB;QAOkD,OAA0B,YAAZ,CAA
Y,C;O;KAP5E,C;yEASA,yB;MAAA,0B;MAAA,oB;QAOiD,OAAyB,WAAZ,CAAY,C;O;KAP1E,C;yEASA,yB;
MAAA,4B;MAAA,oB;QAgBiD,OAA0B,YAAZ,CAAY,C;O;KAhB3E,C;yEAkBA,yB;MAAA,iB;MAAA,oB;QAU
iD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAV/E,C;2EAYA,yB;MAAA,iB;MAAA,oB;QAUkD,OAAA,MAA+B,O
AAZ,CAAY,C;O;KAVjF,C;iFAYA,yB;MA5hBA,4B;MA4hBA,oB;QAUqD,OA5hBE,YA4hBS,CA5hBT,C;O;KA
khBvD,C;2EAYA,yB;MAAA,uC;MAAA,oB;QAQkD,OAAoB,MAAZ,CAAY,C;O;KARtE,C;uEAWA,yB;MAAA,
iB;MAAA,oB;QAUgD,OAAA,MAA6B,KAAZ,CAAY,C;O;KAV7E,C;yEAYA,yB;MAAA,0B;MAAA,oB;QAWi
D,OAAyB,WAAZ,CAAY,C;O;KAX1E,C;wEAeA,yB;MAAA,iB;MAAA,uB;QAO0D,OAAA,MAAW,KAAI,CAA
J,EAAO,CAAP,C;O;KAPrE,C;wEASA,yB;MAAA,iB;MAAA,uB;QAO0D,OAAA,MAAW,KAAI,CAAJ,EAAO,C
AAP,C;O;KAPrE,C;yEAUA,yB;MAAA,iB;MAAA,oB;QAaiD,OAAA,MAA8B,MAAZ,CAAY,C;O;KAb/E,C;sEA

mBA,yB;MAAA,iB;MAAA,+B;QAasD,OAAA,MAA8C,KAA1B,SAA0B,EAAZ,CAAY,C;O;KAbpG,C;uEAeA,y
B;MAAA,iB;MAAA,+B;QAOoD,OAAA,MAA8C,KAA1B,SAA0B,EAAZ,CAAY,C;O;KAPlG,C;kGAmBoD,yB;
MAAA,iB;MAAA,4B;QAAQ,OAAA,MAAgC,KAAZ,SAAY,C;O;KAAxC,C;gFAaT,yB;MAAA,0B;MAAA,4B;Q
AAQ,OAA4B,WAAZ,SAAY,C;O;KAApC,C;gFAE3C,yB;MAAA,6C;MAAA,kC;QAO8D,OAA0C,SAArC,SAAq
C,EAAZ,IAAY,C;O;KAPxG,C;iFASA,yB;MAAA,6C;MAAA,kC;QAK4D,OAA0C,SAArC,SAAqC,EAAZ,IAAY,
C;O;KALtG,C;oFAQA,yB;MAAA,iD;MAAA,4B;QAYmD,OAAW,WAAX,SAAW,C;O;KAZ9D,C;sFAcA,yB;M
AAA,mD;MAAA,4B;QAYqD,OAAW,YAAX,SAAW,C;O;KAZhE,C;IAoBA,kB;MAUqC,OAAI,IAAI,CAAR,GA
AY,CAAC,CAAD,OAAM,CAAlB,GAA0B,C;K;wEAE/D,yB;MAAA,iB;MAAA,uB;QAKoD,OAAA,MAAW,KA
AI,CAAJ,EAAO,CAAP,C;O;KAL/D,C;wEAOA,yB;MAAA,iB;MAAA,uB;QAKoD,OAAA,MAAW,KAAI,CAAJ,
EAAO,CAAP,C;O;KAL/D,C;mGAiBgD,yB;MAAA,mC;MAAA,4B;QAAQ,WAAI,SAAJ,C;O;KAAR,C;IAShB,+
B;MAC5B,gBAAO,CAAP,C;QADoC,OACxB,E;WACZ,gBAAO,CAAP,C;QAFoC,OAExB,C;;QAFwB,OAG5B,C
;K;IAKZ,kB;MASuC,OAAI,eAAI,CAAR,GAAY,CAAD,aAAX,GAAmB,C;K;wEAE1D,gB;MAKuD,OAAI,kBA
AK,CAAL,MAAJ,GAAY,CAAZ,GAAmB,C;K;wEAE1E,gB;MAKuD,OAAI,kBAAK,CAAL,MAAJ,GAAY,CAA
Z,GAAmB,C;K;mGAYxB,yB;MAAA,mC;MAAA,4B;QAAQ,WAAI,SAAJ,C;O;KAAR,C;IASjB,+B;MAC7B,2B
AAO,CAAP,C;QADqC,OACzB,E;WACZ,2BAAO,CAAP,C;QAFqC,OAEzB,C;;QAFyB,OAG7B,C;K;IC5mCZ,4
B;MAI4C,qBAAQ,S;K;IAEpD,4B;MAI2C,qBAAQ,S;K;IAEnD,+B;MAGiD,qBAAQ,wCAAO,kBAAf,IAAoC,cA
AQ,wCAAO,kB;K;IAEpG,iC;MAGgD,qBAAQ,uCAAM,kBAAd,IAAmC,cAAQ,uCAAM,kB;K;IAEjG,6B;MAG+
C,QAAC,qBAAD,IAAiB,CAAC,kB;K;IAEjE,+B;MAG8C,QAAC,uBAAD,IAAiB,CAAC,kB;K;IAGhE,iC;MAOI,
QAAQ,S;MACR,IAAI,CAAC,IAAM,UAAP,KAAsB,CAAE,KAAK,CAAP,GAAc,UAApC,K;MACJ,IAAI,CAAC,
IAAM,SAAP,KAAsB,CAAE,KAAK,CAAP,GAAc,SAApC,K;MACJ,IAAI,CAAC,IAAM,SAAP,KAAsB,CAAE,K
AAK,CAAP,GAAc,SAApC,K;MACJ,IAAI,CAAC,IAAM,QAAP,KAAsB,CAAE,KAAK,CAAP,GAAc,QAApC,K;
MACJ,IAAI,CAAC,IAAM,KAAP,KAAsB,CAAE,KAAK,EAA7B,K;MACJ,OAAO,C;K;kGAGX,yB;MAAA,4B;
MAAA,4B;QAM2D,mBAAY,SAAZ,C;O;KAN3D,C;IAQA,0C;MAOI,YATuD,YASvB,EAAf,aAAQ,CAAC,SAA
D,IAAR,CAAe,CATuB,CASvD,I;K;IAEJ,sC;MAOI,OAAI,cAAQ,CAAZ,GAAe,CAAf,GAAsB,CAAE,IAAI,EAA
J,GAlB+B,sB;K;IAoB3D,qC;MAQI,oBAAS,CAAC,SAAD,IAAT,C;K;IAEJ,yC;MAaI,oBAAI,QAAJ,GAAiB,cAA
K,EAAL,GAAqB,Q;K;IAG1C,0C;MAaI,oBAAI,EAAJ,GAAoB,QAApB,GAAiC,cAAK,Q;K;IAG1C,mC;MAMI,O
AAK,arDhEmD,uBqDgEnD,CAAL,GAA0B,arDjE6B,sBqDiE7B,CAA1B,I;K;IAEJ,2C;MAMU,WAAW,SrDxEuC,
c;MqDyEpD,e;QADJ,OACS,KA7E8C,YrDGA,sBqDHA,CA6E9C,I;;QADT,OA5EuD,YA8E3C,IA9E2C,C;;K;IAi
F3D,4C;MAMU,UAAU,SrDpFuC,a;MqDqFnD,c;QADJ,OACS,KAAqB,sBrDpF0B,uBqDoF1B,CAArB,I;;QADT,
OAEgB,sBAAJ,GAAI,C;K;IAGpB,wC;MAOU,WAAW,SrD/FuC,c;MqDgGpD,e;QAAK,UAAS,kBrDjGqC,sBqDi
GrC,C;QADlB,OrDjG4C,MAAa,KAAK,UAAS,GAAT,EqDkGvB,CrDlGuB,C;;QqDmGlD,aAAa,kBAAL,IAAK,C
;QAFzB,OrDjG4C,MAAa,KAAK,UqDmG7C,CrDnG6C,EAAc,MAAd,C;;K;IqDsGlE,uC;MAOU,UAAU,SrD5Gu
C,a;MqD6GnD,c;QAAK,WAAa,iBrD5GkC,uBqD4GlC,C;QADtB,OrD7G4C,MAAa,KAAK,UqD8GhD,CrD9GgD,
EAAc,IAAd,C;;QqD+GlD,YAAS,iBAAJ,GAAI,C;QAFrB,OrD7G4C,MAAa,KAAK,UAAS,KAAT,EqD+GrB,CrD
/GqB,C;;K;IqDkHlE,2C;MAaI,IAAI,CAAC,WAAa,EAAd,MAAqB,CAAzB,C;QACI,UAAU,SrD/HyC,a;QqDgInD
,WAAW,SrD/HyC,c;QqDgIpD,aAAa,GAAI,IAAI,QAAR,GAAqB,IAAK,MAAK,CAAC,QAAD,IAAL,C;QACvC,
cAAc,IAAK,IAAI,QAAT,GAAsB,GAAI,MAAK,CAAC,QAAD,IAAL,C;QACxC,OAAW,CAAC,WAAa,EAAd,M
AAqB,CAAhC,GrDpIwC,MAAa,KAAK,UqDoIlB,MrDpIkB,EqDoIV,OrDpIU,CqDoI1D,GrDpIwC,MAAa,KAAK,
UqDoIS,OrDpIT,EqDoIkB,MrDpIlB,C;;QqDsInD,Q;QAAA,IAAI,CAAC,WAAa,EAAd,MAAqB,CAAzB,C;UAA
A,OAA4B,S;;uBrDpIiB,uB;UqDoIP,arDrIM,sB;UqDqI5C,OrDtIiC,MAAa,KAAK,kBAAc,MAAd,C;;QqDsI1D,W;;
K;kFAKR,yB;MAAA,4C;MAAA,sC;QAaiE,6BAAW,CAAC,QAAD,IAAX,C;O;KAbjE,C;qECvKA,kC;MAII,OA
AO,SAA8B,MAAK,WAAL,C;K;uEAGzC,8C;MAII,OAAO,SAA8B,MAAK,WAAL,EAAkB,UAAlB,C;K;ICtCzC,
iC;MACI,gBAAW,IAAI,OAAO,EAAG,GAAE,IAAI,IAAI,CAAC,CAAD,EAAI,EAAJ,CAA5B,GAAuC,CAA9C,
C;K;;;IAKJ,sC;MACI,cAAO,QAAP,GAAkB,QAAQ,Q;K;ICP9B,yC;K;;IAWA,+B;K;;4GAYA,yB;MAAA,gC;MA
AA,yD;MAAA,sC;QAQI,OAAK,qBAAL,SAAK,iB;O;KART,C;ICPI,2B;MAAS,Q;MAAD,OAAwB,CAAvB,iEA
AuB,Q;K;IAMhC,+B;MAAQ,iBAAU,SAAV,C;K;;;;;;;ICtB+B,4B;MACvC,8B;K;gEAAA,Y;MAAA,4B;K;2FAII,
Y;MtGO4B,MAAM,yB;K;kCsGLtC,iB;MACI,OAAO,oCAA0B,oBAAU,KAAM,OAAhB,C;K;oCAGrC,Y;MAC+
B,gB;MAAA,8FAA0B,C;K;oCAEzD,Y;MAEI,OAAO,oBAAQ,eAAR,C;K;;IAIyB,kC;MAAuB,sBAAc,MAAd,C;

MACL,Q;MAAtD,4BAAmC,CAAmB,OAAZ,MAAY,WAAnB,kC;K;8FAAnC,Y;MAAA,gC;K;oDAEA,iB;MAC
W,cAAgB,W;MAAvB,OpEoEuD,MAAa,QoEpEpD,KpEoEoD,EAAY,OAAZ,C;K;;IoEhEjC,0E;MAIvC,sBAAc,M
AAd,C;MAFA,wC;MACA,8C;K;2CAEA,iB;MACI,IAAI,0CAAJ,C;QAAsC,OAAO,K;MAC7C,OAAa,uCAAO,K
AAP,CAAN,IAAuB,+BAAmB,KAAM,kBAAzB,C;K;iGAGD,Y;MAAQ,6B;K;uDAEzC,iB;MACI,OAAO,0BAA
mB,KAAnB,C;K;;IAIf,6B;MAAA,iC;MAAoC,sBAAoB,MAApB,C;MAChC,4BAAkC,S;K;+FAAlC,Y;MAAA,gC
;K;qDAEA,iB;MAAgD,Y;K;2FAG5C,Y;MAAQ,MAAM,qCAA8B,6CAA9B,C;K;yCAElB,iB;MAA4C,iBAAU,I;
K;2CAEtD,Y;MAA+B,Q;K;;;IAVnC,yC;MAAA,wC;QAAA,uB;;MAAA,iC;K;IAaA,uB;K;yFACqC,Y;MzG0EY,
MAAM,6ByG1EJ,oCzG0EkC,WAA9B,C;K;4FyGzEf,Y;MzGyES,MAAM,6ByGzED,uCzGyE+B,WAA9B,C;K;+
CyGvEnD,iB;MzGuE6C,MAAM,6ByGvEG,uCzGuE2B,WAA9B,C;K;mCyGrEnD,iB;MAA4C,iBAAU,I;K;qCAEt
D,Y;MAA+B,Q;K;;oHCnE/B,qB;MAAQ,2B;K;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;ICKZ,gE;MAMI,qBAAU,UAAV,EAAgC,OA
AV,WAAU,CAAhC,EAA0C,gBAA1C,C;K;IAEJ,8B;MAC2C,iC;K;IAE3C,kC;MAC+C,qBAAU,cAAA,KAAM,W
AAN,CAAV,EAA8B,KAAM,UAApC,EAA+C,IAA/C,C;K;IAE/C,2D;MAM0B,IAAN,I;MAAAA,QAAM,QAAN,C;
aACZ,I;;UAAA,K;aACA,K;;UAAA,K;;;UAFY,K;;MAAhB,oB;MAMA,OAAO,uBAAmB,IAAnB,EAAqC,OAAZ,
WAAY,CAArC,EAA+C,SAA/C,EAA0D,KAA1D,C;K;IAGX,kC;MAEI,OAAA,uCAAgB,K;K;IAEpB,8C;MAEI,O
AAA,uCAAgB,mBAAU,IAAV,C;K;IAEpB,8C;MAEI,OAAA,uCAAgB,mBAAU,IAAV,C;K;IAEpB,kD;MAEI,O
AAA,uCAAgB,uBAAc,IAAd,C;K;IC/CI,8D;MACpB,sC;MACA,sC;MACA,kD;K;mEAFA,Y;MAAA,gC;K;kEAC
A,Y;MAAA,+B;K;yEACA,Y;MAAA,sC;K;iCAEA,iB;MACI,0CACQ,wBAAc,KAAM,WAApB,CADR,IAC0C,uB
AAa,KAAM,UAAnB,CAD1C,IAC0E,0BAAoB,KAAM,iB;K;mCAExG,Y;MACI,SAAC,CAAW,SAAX,eAAW,C
AAX,GAAwB,EAAxB,QAAuC,SAAV,cAAU,CAAvC,IAAD,IAAsD,EAAtD,QAA4E,SAAjB,qBAAiB,CAA5E,I;
K;mCAEJ,Y;MACkB,UACO,M;MADrB,aAAc,2D;MAEV,cAAU,IAAV,C;QAA6B,SAAX,eAAW,W;WAC7B,IA
AA,MAAO,WAAP,S;QAAoC,SAAP,MAAO,W;;QAC5B,+B;MAHZ,2B;MAMA,WACQ,cAAU,UAAd,GAAyB,E
AAzB,GACe,eAAV,cAAU,EAAa,IAAb,EAAmB,GAAnB,EAAwB,GAAxB,C;MACnB,eAAmB,qBAAJ,GAAsB,
GAAtB,GAA+B,E;MAE9C,OAAO,iBAAiB,IAAjB,GAAwB,Q;K;;IAIvC,wB;MAAA,4B;MACI,4BAAwC,I;MAC
xC,2BAAgD,W;MAChD,kCAAyC,K;K;0FAFzC,Y;MAAA,gC;K;yFACA,Y;MAAA,+B;K;gGACA,Y;MAAA,sC;
K;sCACA,Y;MAAkC,gB;K;;;IAJtC,oC;MAAA,mC;QAAA,kB;;MAAA,4B;K;IC7BsC,oE;MAClC,0B;MACA,wC;
MACA,kC;MACA,oC;K;sEAHA,Y;MAAA,0B;K;6EACA,Y;MAAA,iC;K;0EACA,Y;MAAA,8B;K;2EACA,Y;M
AAA,+B;K;4CAEA,Y;MAAkC,gB;K;;8CANtC,Y;MACI,gB;K;8CADJ,Y;MAEI,uB;K;8CAFJ,Y;MAGI,oB;K;8C
AHJ,Y;MAII,qB;K;gDAJJ,kD;MAAA,8BACI,kCADJ,EAEI,uDAFJ,EAGI,8CAHJ,EAII,iDAJJ,C;K;4CAAA,Y;M
AAA,c;MACI,qD;MACA,4D;MACA,yD;MACA,0D;MAJJ,a;K;0CAAA,iB;MAAA,4IACI,oCADJ,IAEI,kDAFJ,IA
GI,4CAHJ,IAII,8CAJJ,I;K;ICAA,4B;MAAA,gC;MAEI,gBACe,wBAAoB,MAApB,EAA6D,KAA7D,EAAoE,gCA
ApE,C;MAEf,mBACkB,wBAAoB,MAApB,EAAgE,QAAhE,EAA0E,mCAA1E,C;MAElB,oBACmB,+B;MAEnB,
oBACmB,wBAAoB,OAApB,EAAkE,SAAlE,EAA6E,oCAA7E,C;MAEnB,iBACgB,wBAAoB,MAApB,EAA8D,M
AA9D,EAAsE,iCAAtE,C;MAEhB,kBACiB,wBAAoB,MAApB,EAA+D,OAA/D,EAAwE,kCAAxE,C;MAEjB,gB
ACe,wBAAoB,MAApB,EAA6D,KAA7D,EAAoE,gCAApE,C;MAEf,kBACiB,wBAAoB,MAApB,EAA+D,OAA/
D,EAAwE,kCAAxE,C;MAEjB,mBACkB,wBAAoB,MAApB,EAAgE,QAAhE,EAA0E,mCAA1E,C;MAElB,kBAC
iB,wBAAoB,KAApB,EAAiE,OAAjE,EAA0E,kCAA1E,C;MAEjB,mBACkB,wBAAoB,MAApB,EAAgE,QAAhE,
EAA0E,mCAA1E,C;MAElB,sBACqB,wBAAoB,KAApB,EAAkE,WAAlE,EAA+E,sCAA/E,C;MAErB,yBACwB,
wBAAoB,KAApB,EAAqE,cAArE,EAAqF,yCAArF,C;MAExB,sBACqB,wBAAoB,WAApB,EAAwE,WAAxE,EA
AqF,sCAArF,C;MAErB,sBACqB,wBAAoB,SAApB,EAAsE,WAAtE,EAAmF,sCAAnF,C;MAErB,uBACsB,wBA
AoB,UAApB,EAAwE,YAAxE,EAAsF,uCAAtF,C;MAEtB,qBACoB,wBAAoB,UAApB,EAAsE,UAAtE,EAAkF,q
CAAlF,C;MAEpB,sBACqB,wBAAoB,KAApB,EAAkE,WAAlE,EAA+E,sCAA/E,C;MAErB,uBACsB,wBAAoB,Y
AApB,EAA0E,YAA1E,EAAwF,uCAAxF,C;MAEtB,wBACuB,wBAAoB,YAApB,EAA2E,aAA3E,EAA0F,wCAA
1F,C;K;IAMkB,qE;MAAA,qB;QAAE,O1E/DD,O0E+DU,EAAT,KAAiB,UAAjB,IAAkC,EAAY,OAAf,KAA0B,a;
O;K;+CAJpG,iB;MAE2B,Q;MAAhB,U;MAAAA,KAAgB,OAAhB,eAAgB,CAAI,KAAJ,CAAhB,U;QAAAA,a;;QAC
H,aAAa,wBAAoB,QAApB,EAA+D,kBAA/D,EACoB,mDADpB,C;QAEG,eAAhB,UAAqC,M;QAHlC,SAIH,M;;
MAJJ,a;K;IA7D+E,8C;MAAE,6B;K;IAGO,iD;MAAE,0B;K;IAME,kD;MAAE,8B;K;IAGZ,+C;MAAE,6B;K;IAG
C,gD;MAAE,6B;K;IAGR,8C;MAAE,6B;K;IAGI,gD;MAAE,6B;K;IAGC,iD;MAAE,6B;K;IAGH,gD;MAAE,yB;K
;IAGD,iD;MAAE,6B;K;IAGM,oD;MAAE,mC;K;IAGO,uD;MAAE,gC;K;IAGL,oD;MAAE,6B;K;IAGJ,oD;MAA

E,6B;K;IAGE,qD;MAAE,8B;K;IAGR,mD;MAAE,4B;K;IAGJ,oD;MAAE,6B;K;IAGQ,qD;MAAE,8B;K;IAGC,sD;
MAAE,+B;K;;;IA5DvH,wC;MAAA,uC;QAAA,sB;;MAAA,gC;K;;ICCA,2B;MAEW,Q;MAAA,IAAI,KAAY,SAA
Q,MAAR,CAAhB,C;QACH,kBAAW,MAAX,C;;QAEA,kBAAW,MAAX,C;;MAHJ,W;K;IAOJ,8B;MAC4E,QAA
M,QAAS,OAAf,C;aACxE,C;UADwE,OACnE,WAAW,SAAS,CAAT,CAAX,C;aACL,C;UAFwE,OAEnE,+B;;UA
FmE,OAGhE,iB;;K;IAGZ,oC;MAEU,IAAN,I;MAAA,Q3EhB0C,O2EgB3B,CAAf,C;aACI,Q;UAA6B,OAAjB,8B
AAiB,Y;UAA7B,K;aACA,Q;UAAY,OAAI,CAAY,ClEbhC,GkEamC,CAAf,MAAkC,CAAtC,GAAyC,8BAAiB,S
AA1D,GAAwE,8BAAiB,Y;UAArG,K;aACA,S;UAA8B,OAAjB,8BAAiB,a;UAA9B,K;aACA,U;UAA+B,OAAjB,
8BAAiB,eAAgB,CAAY,OAA5B,C;UAA/B,K;;UAGQ,6B;YAAsC,OAAjB,8BAAiB,kB;eACtC,0B;YAAmC,OAA
jB,8BAAiB,e;eACnC,0B;YAAmC,OAAjB,8BAAiB,e;eACnC,2B;YAAoC,OAAjB,8BAAiB,gB;eACpC,yB;YAAk
C,OAAjB,8BAAiB,c;eAClC,0B;YAAmC,OAAjB,8BAAiB,e;eACnC,2B;YAAoC,OAAjB,8BAAiB,gB;eACpC,4B;
YAAqC,OAAjB,8BAAiB,iB;eACrC,6B;;eACA,sB;YAAkC,OAAjB,8BAAiB,W;;YAE9B,kBAAkB,MAAa,gBAAe
,CAAf,CAAkB,Y;YAE7C,oBAAgB,MAAhB,C;cAAiD,OAAjB,8BAAiB,S;iBACjD,oBAAgB,KAAhB,C;cAAgD,
OAAjB,8BAAiB,e;;cAE5C,cAA0B,W;cAC1B,kBAAW,OAAX,C;;;;UAxBxB,K;;MAAA,W;K;IAgCJ,4B;MAMW,
Q;MAJP,IAAI,WAAW,MAAf,C;QAA6B,OAAO,8BAAiB,Y;;MAErD,eAAsB,MAAY,W;MAE3B,IAAI,gBAAJ,C
;QACH,IAAI,QAAS,SAAT,QAAJ,C;UACI,aAAa,qBAAiB,MAAjB,C;UACb,oBAAsB,M;UACtB,a;;UAES,OAAT
,QAAS,S;;;QAGb,4BAAiB,MAAjB,C;;MATJ,W;K;ICrCJ,0B;MAII,sBAAY,C;K;qEAchB,4B;MAIkE,iBAAY,KA
AZ,C;K;2EAElE,qB;MAI8D,gB;K;IClDb,2C;MAC7C,qBAAwC,Q;K;iDAExC,Y;MACkC,IAAf,I;MAAA,yB;MA
AA,iB;QAAe,MAAM,6BAAsB,0CAAtB,C;;MAApC,eAAe,I;MACf,qBAAc,I;MACd,OAAO,QAAS,W;K;;;;;ICLa,
kD;MADrC,e;MACsC,0B;MAAyB,gB;MAD/D,iB;MAAA,uB;K;IAAA,mC;MAAA,sC;O;MAEI,qEAGW,CAHX,
EAGc,IAHd,C;MAKA,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,iFA
GiB,CAHjB,EAGoB,IAHpB,C;MAKA,+EAGgB,CAHhB,EAGmB,IAHnB,C;MAKA,yEAGa,CAHb,EAGgB,IAHh
B,C;MAKA,iFAGiB,CAHjB,EAGoB,IAHpB,C;MAKA,6EAGe,CAHf,EAGkB,IAHlB,C;MAKA,6FAGuB,CAHvB
,EAG0B,IAH1B,C;MAKA,yFAGqB,CAHrB,EAGwB,IAHxB,C;MAKA,4EAGc,EAHd,EAGkB,IAHlB,C;MAKA,
0EAGa,EAHb,EAGiB,IAHjB,C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB,C;MAKA,8EAGe,EAHf,EAGmB,IAHn
B,C;MAKA,wFAGoB,EAHpB,EAGwB,IAHxB,C;MAKA,gEAGQ,EAHR,EAGY,IAHZ,C;MAKA,8DAGO,EAHP
,EAGW,IAHX,C;MAKA,wEAGY,EAHZ,EAGgB,IAHhB,C;MAKA,oEAGU,EAHV,EAGc,IAHd,C;MAKA,kFAG
iB,EAHjB,EAGqB,IAHrB,C;MAKA,oFAGkB,EAHlB,EAGsB,IAHtB,C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB
,C;MAKA,4FAGsB,EAHtB,EAG0B,IAH1B,C;MAKA,oFAGkB,EAHlB,EAGsB,IAHtB,C;MAKA,wEAGY,EAHZ
,EAGgB,IAHhB,C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB,C;MAKA,gFAGgB,EAHhB,EAGoB,IAHpB,C;MA
KA,0EAGa,EAHb,EAGiB,IAHjB,C;MAKA,oGAG0B,EAH1B,EAG8B,IAH9B,C;MAKA,gGAGwB,EAHxB,EAG
4B,IAH5B,C;MAUA,oC;K;;IA3JA,+C;MAAA,yB;MAAA,uC;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,qD;
MAAA,yB;MAAA,6C;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,iD;MA
AA,yB;MAAA,yC;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,mD;MAAA,yB;MAAA,2C;K;;IAKA,2D;MAA
A,yB;MAAA,mD;K;;IAKA,yD;MAAA,yB;MAAA,iD;K;;IAKA,kD;MAAA,yB;MAAA,0C;K;;IAKA,iD;MAAA,y
B;MAAA,yC;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,mD;MAAA,yB;MAAA,2C;K;;IAKA,wD;MAAA,yB;
MAAA,gD;K;;IAKA,4C;MAAA,yB;MAAA,oC;K;;IAKA,2C;MAAA,yB;MAAA,mC;K;;IAKA,gD;MAAA,yB;M
AAA,wC;K;;IAKA,8C;MAAA,yB;MAAA,sC;K;;IAKA,qD;MAAA,yB;MAAA,6C;K;;IAKA,sD;MAAA,yB;MAA
A,8C;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,0D;MAAA,yB;MAAA,kD;K;;IAKA,sD;MAAA,yB;MAAA,8
C;K;;IAKA,gD;MAAA,yB;MAAA,wC;K;;IAKA,oD;MAAA,yB;MAAA,4C;K;;IAKA,oD;MAAA,yB;MAAA,4C;
K;;IAKA,iD;MAAA,yB;MAAA,yC;K;;IAKA,8D;MAAA,yB;MAAA,sD;K;;IAKA,4D;MAAA,yB;MAAA,oD;K;8
CAKA,gB;MAG2D,OAAK,iBAAL,IAAK,CAAL,KAA2B,IAAK,c;K;IAE3F,kC;MAAA,sC;K;uDACI,oB;MAEQ,I
ADE,QACF,IAAG,CAAH,IADE,QACF,IAAM,EAAN,C;QADJ,OACgB,sBAAS,QAAT,C;WACZ,IAFE,QAEF,IA
AG,EAAH,IAFE,QAEF,IAAO,EAAP,C;QAFJ,OAEiB,sBAAS,WAAW,CAAX,IAAT,C;;QACL,MAAM,gCAAyB
,eAAY,QAAZ,qBAAzB,C;K;;;IAL1B,8C;MAAA,yB;MAAA,6C;QAAA,4B;;MAAA,sC;K;;IA7JJ,+B;MAAA,+yC
;K;;IAAA,oC;MAAA,a;aAAA,Y;UAAA,4C;aAAA,kB;UAAA,kD;aAAA,kB;UAAA,kD;aAAA,kB;UAAA,kD;aA
AA,iB;UAAA,iD;aAAA,c;UAAA,8C;aAAA,kB;UAAA,kD;aAAA,gB;UAAA,gD;aAAA,wB;UAAA,wD;aAAA,sB
;UAAA,sD;aAAA,e;UAAA,+C;aAAA,c;UAAA,8C;aAAA,iB;UAAA,iD;aAAA,gB;UAAA,gD;aAAA,qB;UAAA,q
D;aAAA,S;UAAA,yC;aAAA,Q;UAAA,wC;aAAA,a;UAAA,6C;aAAA,W;UAAA,2C;aAAA,kB;UAAA,kD;aAAA,

mB;UAAA,mD;aAAA,iB;UAAA,iD;aAAA,uB;UAAA,uD;aAAA,mB;UAAA,mD;aAAA,a;UAAA,6C;aAAA,iB;U
AAA,iD;aAAA,iB;UAAA,iD;aAAA,c;UAAA,8C;aAAA,2B;UAAA,2D;aAAA,yB;UAAA,yD;;UAAA,6D;;K;;ICKi
D,2C;uBAA+B,O;;K;;IAC5E,8C;MAAA,kE;MAAuB,qCAAK,IAAL,C;MAAvB,Y;K;ICD8B,gC;MAe9B,gBAAiC
,YAAY,SAAhB,GAA2B,OAA3B,GAAwC,E;K;uFAGjE,Y;MAAQ,OAAO,aAAY,O;K;yCAE/B,iB;MACW,gBAA
P,a;MlGqGG,Q;MAAA,IkGrGc,KlGqGV,IAAS,CAAT,IkGrGU,KlGqGI,IAAS,2BAA3B,C;QAAA,OAAsC,qBkGr
GxB,KlGqGwB,C;;QkGrGf,MAAM,8BAA0B,mCAAyB,WAAzB,MAA1B,C;;MAAhC,W;K;kDAEJ,gC;MAAgF,
OAAA,avG0NY,WuG1NK,UvG0NL,EuG1NiB,QvG0NjB,C;K;6CuGxN5F,iB;MACI,qCAAU,KAAV,C;MACA,O
AAO,I;K;6CAGX,iB;MACI,iBAAgB,SAAN,KAAM,C;MAChB,OAAO,I;K;6CAGX,uC;MACI,OAAA,IAAK,qB
AAY,wBAAS,MAArB,EAA6B,UAA7B,EAAyC,QAAzC,C;K;sCAET,Y;MAayB,UAEK,M;MAL1B,eAAe,E;MA
Cf,YAAY,aAAO,OAAP,GAAgB,CAAhB,I;MACZ,OAAO,SAAS,CAAhB,C;QACI,UAAU,0BAAO,YAAP,EAAO
,oBAAP,Q;QACV,IAAQ,eAAJ,GAAI,CAAJ,IAAwB,SAAS,CAArC,C;UACI,WAAW,0BAAO,cAAP,EAAO,sBA
AP,U;UACX,IAAS,gBAAL,IAAK,CAAT,C;YACI,WAAW,+BAAW,iBAAX,wBAAkB,gBAAlB,C;;YAEX,WAA
W,+BAAW,gBAAX,wBAAiB,iBAAjB,C;;;UAGf,gCAAY,GAAZ,C;;;MAGR,gBAAS,Q;MACT,OAAO,I;K;6CAG
X,iB;MAOI,iBAAgB,SAAN,KAAM,C;MAChB,OAAO,I;K;6CAGX,iB;MAQI,iBAAU,K;MACV,OAAO,I;K;6CA
GX,iB;MAQI,iBAAgB,eAAN,KAAM,C;MAChB,OAAO,I;K;6CAGX,iB;MAOI,gBAAA,IAAK,SAAL,IAAe,wBA
AS,MAAxB,C;MACA,OAAO,I;K;uCAGX,Y;MAU6B,kB;K;qDAE7B,2B;K;8CAcA,kB;MAO0C,OAAA,IAAY,S
AAY,SAAQ,MAAR,C;K;8CAElE,8B;MAQ2D,OAAA,IAAY,SAAY,SAAQ,MAAR,EAAgB,UAAhB,C;K;kDAEn
F,kB;MAQ8C,OAAA,IAAY,SAAY,aAAY,MAAZ,C;K;kDAEtE,8B;MASI,IAAI,MjG0GwC,YAAU,CiG1GlD,IA
AoB,aAAa,CAArC,C;QAAwC,OAAO,E;MAC/C,OAAO,IAAY,SAAY,aAAY,MAAZ,EAAoB,UAApB,C;K;4CA
GnC,wB;MAWI,oCAAa,4BAAmB,KAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,avG+C+E,WuG/C9D,CvG+C8D,
EuG/C3D,KvG+C2D,CuG/C/E,YAA6B,KAA7B,IAAqC,avG4C2B,WuG5CV,KvG4CU,C;MuG3CzE,OAAO,I;K;6
CAGX,wB;MAQI,oCAAa,4BAAmB,KAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,avGiC+E,WuGjC9D,CvGiC8D,
EuGjC3D,KvGiC2D,CuGjC/E,uBAA6B,kBAA7B,IAAqC,avG8B2B,WuG9BV,KvG8BU,C;MuG7BzE,OAAO,I;K;
6CAGX,wB;MAUI,oCAAa,4BAAmB,KAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,avGiB+E,WuGjB9D,CvGiB8D
,EuGjB3D,KvGiB2D,CuGjB/E,GAAmC,eAAN,KAAM,CAAnC,GAAsD,avGcU,WuGdO,KvGcP,C;MuGbzE,OA
AO,I;K;6CAGX,wB;MAaI,oCAAa,4BAAmB,KAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,avGF+E,WuGE9D,CvG
F8D,EuGE3D,KvGF2D,CuGE/E,GAAmC,SAAN,KAAM,CAAnC,GAAgD,avGLgB,WuGKC,KvGLD,C;MuGMz
E,OAAO,I;K;6CAGX,wB;MAWI,oCAAa,4BAAmB,KAAnB,EAA0B,WAA1B,C;MAEb,gBAAS,avGnB+E,WuG
mB9D,CvGnB8D,EuGmB3D,KvGnB2D,CuGmB/E,GAAmC,SAAN,KAAM,CAAnC,GAAgD,avGtBgB,WuGsBC,
KvGtBD,C;MuGuBzE,OAAO,I;K;6CAGX,wB;MAUI,oCAAa,4BAAmB,KAAnB,EAA0B,WAA1B,C;MAEb,eAA
e,wBAAS,M;MACxB,gBAAc,IAAK,SvGpCqE,WuGoCpD,CvGpCoD,EuGoCjD,KvGpCiD,CuGoC1E,GAAkC,Q
AAlC,GAA6C,IAAK,SvGvCS,WuGuCQ,KvGvCR,C;MuGwCzE,OAAO,I;K;gDAGX,qB;MAcI,IAAI,YAAY,CA
AhB,C;QACI,MAAM,gCAAyB,0BAAuB,SAAvB,MAAzB,C;;MAGV,IAAI,aAAa,WAAjB,C;QACI,gBAAS,avG3
D2E,WuG2D1D,CvG3D0D,EuG2DvD,SvG3DuD,C;;QuG6DpF,aAAU,WAAV,MAAuB,SAAvB,M;UACI,qCAA
U,CAAV,C;;;K;gDAKZ,sB;MAQI,oCAAa,4BAAmB,UAAnB,EAA+B,WAA/B,C;MAEb,OAAO,avGhFkE,WuGg
FjD,UvGhFiD,C;K;gDuGmF7E,gC;MAQI,oCAAa,4BAAmB,UAAnB,EAA+B,QAA/B,EAAyC,WAAzC,C;MAEb,
OAAO,avG1FiF,WuG0FhE,UvG1FgE,EuG0FpD,QvG1FoD,C;K;yCuG6F5F,Y;K;uCAcA,Y;MAAkC,oB;K;oCAEl
C,Y;MAOI,gBAAS,E;MACT,OAAO,I;K;0CAGX,wB;MAQI,oCAAa,2BAAkB,KAAlB,EAAyB,WAAzB,C;MAEb
,gBAAS,avGlI+E,WuGkI9D,CvGlI8D,EuGkI3D,KvGlI2D,CuGkI/E,uBAA6B,kBAA7B,IAAqC,avGrI2B,WuGqIV
,QAAQ,CAAR,IvGrIU,C;K;+CuGwI7E,uC;MAYI,yBAAkB,UAAlB,EAA8B,QAA9B,EAAwC,WAAxC,C;MAEA
,gBAAc,IAAK,SvGnJqE,WuGmJpD,CvGnJoD,EuGmJjD,UvGnJiD,CuGmJ1E,GAAuC,KAAvC,GAA+C,IAAK,Sv
GtJO,WuGsJU,QvGtJV,C;MuGuJzE,OAAO,I;K;kDAGX,wC;MACI,IAAI,aAAa,CAAb,IAAkB,aAAa,MAAnC,C;
QACI,MAAM,8BAA0B,iBAAc,UAAd,kBAAmC,MAA7D,C;;MAEV,IAAI,aAAa,QAAjB,C;QACI,MAAM,gCAA
yB,gBAAa,UAAb,qBAAqC,QAArC,MAAzB,C;;K;+CAId,iB;MAYI,oCAAa,2BAAkB,KAAlB,EAAyB,WAAzB,C
;MAEb,gBAAS,avG9K+E,WuG8K9D,CvG9K8D,EuG8K3D,KvG9K2D,CuG8K/E,GAA6B,avGjLmC,WuGiLlB,Q
AAQ,CAAR,IvGjLkB,C;MuGkLzE,OAAO,I;K;kDAGX,gC;MAWI,yBAAkB,UAAlB,EAA8B,QAA9B,EAAwC,EAAAwC,
WAAxC,C;MAEA,gBAAS,avG/L+E,WuG+L9D,CvG/L8D,EuG+L3D,UvG/L2D,CuG+L/E,GAAkC,avGlM8B,Wu
GkMb,QvGlMa,C;MuGmMzE,OAAO,I;K;kDAGX,gE;MAc+C,iC;QAAAA,oBAAAyB,C;MAAAG,0B;QAAAA,aAAAkB,

C;MAAG,wB;QAAA,WAAgB,IAAK,O;MAKlF,IACf,I;MALhB,oCAAa,4BAAmB,UAAnB,EAA+B,QAA/B,EAA
yC,WAAzC,C;MACb,oCAAa,4BAAmB,iBAAnB,EAAsC,oBAAoB,QAApB,GAA+B,UAA/B,IAAtC,EAAiF,WA
AY,OAA7F,C;MAEb,eAAe,iB;MACf,iBAAc,UAAd,UAA+B,QAA/B,U;QACI,YAAY,eAAZ,EAAY,uBAAZ,UA
A0B,yBAAO,KAAP,C;;K;kDAIlC,uC;MAcI,iBAAgB,iBAAN,KAAM,EAAe,UAAf,EAA2B,QAA3B,C;MAChB,
OAAO,I;K;kDAGX,uC;MAYI,gBAAgB,KAAM,W;MACtB,oCAAa,4BAAmB,UAAnB,EAA+B,QAA/B,EAAyC,
SAAU,OAAnD,C;MAEb,iBAAU,SvG5P8E,WuG4P1D,UvG5P0D,EuG4P9C,QvG5P8C,C;MuG6PxF,OAAO,I;K;k
DAGX,8C;MAgBI,oCAAa,4BAAmB,KAAnB,EAA0B,IAAK,OAA/B,C;MAEb,gBAAS,avGlR+E,WuGkR9D,CvG
lR8D,EuGkR3D,KvGlR2D,CuGkR/E,GAAmC,iBAAN,KAAM,EAAe,UAAf,EAA2B,QAA3B,CAAnC,GAA0E,av
GrRV,WuGqR2B,KvGrR3B,C;MuGsRzE,OAAO,I;K;kDAGX,8C;MAgBI,oCAAa,4BAAmB,KAAnB,EAA0B,WA
A1B,C;MAEb,gBAAgB,KAAM,W;MACtB,oCAAa,4BAAmB,UAAnB,EAA+B,QAA/B,EAAyC,SAAU,OAAnD,
C;MAEb,gBAAS,avG3S+E,WuG2S9D,CvG3S8D,EuG2S3D,KvG3S2D,CuG2S/E,GAA6B,SvG3SkD,WuG2S9B,U
vG3S8B,EuG2SlB,QvG3SkB,CuG2S/E,GAAyE,avG9ST,WuG8S0B,KvG9S1B,C;MuG+SzE,OAAO,I;K;;IA5hBX,
6C;MAAA,uD;MAKoC,2B;MALpC,Y;K;IAQA,8C;MAAA,uD;MAC4C,0BAAK,OAAQ,WAAb,C;MAD5C,Y;K;I
AGA,qC;MAAA,uD;MACuB,0BAAK,EAAL,C;MADvB,Y;K;2EAshBJ,qB;MAOgE,OAAA,SAAK,Q;K;uEAErE,
mC;MAQ+E,SAAK,aAAI,KAAJ,EAAW,KAAX,C;K;+EAEpF,kD;MAaI,OAAA,SAAK,kBAAS,UAAT,EAAqB,Q
AArB,EAA+B,KAA/B,C;K;+EAET,4B;MAY6E,OAAA,SAAK,kBAAS,KAAT,C;K;qFAElF,2C;MAWoG,OAAA,
SAAK,qBAAY,UAAZ,EAAwB,QAAxB,C;K;uFAEzG,2E;MAe2E,iC;QAAA,oBAAyB,C;MAAG,0B;QAAA,aAA
kB,C;MAAG,wB;QAAA,WAAgB,SAAK,O;MAC7I,SAAK,qBAAY,WAAZ,EAAyB,iBAAzB,EAA4C,UAA5C,E
AAwD,QAAxD,C;K;qFAET,kD;MAeI,OAAA,SAAK,qBAAY,KAAZ,EAAmB,UAAnB,EAA+B,QAA/B,C;K;uFA
ET,kD;MAaI,OAAA,SAAK,qBAAY,KAAZ,EAAmB,UAAnB,EAA+B,QAA/B,C;K;qFAET,yD;MAiBI,OAAA,S
AAK,qBAAY,KAAZ,EAAmB,KAAnB,EAA0B,UAA1B,EAAsC,QAAtC,C;K;uFAET,yD;MAiBI,OAAA,SAAK,q
BAAY,KAAZ,EAAmB,KAAnB,EAA0B,UAA1B,EAAsC,QAAtC,C;K;qFxG1rBT,qB;MAMoD,OA6BW,8BAAY,
cAfrB,YAAY,CAAZ,C;K;yFAZtD,qB;MAYsD,OAeS,8BAAY,cAfrB,YAAY,CAAZ,C;K;iFAEtD,qB;MAaoD,OA
AW,8BAAY,c;K;qFAE3E,yB;MAAA,uD;MAAA,4B;QAMoD,+B;O;KANpD,C;IAQA,kC;MAYI,gBAiB2D,8BA
AY,c;MAhBvE,OAAW,SAAU,OAAV,GAAmB,CAAvB,GAA0B,SAA1B,GAAoC,qBAAU,CAAV,C;K;iFAG/C,q
B;MAaoD,OAAW,8BAAY,c;K;IAE3E,kC;MAU+C,mC;K;IAE/C,oC;MAGoD,QAAQ,cAAA,sCAAK,mBAAL,E
AAyB,sCAAK,mBAA9B,CAAR,6B;K;IAEpD,mC;MAGmD,QAAQ,cAAA,sCAAK,kBAAL,EAAwB,sCAAK,kB
AA7B,CAAR,6B;K;IAO/C,iC;MAAQ,OAAA,oCAAa,iBAAQ,2BAAR,C;K;IAEzB,8B;MAOI,IAAI,YAAO,GAA
X,C;QACI,OAAO,I;;MAEX,OAAO,gCAA8C,mD;K;IAGzD,6B;MAUI,IAAI,CAAQ,kBAAK,GAAL,CAAR,iCA
AoB,CAAQ,kBAAK,EAAL,CAAR,6BAAxB,C;QACI,OAAO,I;;MAEX,IAAI,YAAO,GAAX,C;QACI,OAAO,K;;
MAEX,OAAO,uB;K;IAGX,oC;MAUI,IAAI,CAAQ,kBAAK,GAAL,CAAR,iCAAoB,CAAQ,kBAAK,EAAL,CAA
R,6BAApB,IAAwC,CAAQ,kBAAK,EAAL,CAAR,6BAA5C,C;QACI,OAAO,I;;MAEX,IAAI,YAAO,GAAX,C;Q
ACI,OAAO,K;;MAGX,OAAO,0BAAiB,uB;K;IAG5B,4B;MASI,IAAI,CAAQ,kBAAK,EAAL,CAAR,6BAAJ,C;Q
ACI,OAAO,I;;MAEX,IAAI,YAAO,GAAX,C;QACI,OAAO,K;;MAEX,OAAO,sB;K;IAGX,gC;MAUI,IAAI,CAA
Q,kBAAK,EAAL,CAAR,6BAAJ,C;QACI,OAAO,I;;MAEX,IAAI,YAAO,GAAX,C;QACI,OAAO,K;;MAEX,OAA
O,0B;K;IAGX,gC;MAUI,IAAI,CAAQ,kBAAK,GAAL,CAAR,6BAAJ,C;QACI,OAAO,I;;MAEX,IAAI,YAAO,GA
AX,C;QACI,OAAO,K;;MAEX,OAAO,0B;K;IAGX,gC;MASI,IAAI,YAAO,GAAX,C;QACI,OAAO,K;;MAEX,OA
AO,gCAAoD,yD;K;IAG/D,iC;MAUI,OAAO,aAAQ,EAAR,IAAoB,CAAQ,mBAAU,GAAV,CAAR,6B;K;IAG/B,i
C;MAMiD,kC;K;iFyGtPjD,yB;MAAA,+C;MAAA,4B;QAMuD,OAAK,UAAL,SAAK,C;O;KAN5D,C;IAQA,gC;
MAMiD,4B;MAAA,S;QAAgB,cAAA,SxG4LC,cwG5LD,EAAoB,MAApB,C;;MAAhB,W;K;IAEjD,6B;MAI0C,Q;
MAAA,yDAAkB,kBAAkB,SAAlB,C;K;IAE5D,oC;MAKoD,Q;MAAA,yCAAa,KAAb,oBAAuB,kBAAkB,SAAlB,
C;K;IAG3E,8B;MAI4C,Q;MAAA,0DAAmB,kBAAkB,SAAlB,C;K;IAE/D,qC;MAKsD,Q;MAAA,0CAAc,KAAd,
oBAAwB,kBAAkB,SAAlB,C;K;IAE9E,0B;MAIwC,Q;MAAA,wDAAiB,kBAAkB,SAAlB,C;K;IAEzD,mC;MAKk
D,Q;MAAA,wCAAY,KAAZ,oBAAsB,kBAAkB,SAAlB,C;K;IAExE,2B;MAI0C,Q;MAAA,yDAAkB,kBAAkB,SA
AlB,C;K;IAE5D,oC;MAKoD,Q;MAAA,yCAAa,KAAb,oBAAuB,kBAAkB,SAAlB,C;K;IAE3E,6B;MAIyF,kBAA
1C,CAAO,S;MAClD,IAAO,QlHeD,WkHfC,CAAH,IAAc,CAAM,kBAApB,KlHeE,WkHf6B,KAAM,GAAN,IAAk
B,kBAAjD,CAAJ,C;QACI,4B;MAFsC,OlHiBnC,W;K;6EkHZX,yB;MAAA,6C;MAAA,4B;QAKmD,0B;O;KALn
D,C;IAOA,mC;MAIgG,kBAA1C,CAAO,S;MAAR,OACjD,EAAK,QlH2BgB,WkH3BhB,CAAH,IAAc,CAAM,kB

AApB,KlH2BmB,WkH3BY,KAAM,GAAN,IAAkB,kBAAjD,CAAF,ClH2BO,GAAqB,WAArB,GAA+B,I;K;yFkH
xB1C,yB;MAAA,yD;MAAA,4B;QAK0D,gC;O;KAL1D,C;iFAOA,yB;MAAA,6C;MAAA,mC;QAO6D,OAAa,SA
AR,SAAQ,EAAS,KAAT,C;O;KAP1E,C;iFASA,yB;MAAA,6C;MAAA,mC;QAO8D,OAAa,SAAR,SAAQ,EAAS,
KAAT,C;O;KAP3E,C;IASA,sC;MAMqD,OAAA,SAAY,UAAS,WAAW,KAAX,CAAT,C;K;IAEjE,4B;MAAsC,Q
AAM,SxG4EsB,cwG5E5B,C;aAClC,K;aAAA,M;aAAA,M;UADkC,OACT,I;;UADS,OAE1B,K;;K;IAGZ,2B;MA
KI,IAAI,EAAU,CAAV,sBAAa,EAAb,CAAJ,C;QACI,MAAM,gCAAyB,WAAQ,KAAR,kCAAzB,C;;MAEV,OAA
O,K;K;IAGX,8B;MAA2D,Q;MACvD,YAAQ,EAAR,IAAe,QAAQ,EAAvB,C;QAA8B,cAAO,E;WACrC,YAAQ,E
AAR,IAAe,QAAQ,EAAvB,C;QAA8B,cAAO,EAAP,GAAa,EAAb,I;WAC9B,YAAQ,EAAR,IAAe,QAAQ,GAAvB
,C;QAA8B,cAAO,EAAP,GAAa,EAAb,I;WAC9B,WAAO,GAAP,C;QAAmB,S;WACnB,YAAQ,KAAR,IAAoB,Q
AAQ,KAA5B,C;QAAwC,cAAO,KAAP,GAAkB,EAAlB,I;WACxC,YAAQ,KAAR,IAAoB,QAAQ,KAA5B,C;QA
AwC,cAAO,KAAP,GAAkB,EAAlB,I;;QAC3B,sBAAL,IAAK,C;MlH9CN,a;MkHuCgD,OAQ/C,WAAJ,GAAiB,E
AAjB,GAAyB,E;K;IClJG,2C;MAHpC,e;MAGqC,kB;MAHrC,iB;MAAA,uB;K;IAAA,kC;MAAA,qC;O;MAII,qEA
CY,GADZ,C;MAEA,iEAIU,GAJV,C;K;;IAFA,+C;MAAA,wB;MAAA,uC;K;;IAEA,6C;MAAA,wB;MAAA,qC;K;
;IANJ,8B;MAAA,mF;K;;IAAA,mC;MAAA,a;aAAA,a;UAAA,4C;aAAA,W;UAAA,0C;;UAAA,4D;;K;;IAawG,4B;
MAAE,OAAA,EAAG,M;K;IAA7G,qC;MAAqE,iCAAa,EAAb,EAA0B,OAA1B,0BAAmC,cAAnC,C;K;IAQlC,2B
;MAAC,kB;K;;sCALpC,Y;MAKoC,iB;K;wCALpC,iB;MAAA,sBAKoC,qCALpC,C;K;oCAAA,Y;MAAA,OAKoC
,iDALpC,M;K;oCAAA,Y;MAAA,c;MAKoC,sD;MALpC,a;K;kCAAA,iB;MAAA,2IAKoC,sCALpC,G;K;IAQA,g
C;MAUsB,gB;MAAA,iF;MAAA,mB;QACX,MAAM,qCAA8B,8DAA9B,C;;MADb,kBAAkB,M;MAGlB,OAAO,
wBAAY,IAAZ,C;K;IAiBe,iC;MA4PtB,6B;MAnPA,eACoC,O;MACpC,eACsD,QAAR,OAAQ,C;MACtD,uBAAo
C,WAAO,OAAP,EAAwB,QAAR,OAAQ,EAAQ,IAAR,CAAxB,C;MACpC,6BAA2C,I;MAI3C,oCAAkD,I;K;0CA
HlD,Y;MACI,Q;MAAA,U;MAAA,gD;QAAA,a;;QAA8D,gBAAvC,WAAO,YAAP,EAAwB,QAAR,YAAQ,EAAQ
,IAAR,CAAxB,C;QAA8C,6BnHkBnE,S;QmHlBF,SnHmBG,S;;MmHnBH,a;K;iDAGJ,Y;MACI,Q;MAAA,U;MA
AA,uD;QAAA,a;;QnH3BG,gB;QmH4BC,IAAY,aAAR,YAAQ,EAAW,EAAX,CAAR,IAAmC,WAAR,YAAQ,EA
AS,EAAT,CAAvC,C;UAAA,eACI,oB;;UAEA,OAAO,WAAO,MAA2B,UAAf,YAAR,YAAQ,qBAAU,EAAV,EA
Ae,qBAAQ,EAAR,EAA3B,MAAP,EAA2D,QAAR,YAAQ,EAAQ,IAAR,CAA3D,C;QACb,4B;QAAO,oCnHSP,S;
QmHdF,SnHeG,S;;MmHfH,a;K;sCAQJ,iB;MAEkB,MAAd,oBAAc,C;MACd,YAAY,oBAAc,MAAK,KAAM,WA
AX,C;MAC1B,OAAO,iBAAiB,KAAM,MAAN,KAAe,CAAhC,IAAqC,oBAAc,UAAd,KAA2B,KAAM,O;K;8CA
GjF,iB;MAEkB,MAAd,oBAAc,C;MACd,OAAO,oBAAc,MAAK,KAAM,WAAX,C;K;wCAGzB,wB;MAGI,IAAI,
QAAQ,CAAR,IAAa,QAAQ,KAAM,OAA/B,C;QACI,MAAM,8BAA0B,0BAAuB,KAAvB,wBAA8C,KAAM,OA
A9E,C;;MAEV,cAAc,0B;MACd,oBAAoB,K;MACpB,OAAO,OAAQ,MAAK,KAAM,WAAX,C;K;mCAGnB,6B;
MAS4C,0B;QAAA,aAAkB,C;MAC1D,IAAI,aAAa,CAAb,IAAkB,aAAa,KAAM,OAAzC,C;QACI,MAAM,8BAA
0B,gCAA6B,UAA7B,wBAAyD,KAAM,OAAzF,C;;MAEV,OAAqB,SAAd,oBAAc,EAAS,KAAM,WAAf,EAA2B,
UAA3B,EAAuC,oBAAvC,C;K;IAeG,6E;MAAA,mB;QAAE,+BAAK,aAAL,EAAY,kBAAZ,C;O;K;IAA2B,uC;M
AAW,OAAA,KAAM,O;K;sCAZ1E,6B;MAQ+C,0B;QAAA,aAAkB,C;MAC7D,IAAI,aAAa,CAAb,IAAkB,aAAa,
KAAM,OAAzC,C;QACI,MAAM,8BAA0B,gCAA6B,UAA7B,wBAAyD,KAAM,OAAzF,C;;MAEV,OAAO,mBA
AiB,6CAAjB,EAA8C,sBAA9C,C;K;0CAGX,iB;MAMI,OAA2B,SAA3B,iCAA2B,EAAS,KAAM,WAAf,EAA2B,
CAA3B,EAA8B,oBAA9B,C;K;sCAE/B,wB;MAGI,IAAI,QAAQ,CAAR,IAAa,QAAQ,KAAM,OAA/B,C;QACI,M
AAM,8BAA0B,0BAAuB,KAAvB,wBAA8C,KAAM,OAA9E,C;;MAEV,OAA2B,SAApB,0BAAoB,EAAS,KAAM
,WAAf,EAA2B,KAA3B,EAAkC,oBAAlC,C;K;IA2BL,mD;MAAA,qB;QAAE,2BAAoB,EAApB,EAAwB,mBAAx
B,C;O;K;sCAvB5B,8B;MAoBI,IAAI,CAAa,YAAZ,WAAY,EAAS,EAAT,CAAb,IAA+B,CAAa,YAAZ,WAAY,E
AAS,EAAT,CAAhD,C;QACI,OAAO,KAAM,WzGoF4E,SyGpFnD,oBzGoFmD,EyGpFpC,WzGoFoC,C;;MyGlF7
F,OAAO,qBAAQ,KAAR,EAAe,iCAAf,C;K;sCAGX,4B;MAMI,YAAY,kBAAK,KAAL,C;MACZ,IAAI,aAAJ,C;Q
AAmB,OAAO,KAAM,W;MAEhC,gBAAgB,C;MAChB,aAAa,KAAM,O;MACnB,SAAS,mBAAc,MAAd,C;;QAE
L,iBAAiB,oB;QACjB,EAAG,gBAAO,KAAP,EAAc,SAAd,EAAyB,UAAW,MAAM,MAA1C,C;QACH,EAAG,gB
AAO,UAAU,UAAV,CAAP,C;QACH,YAAY,UAAW,MAAM,aAAjB,GAAgC,CAAhC,I;QACZ,QAAQ,UAAW,O
;;MACd,oBAAY,MAAZ,IAAsB,aAAtB,C;MAET,IAAI,YAAY,MAAhB,C;QACI,EAAG,gBAAO,KAAP,EAAc,S
AAd,EAAyB,MAAzB,C;;MAGP,OAAO,EAAG,W;K;2CAGd,8B;MAyB+B,IAAf,I;MALZ,IAAI,CAAa,YAAZ,W
AAY,EAAS,EAAT,CAAb,IAA+B,CAAa,YAAZ,WAAY,EAAS,EAAT,CAAhD,C;QACI,uBAA+B,QAAR,YAAQ

,EAAQ,GAAR,C;QAC/B,OAAO,KAAM,WzG8B4E,SyG9BnD,WAAO,YAAP,EAAgB,gBAAhB,CzG8BmD,EyG
9BhB,WzG8BgB,C;;MyG3BjF,yBAAK,KAAL,C;MAAA,iB;QAAe,OAAO,KAAM,W;;MAAxC,YAAY,I;MCqKO
,gBAAhB,sB;MDlKC,yBnG4KgF,0BmG5KzD,CnG4KyD,EmG5KhD,WAAM,MnG4K0C,CAAkC,WmG5KlH,C;
MACA,yBAAO,uCAAP,C;MACA,yBnG0KgF,0BmG1KnD,WAAM,KAAZ,GAAmB,CAAnB,InG0KyD,EmG1K7
B,YnG0K6B,CAAkC,WmG1KlH,C;MAHJ,OnHjKG,SoHoUqC,W;K;oCD5J5C,wB;MAO6C,qB;QAAA,QAAa,C;
MAMxC,Q;MALd,wBAAwB,KAAxB,C;MnHpJG,SmHqJW,qBAAQ,KAAR,C;MAAd,cAAuC,UAAS,CAAb,GA
AgB,EAAhB,GAA2B,OAAH,EAAG,EAAK,QAAQ,CAAR,IAAL,C;MAC9D,ahI1KgD,gB;MgI2KhD,gBAAgB,C;
MAEF,yB;MAAd,OAAc,cAAd,C;QAAc,uB;QACV,MAAO,WAAU,mBAAN,KAAM,EAAY,SAAZ,EAAuB,KAA
M,MAAM,MAAnC,CAA0C,WApD,C;QACP,YAAY,KAAM,MAAM,aAAZ,GAA2B,CAA3B,I;;MAEhB,MAA
O,WAAU,mBAAN,KAAM,EAAY,SAAZ,EAAuB,KAAM,OAA7B,CAAqC,WAA/C,C;MACP,OAAO,M;K;IAgB
S,yI;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,0C;MAAA,oC;MAAA,0C;MAAA,yB;MAAA,6B;MAAA,8B;MA
AA,8B;MAAA,kC;K;;;;gEAAA,Y;;;;;iCACA,mCAAK,wBAAL,C;cACZ,IAAI,4BAAiB,6BAAS,CAA9B,C;gBAC
I,gB;gCAAA,iCAAM,wBAAM,WAAZ,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBADJ,gB;;;;;;;cAEI,M;;qCAGY,C
;sCACC,C;cAEjB,gB;;;sCACqB,+B;cACjB,gB;8BAAA,iCnGwH4E,mBmGxHtE,wBnGwHsE,EmGxHtD,oBnGw
HsD,EmGxH3C,qBAAW,MAAM,MnGwH0B,CAAkC,WmGxH9G,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cACA,u
BAAY,qBAAW,MAAM,aAAjB,GAAgC,CAAhC,I;cACZ,mBAAQ,qBAAW,O;cAJvB,KAKS,qDALT,EAKS,qBA
LT,OAKyB,2BAAQ,CAAR,IALzB,KAKsC,gBALtC,S;gBAAA,gB;;;;cAAA,gB;;;cAOA,gB;8BAAA,iCnGmHgF,
mBmGnH1E,wBnGmH0E,EmGnH1D,oBnGmH0D,EmGnH/C,wBAAM,OnGmHyC,CAAkC,WmGnHlH,O;kBAA
A,2C;uBAAA,yB;cAAA,Q;;cAhBA,OAgBA,a;;;;;;;;;;;;;;K;IAjBY,sF;MAAA,yD;uBAAA,6H;YAAA,S;iBAAA,Q;;
iBAAA,uB;O;K;8CAbpB,wB;MAUuD,qB;QAAA,QAAa,C;MAChE,wBAAwB,KAAxB,C;MAEA,OAAO,SAAS,g
DAAT,C;K;+BAsBX,Y;MAMyC,OAAA,oBAAc,W;K;IAEvD,2B;MAAA,+B;MAmBI,uBAA4B,WAAO,uBAAP,
EAAiC,GAAjC,C;MAC5B,2BAAgC,WAAO,SAAP,EAAoB,GAApB,C;MAGhC,iCAAsC,WAAO,KAAP,EAAiB,
GAAjB,C;K;oDAtBtC,mB;MAIwD,oBAAM,oBAAO,OAAP,CAAN,C;K;+CAExD,mB;MAIoD,OAAA,OzGzDyC
,SyGyDnB,oBzGzDmB,EyGyDJ,MzGzDI,C;K;0DyG2D7F,mB;MAI+D,OAAA,OzG/D8B,SyG+DR,wBzG/DQ,Ey
G+DW,MzG/DX,C;K;gEyGoE7F,mB;MAAgE,OAAA,OzGpE6B,SyGoEP,8BzGpEO,EyGoEkB,MzGpElB,C;K;;;I
yG8CjG,uC;MAAA,sC;QAAA,qB;;MAAA,+B;K;;IA1PA,4C;MAAA,+C;MACkE,kBAAK,OAAL,EAAc,MAAM,
MAAN,CAAd,C;MADlE,Y;K;IAGA,sC;MAAA,+C;MAC6C,kBAAK,OAAL,EAAc,UAAd,C;MAD7C,Y;K;IA0R
O,kG;MAAA,kC;MAAA,8C;MAAA,kC;MAAA,kC;MACH,uBAA+B,a;MAI/B,4F;MA0BA,sBAA0C,I;K;+FA9B1
C,Y;MAAA,2B;K;+FAEI,Y;MAAQ,qBAAA,kBN9S8C,CM8SxC,CN9SwC,CM8S9C,C;K;gGAEZ,Y;MAAA,4B;
K;iEAqBA,mB;MACI,OAAO,MAAa,UAAU,eAAe,MAAK,CAAL,EAAQ,IAAR,C;K;IAStB,oG;MAAA,kC;MAA
S,uB;K;mJACG,Y;MAAQ,OAAA,kBAAM,O;K;wGACrC,iB;MAAuC,Q;MAAA,eAAA,kBNjVG,CMiVG,KNjVH
,CMiVH,mBAAgB,E;K;;qGAJnE,Y;MACI,IAAI,2BAAJ,C;QACI,yH;;MAKJ,OAAO,kC;K;4CAGf,Y;MACI,OAA
Y,SAAZ,wBAAY,EAAS,kBAAT,EAAoB,kBAAM,UAAV,GAAqB,8BAAuB,kBAAM,MAA7B,CAArB,GAA8D,
kBAAM,aAAN,GAAqB,CAArB,IAA9E,EAAsG,wBAAtG,C;K;gEAEhB,iB;MACI,IAAI,QAAc,iBAAN,kBAAM,
CAAlB,C;QACI,YAAkB,kBAAY,YAAW,KAAX,C;QAC9B,IAAa,KAAT,sBAAiB,KAArB,C;UACI,YAAkB,kB
AAY,YAAW,QAAQ,CAAR,IAAX,C;UAC9B,IAAa,KAAT,sBAAiB,KAArB,C;YACI,OAAO,QAAQ,CAAR,I;;;;
MAInB,OAAO,QAAQ,CAAR,I;K;IApDiC,2E;MAAA,kC;MAAA,kB;MAAoC,6B;K;mHACrD,Y;MAAQ,OAAA,
kBAAM,O;K;IACqC,4E;MAAA,qB;QAAE,yBAAK,EAAL,C;O;K;qEAA5E,Y;MAAiD,OAAqB,OAAb,aAAR,oB
AAQ,CAAa,EAAI,iEAAJ,CAAiB,W;K;wEACvF,iB;MAA4C,Q;MAAA,eAAA,kBNnTU,CMmTJ,KNnTI,CMmT
V,YAAoB,oBAApB,O;K;wEAE5C,gB;MAIW,IADwB,IACxB,EAQ6C,MAR7C,EAQA,M;MATwB,OAAZ,kBAA
Y,O;MAAlB,iB;QACN,MAAM,gCAAyB,gCAA6B,IAA7B,oEAAzB,C;;MADb,aAAa,I;MAKb,IAAI,CAAC,qCA
AwB,MAAxB,EAAgC,IAAhC,CAAL,C;QACI,MAAM,gCAAyB,gCAA6B,IAA7B,qBAAzB,C;MAEV,YAAY,O
AAO,IAAP,C;MACL,IAAI,SAAS,SAAb,C;QAAwB,a;;QAAU,wBAAW,4DAAX,C;;MAAzC,a;K;;;IA5BhB,uD;M
ACI,sBAAiB,I;MACjB,YAAY,eAAK,KAAL,C;MACZ,IAAI,aAAJ,C;QAAmB,OAAO,I;MAC1B,YAAY,aAAA,K
AAM,MAAN,EAAa,sBAAY,CAAZ,IAAb,C;MAEZ,mE;K;IA8DJ,iD;MAM+B,UAKO,MALP,EAoBD,MApBC,E
AoBD,MApBC,EAiCD,MAjCC,EAiCD,M;MArC1B,YAAY,C;MACZ,aAAa,sB;MAEb,OAAO,QAAQ,WAAY,O
AA3B,C;QACI,WAAW,wBAAY,YAAZ,EAAY,oBAAZ,Q;QACX,IAAI,SAAQ,EAAZ,C;UACI,IAAI,UAAS,WA
AY,OAAzB,C;YACI,MAAM,gCAAyB,mCAAzB,C;UAEV,MAAO,gBAAO,wBAAY,cAAZ,EAAY,sBAAZ,UAA

P,C;eACJ,IAAI,SAAQ,EAAZ,C;UACH,IAAI,UAAS,WAAY,OAAzB,C;YACI,MAAM,gCAAyB,kCAAzB,C;UA
EV,IAAI,uBAAY,KAAZ,MAAsB,GAA1B,C;YACI,eAA2B,cAAZ,WAAY,GAAc,qBAAd,EAAc,KAAd,E;YAE3
B,IAAI,UAAS,QAAb,C;cACI,MAAM,gCAAyB,8DAAzB,C;YACV,IAAI,aAAY,WAAY,OAAxB,IAAkC,uBAAY
,QAAZ,MAAyB,GAA/D,C;cACI,MAAM,gCAAyB,yDAAzB,C;YAEV,gBAAgB,WzGvLgE,WyGuL1C,KzGvL0C
,EyGuLnC,QzGvLmC,C;YyGyLhF,MAAO,gBAAO,0BAAA,KAAM,OAAN,EAAa,SAAb,qDAAkC,EAAzC,C;Y
ACP,QAAQ,WAAW,CAAX,I;;YAER,IAAI,EAAuB,kBAAK,EAAL,CAAvB,0CAAY,KAAZ,EAAJ,C;cACI,MAA
M,gCAAyB,mCAAzB,C;YAEV,aAAa,KAAM,O;YACnB,iBAA2B,eAAZ,WAAY,EAAe,KAAf,EAAsB,MAAO,K
AA7B,C;YAC3B,iBAAwD,MAAvC,WzGjM+D,WyGiMzC,KzGjMyC,EyGiMlC,UzGjMkC,CyGiMxB,C;YAExD,
IAAI,cAAc,MAAO,KAAzB,C;cACI,MAAM,8BAA0B,sBAAmB,UAAnB,oBAA1B,C;YAEV,MAAO,gBAAO,uC
AAO,UAAP,qDAA6B,EAApC,C;YACP,QAAQ,U;;;UAGZ,MAAO,gBAAO,IAAP,C;;;MAGf,OAAO,MAAO,W;
K;IAGlB,8C;MAKI,YAAY,U;MACZ,OAAO,QAAQ,gBAAf,C;QACI,IAAI,qBAAK,KAAL,MAAe,GAAnB,C;UA
CI,K;;UAEA,qB;;;MAGR,OAAO,K;K;IAGX,2D;MAEI,YAAY,aAAa,CAAb,I;MACZ,iBAAiB,qBAAK,UAAL,IA
AmB,E;MAGpC,OAAO,QAAQ,gBAAR,IAAkB,CAAe,kBAAK,EAAL,CAAf,wCAAK,KAAL,EAAzB,C;QACI,o
BAAoB,CAAC,aAAa,EAAb,IAAD,KAAqB,qBAAK,KAAL,IAAc,EAAnC,K;QACpB,IAAqB,CAAjB,qCAAyB,U
AA7B,C;UACI,aAAa,a;UACb,qB;;UAEA,K;;;MAGR,OAAO,K;K;IzGneX,yB;MAQiB,Q;MADb,aAAa,E;MACb,
wBAAa,KAAb,gB;QAAa,WAAb,UAAa,KAAb,O;QACI,8BAAU,IAAV,C;;MAEJ,OAAO,M;K;IAGX,yC;MAa+B
,Q;MAH3B,IAAI,SAAS,CAAT,IAAc,SAAS,CAAvB,IAA4B,CAAA,KAAM,OAAN,GAAa,MAAb,QAAsB,MAAt
D,C;QACI,MAAM,8BAA0B,WAAS,KAAM,OAAf,kBAA+B,MAA/B,kBAAgD,MAA1E,C;MACV,aAAa,E;MAC
c,gBAAS,MAAT,I;MAA3B,iBAAc,MAAd,wB;QACI,8BAAU,MAAM,KAAN,CAAV,C;;MAEJ,OAAO,M;K;IAG
X,mC;MAOiB,Q;MADb,aAAa,E;MACb,wBAAa,SAAb,gB;QAAa,WAAb,UAAa,SAAb,O;QACI,8BAAU,IAAV,
C;;MAEJ,OAAO,M;K;IAGX,2D;MAY2C,0B;QAAA,aAAkB,C;MAAG,wB;QAAA,WAAgB,SAAK,O;MACjF,oC
AAa,4BAAmB,UAAnB,EAA+B,QAA/B,EAAyC,SAAK,OAA9C,C;MACb,aAAa,E;MACb,iBAAc,UAAd,UAA+
B,QAA/B,U;QACI,8BAAU,UAAK,KAAL,CAAV,C;;MAEJ,OAAO,M;K;IASkB,gD;MAAA,qB;QAAE,+CAAI,E
AAJ,E;O;K;IAN/B,kC;MAMI,OAAO,kBAAU,gBAAV,EAAkB,+BAAlB,C;K;IAiBiC,oE;MAAA,qB;QAAE,+CA
AI,qBAAa,EAAb,IAAJ,E;O;K;IAd9C,wD;MAYqC,0B;QAAA,aAAkB,C;MAAG,wB;QAAA,WAAgB,SAAK,O;M
AC3E,oCAAa,4BAAmB,UAAnB,EAA+B,QAA/B,EAAyC,gBAAzC,C;MACb,OAAO,kBAAU,WAAW,UAAX,IA
AV,EAAiC,2CAAjC,C;K;IAGX,mC;MAQI,OAAO,WAAW,SAAX,EAAiB,CAAjB,EAAoB,gBAApB,EAA0B,KA
A1B,C;K;IAGX,mF;MAeI,0B;QAAA,aAAkB,C;MAClB,wB;QAAA,WAAgB,SAAK,O;MACrB,sC;QAAA,yBAA
kC,K;MAElC,oCAAa,4BAAmB,UAAnB,EAA+B,QAA/B,EAAyC,SAAK,OAA9C,C;MACb,OAAO,WAAW,SAA
X,EAAiB,UAAjB,EAA6B,QAA7B,EAAuC,sBAAvC,C;K;IAGX,sC;MAQI,OAAO,WAAW,SAAX,EAAiB,CAAj
B,EAAoB,gBAApB,EAA4B,KAA5B,C;K;IAGX,sF;MAeI,0B;QAAA,aAAkB,C;MAClB,wB;QAAA,WAAgB,SA
AK,O;MACrB,sC;QAAA,yBAAkC,K;MAElC,oCAAa,4BAAmB,UAAnB,EAA+B,QAA/B,EAAyC,gBAAzC,C;M
ACb,OAAO,WAAW,SAAX,EAAiB,UAAjB,EAA6B,QAA7B,EAAuC,sBAAvC,C;K;uFAGX,qB;MAMwD,OAA
A,SAAY,c;K;mFAEpE,qB;MAWsD,OAAA,SAAY,c;K;uFAElE,qB;MAMwD,OAAA,SAAY,c;K;mFAEpE,qB;M
AWsD,OAAA,SAAY,c;K;yFAElE,qC;MACoF,OAAA,SAAY,SAAQ,GAAR,EAAa,SAAb,C;K;iGAEhG,qC;MAC
wF,OAAA,SAAY,aAAY,GAAZ,EAAiB,SAAjB,C;K;+FAEpG,kC;MAWiF,OAAA,SAAY,YAAW,CAAX,EAAc,
QAAd,C;K;2FAE7F,wB;MAgBgE,OAAA,SAAY,UAAS,CAAT,C;K;iFAE5E,iC;MACqE,OAAA,SAAY,WAAU,
UAAV,C;K;mFAEjF,2C;MACoF,OAAA,SAAY,WAAU,UAAV,EAAsB,QAAtB,C;K;4EAEhG,0B;MAGuD,OAA
A,SAAY,QAAO,GAAP,C;K;wEAEnE,4B;MAGgE,OAAA,SAAY,OAAM,KAAN,C;K;yFAK5E,2C;MACyF,OAA
A,SAAY,SAAQ,OAAR,EAAiB,WAAjB,C;K;IAErG,iD;MAOkD,0B;QAAA,aAAsB,K;MACpE,IAAI,UAAJ,C;Q
ACI,SAAS,SAAK,O;QACd,SAAS,KAAM,O;QACf,UTtBG,MAAO,KSsBM,ETtBN,ESsBU,ETtBV,C;QSuBV,IA
AI,QAAO,CAAX,C;UAAc,OAAO,KAAK,EAAL,I;QACrB,iBAAc,CAAd,UAAsB,GAAtB,U;UACI,eAAe,qBAA
K,KAAL,C;UACf,gBAAgB,iBAAM,KAAN,C;UAEhB,IAAI,aAAY,SAAhB,C;YACI,WAAoB,cAAT,QAAS,C;Y
ACpB,YAAsB,cAAV,SAAU,C;YAEtB,IAAI,aAAY,SAAhB,C;cACwB,kBAAT,Q;cAAX,WD3P2C,gCAAY,cAfr
B,YAAY,CAAZ,C;cC2QZ,kBAAV,S;cAAZ,YD5P2C,gCAAY,cAfrB,YAAY,CAAZ,C;cC6QlC,IAAI,aAAY,SAA
hB,C;gBACI,OAAgB,iBAAT,QAAS,EAAU,SAAV,C;;;;;QAKhC,OAAO,KAAK,EAAL,I;;QAEP,OAAO,4BAAU,
KAAV,C;;K;IAIf,4C;MAOqF,oCAAkB,KAAlB,C;K;IAErF,wD;MASI,OAAW,UAAJ,GACE,4BAAL,SAAK,EA
4B,KAA5B,CADF,GAGE,kBAAL,SAAK,EAAkB,KAAlB,C;K;IAIkD,oD;MAAU,OAAE,UAAF,CAAE,EAAU,C

AAV,EAA0B,IAA1B,C;K;;IAIvE,+C;MAAQ,oC;K;2F2GxUZ,oC;MACiF,O3G2Me,kB2G3ME,oBAAH,EAAG,C
3G2MF,E2G3Mc,S3G2Md,C;K;mG2GzMhG,oC;MACqF,O3G2Me,sB2G3MM,oBAAH,EAAG,C3G2MN,E2G3M
kB,S3G2MlB,C;K;I2GzMpG,mD;MAIoD,0B;QAAA,aAAsB,K;MACtE,IAAI,CAAC,UAAL,C;QACI,O3GgNqF,q
B2GhN7D,M3GgN6D,E2GhNrD,C3GgNqD,C;;Q2G9MrF,OAAO,yBAAc,CAAd,EAAiB,MAAjB,EAAyB,CAAz
B,EAA4B,MAAO,OAAnC,EAA2C,UAA3C,C;K;IAGf,iE;MAIqE,0B;QAAA,aAAsB,K;MACvF,IAAI,CAAC,UA
AL,C;QACI,O3GqMqF,qB2GrM7D,M3GqM6D,E2GrMrD,U3GqMqD,C;;Q2GnMrF,OAAO,yBAAc,UAAd,EAA0
B,MAA1B,EAAkC,CAAlC,EAAqC,MAAO,OAA5C,EAAoD,UAApD,C;K;IAGf,iD;MAIkD,0B;QAAA,aAAsB,K;
MACpE,IAAI,CAAC,UAAL,C;QACI,O3G4MoE,mB2G5M9C,M3G4M8C,C;;Q2G1MpE,OAAO,yBAAc,mBAAS
,MAAO,OAAhB,IAAd,EAAsC,MAAtC,EAA8C,CAA9C,EAAiD,MAAO,OAAxD,EAAgE,UAAhE,C;K;IAGf,mC;
MAGI,aACa,S3GmN2D,O2GnNhD,K3GmNgD,C;M2GlNxE,OAAO,kBAAkB,MAAO,OAAP,KAAe,C;K;IAG5C,
4B;MAKoD,gCAAU,C;MAAV,U;QAAuB,kBAAR,yB;QAAQ,c;;UlHioDvD,U;UADhB,IAAI,0CAAsB,qBAA1B,
C;YAAqC,aAAO,I;YAAP,e;;UACrB,+B;UAAhB,OAAgB,gBAAhB,C;YAAgB,2B;YAAM,IAAI,CkHjoD4D,aAA
T,qBlHioDxC,OkHjoDwC,CAAS,ClHioDhE,C;cAAyB,aAAO,K;cAAP,e;;;UAC/C,aAAO,I;;;QkHloDgE,iB;;MAA
vB,W;K;IAEpD,gD;MASiD,0B;QAAA,aAAsB,K;MAOxC,Q;MAN3B,IAAI,iBAAJ,C;QAAkB,OAAO,a;MACzB,I
AAI,aAAJ,C;QAAmB,OAAO,K;MAC1B,IAAI,CAAC,UAAL,C;QAAiB,OAAO,kBAAQ,KAAR,C;MAExB,IAAI,
SAAK,OAAL,KAAe,KAAM,OAAzB,C;QAAiC,OAAO,K;MAEb,OAAL,SAAK,O;MAA3B,iBAAc,CAAd,wB;Q
ACI,eAAe,qBAAK,KAAL,C;QACf,gBAAgB,iBAAM,KAAN,C;QAChB,IAAI,CAAU,SAAT,QAAS,EAAO,SAA
P,EAAkB,UAAlB,CAAd,C;UACI,OAAO,K;;;MAIf,OAAO,I;K;IAIX,sF;MACkH,0B;QAAA,aAAsB,K;MACpI,oC
AAkB,UAAlB,EAA8B,KAA9B,EAAqC,WAArC,EAAkD,MAAlD,EAA0D,UAA1D,C;K;IAGJ,+B;MAYI,OrGm
MmD,mBAAS,CqGnM5D,G3GiJ4F,oB2GjJzD,C3GiJyD,E2GjJtD,C3GiJsD,CAhE9B,c2GjFrC,G3G8IoD,oB2G9I
Z,C3G8IY,C2G9I7E,GAAyE,S;K;IAG7E,iC;MASI,OrGuLmD,mBAAS,CqGvL5D,G3GqI4F,oB2GrIzD,C3GqIyD,
E2GrItD,C3GqIsD,CA3C9B,c2G1FrC,G3GkIoD,oB2GlIZ,C3GkIY,C2GlI7E,GAAyE,S;K;IAG7E,8B;MAOiB,IAA
N,I;MxH/FP,IAAI,EwH8FI,KAAK,CxH9FT,CAAJ,C;QACI,cwH6Fc,oD;QxH5Fd,MAAM,gCAAyB,OAAQ,WAA
jC,C;;MwH6FH,QAAM,CAAN,C;aACH,C;UAAK,S;UAAL,K;aACA,C;UAAU,OAAL,SAAK,W;UAAV,K;;UAEI
,aAAa,E;UACb,IAAI,ErGgKoC,qBAAU,CqGhK9C,CAAJ,C;YACI,QAAQ,SAAK,W;YACb,YAAY,C;YACZ,OA
AO,IAAP,C;cACI,IAAI,CAAC,QAAU,CAAX,MAAiB,CAArB,C;gBACI,UAAU,C;;cAEd,QAAQ,UAAW,C;cAC
nB,IAAI,UAAS,CAAb,C;gBACI,K;;cAEJ,KAAK,C;;;;UAGb,OAAO,M;;MAnBf,W;K;IAwBJ,4D;MAOqE,0B;QA
AA,aAAsB,K;MACvF,O3G2GiG,kB2G3GnF,WAAO,6BAAM,gBAAO,QAAP,CAAb,EAAmC,UAAJ,GAAgB,K
AAhB,GAA2B,IAA1D,C3G2GmF,E2G3GlB,6BAAM,iCAAwB,QAAxB,C3G2GY,C;K;I2GzGrG,4D;MAM+D,0B
;QAAA,aAAsB,K;MACjF,O3GkGiG,kB2GlGnF,WAAO,6BAAM,gBAAe,oBAAR,OAAQ,CAAf,CAAb,EAA6C,
UAAJ,GAAgB,KAAhB,GAA2B,IAApE,C3GkGmF,E2GlGA,oBAAR,OAAQ,C3GkGA,C;K;I2GhGrG,iE;MAC0E,
0B;QAAA,aAAsB,K;MAC5F,O3G8FiG,kB2G9FnF,WAAO,6BAAM,gBAAO,QAAP,CAAb,EAAmC,UAAJ,GAA
gB,IAAhB,GAA0B,GAAzD,C3G8FmF,E2G9FpB,6BAAM,iCAAwB,QAAxB,C3G8Fc,C;K;I2G5FrG,iE;MACoE,0
B;QAAA,aAAsB,K;MACtF,O3G0FiG,kB2G1FnF,WAAO,6BAAM,gBAAe,oBAAR,OAAQ,CAAf,CAAb,EAA6C,
UAAJ,GAAgB,IAAhB,GAA0B,GAAnE,C3G0FmF,E2G1FF,oBAAR,OAAQ,C3G0FE,C;K;I4GtQrG,kD;MAEI,IA
AI,gBAAJ,C;QAAsB,MAAM,6BAAyB,qCAAkC,QAAQ,CAAR,IAAlC,CAAzB,C;MAC5B,OAAO,CAAC,IAAD,
I;K;IAGX,iF;MAQI,IAAI,EAAS,KAAT,oBAAiB,KAAjB,KAA2B,SAAS,QAAxC,C;QACI,OAAO,UAAU,CAAV,
EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,UAAU,kBAAO,KAAP,C1GyBgC,I;M0GxB1C,IAAI,EAAQ,KAAR,kB
AAgB,KAAhB,CAAJ,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,OAAO,SAAW,C
AAC,OAAS,IAAV,KAAqB,EAAhC,IAAwC,MAAQ,I;K;IAG3D,yE;MAQI,IAAI,SAAU,EAAV,MAAkB,CAAlB,I
AAuB,SAAS,QAApC,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,YAAY,KAAa,C
AAP,KAAO,C;MACzB,IAAI,SAAU,GAAV,MAAkB,GAAtB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAo
B,gBAApB,C;;MAEX,OAAQ,SAAU,CAAX,GAAkB,KAAlB,GAA4B,I;K;IAGvC,yE;MASI,IAAI,SAAS,QAAb,C
;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGX,YAAY,KAAa,CAAP,KAAO,C;MACzB,
IAAI,SAAU,EAAV,MAAiB,CAArB,C;QACI,IAAI,SAAU,GAAV,MAAkB,GAAtB,C;UAEI,OAAO,UAAU,CAA
V,EAAa,KAAb,EAAoB,gBAApB,C;;aAER,IAAI,SAAU,EAAV,MAAiB,EAArB,C;QACH,IAAI,SAAU,GAAV,M
AAkB,GAAtB,C;UAEI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;aAER,IAAI,SAAU,GAAV,MA
AkB,GAAtB,C;QACH,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGX,IAAI,SAAQ,CAAR,UA

Aa,QAAjB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,YAAY,KAAiB,CAAX,QA
AQ,CAAR,IAAW,C;MAC7B,IAAI,SAAU,GAAV,MAAkB,GAAtB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,
EAAoB,gBAApB,C;;MAGX,OAAQ,SAAU,EAAX,GAAoB,SAAU,CAA9B,GAAqC,KAArC,GAA+C,O;K;IAG1
D,yE;MASI,IAAI,SAAS,QAAb,C;QACI,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGJ,YAAY,KAAa,
CAAP,KAAO,C;MACzB,IAAI,SAAU,EAAV,MAAiB,CAArB,C;QACI,IAAI,SAAU,GAAV,KAAkB,GAAtB,C;U
AEI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;aAER,IAAI,SAAU,EAAV,MAAiB,CAArB,C;QAC
H,IAAI,SAAU,GAAV,MAAkB,GAAtB,C;UAEI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;aAER,
IAAI,SAAU,EAAV,IAAgB,CAApB,C;QACH,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;aACJ,IA
AI,SAAU,GAAV,MAAkB,GAAtB,C;QACH,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGX,IA
AI,SAAQ,CAAR,UAAa,QAAjB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,YAA
Y,KAAiB,CAAX,QAAQ,CAAR,IAAW,C;MAC7B,IAAI,SAAU,GAAV,MAAkB,GAAtB,C;QACI,OAAO,UAAU,
CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAGX,IAAI,SAAQ,CAAR,UAAa,QAAjB,C;QACI,OAAO,UAAU,CA
AV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,YAAY,KAAiB,CAAX,QAAQ,CAAR,IAAW,C;MAC7B,IAAI,SA
AU,GAAV,MAAkB,GAAtB,C;QACI,OAAO,UAAU,CAAV,EAAa,KAAb,EAAoB,gBAApB,C;;MAEX,OAAQ,S
AAU,EAAX,GAAoB,SAAU,EAA9B,GAAuC,SAAU,CAAjD,GAAwD,KAAxD,GAAkE,O;K;;;IAmB7E,oE;MAk
B0B,UAGJ,MAHI,EAKJ,MALI,EAMJ,MANI,EASJ,MATI,EAUJ,MAVI,EAWJ,MAXI,EAgBA,MAhBA,EAiBA,
MAjBA,EAkBA,MAlBA,EAoBA,MApBA,EAqBA,OArBA,EAsBA,OAtBA,EAuBA,O;MzH9JtB,IAAI,EyHgII,cA
Ac,CAAd,IAAmB,YAAY,MAAO,OAAtC,IAAgD,cAAc,QzHhIlE,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAy
B,OAAQ,WAAjC,C;;MyHgIV,YAAY,cAAU,CAAC,WAAW,UAAX,IAAD,IAA0B,CAA1B,IAAV,C;MACZ,gBA
AgB,C;MAChB,gBAAgB,U;MAEhB,OAAO,YAAY,QAAnB,C;QACI,WAAW,mBAAO,gBAAP,EAAO,wBAAP,
Q1GzH2B,I;Q0G2HlC,WAAO,GAAP,C;UACI,MAAM,kBAAN,EAAM,0BAAN,YAA0B,OAAL,IAAK,C;eAC9B
,WAAO,IAAP,C;UACI,MAAM,kBAAN,EAAM,0BAAN,YAA4C,OAArB,QAAS,CAAV,GAAgB,GAAM,C;UAC
5C,MAAM,kBAAN,EAAM,0BAAN,YAA+C,OAAxB,OAAS,EAAV,GAAmB,GAAM,C;eAEnD,WAAO,KAAP,I
AAiB,QAAQ,KAAzB,C;UACI,MAAM,kBAAN,EAAM,0BAAN,YAA6C,OAAtB,QAAS,EAAV,GAAiB,GAAM,
C;UAC7C,MAAM,kBAAN,EAAM,0BAAN,YAAuD,OAA/B,QAAS,CAAV,GAAiB,EAAlB,GAA2B,GAAM,C;U
ACvD,MAAM,kBAAN,EAAM,0BAAN,YAA+C,OAAxB,OAAS,EAAV,GAAmB,GAAM,C;;UAG/C,gBAAgB,u
BAAuB,MAAvB,EAA+B,IAA/B,EAAqC,SAArC,EAAgD,QAAhD,EAA0D,gBAA1D,C;UAChB,IAAI,aAAa,CA
AjB,C;YACI,MAAM,kBAAN,EAAM,0BAAN,YAAqB,0BAA0B,CAA1B,C;YACrB,MAAM,kBAAN,EAAM,0B
AAN,YAAqB,0BAA0B,CAA1B,C;YACrB,MAAM,kBAAN,EAAM,0BAAN,YAAqB,0BAA0B,CAA1B,C;;YAEr
B,MAAM,kBAAN,EAAM,0BAAN,YAAkD,OAA3B,aAAc,EAAf,GAAsB,GAAM,C;YAClD,MAAM,mBAAN,E
AAM,2BAAN,aAA6D,OAArC,aAAc,EAAf,GAAuB,EAAxB,GAAiC,GAAM,C;YAC7D,MAAM,mBAAN,EAAM
,2BAAN,aAA4D,OAApC,aAAc,CAAf,GAAsB,EAAvB,GAAgC,GAAM,C;YAC5D,MAAM,mBAAN,EAAM,2B
AAN,aAAoD,OAA7B,YAAc,EAAf,GAAwB,GAAM,C;YACpD,6B;;;;MAMhB,OAAW,KAAM,OAAN,KAAc,SA
AlB,GAA6B,KAA7B,GAA8C,UAAN,KAAM,EAAO,SAAP,C;K;;IAQzD,mE;MAiByB,Q;MzH9LrB,IAAI,EyHw
LI,cAAc,CAAd,IAAmB,YAAY,KAAM,OArC,IAA6C,cAAc,QzHxL/D,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,g
CAAyB,OAAQ,WAAjC,C;;MyHwLV,gBAAgB,U;MAChB,oBAAoB,sB;MAEpB,OAAO,YAAY,QAAnB,C;QAC
I,WAAW,KAAmB,CAAb,gBAAa,EAAb,wBAAa,O;QAE1B,YAAQ,CAAR,C;UACI,aAAc,gBAAY,OAAL,IAAK,
CAAZ,C;aAClB,YAAS,CAAT,KAAc,EAAd,C;UACI,WAAW,eAAe,KAAf,EAAsB,IAAtB,EAA4B,SAA5B,EAA
uC,QAAvC,EAAiD,gBAAjD,C;UACX,IAAI,QAAQ,CAAZ,C;YACI,aAAc,gBAAO,gBAAP,C;YACd,yBAAa,CA
AC,IAAD,IAAb,K;;YAEA,aAAc,gBAAY,OAAL,IAAK,CAAZ,C;YACd,wBAAa,CAAb,I;;eAGR,YAAS,CAAT,K
AAc,EAAd,C;UACI,aAAW,eAAe,KAAf,EAAsB,IAAtB,EAA4B,SAA5B,EAAuC,QAAvC,EAAiD,gBAAjD,C;U
ACX,IAAI,UAAQ,CAAZ,C;YACI,aAAc,gBAAO,gBAAP,C;YACd,yBAAa,CAAC,MAAD,IAAb,K;;YAEA,aAAc
,gBAAY,OAAL,MAAK,CAAZ,C;YACd,wBAAa,CAAb,I;;eAGR,YAAS,CAAT,KAAc,EAAd,C;UACI,aAAW,eA
Ae,KAAf,EAAsB,IAAtB,EAA4B,SAA5B,EAAuC,QAAvC,EAAiD,gBAAjD,C;UACX,IAAI,UAAQ,CAAZ,C;YA
CI,aAAc,gBAAO,gBAAP,C;YACd,yBAAa,CAAC,MAAD,IAAb,K;;YAEA,WAAY,MAAD,GAAQ,KAAR,IAAAq
B,EAArB,GAA2B,K;YACtC,UAAW,SAAS,IAAV,GAAoB,K;YAC9B,aAAc,gBAAY,OAAL,IAAK,CAAZ,C;YA
Cd,aAAc,gBAAW,OAAJ,GAAI,CAAX,C;YACd,wBAAa,CAAb,I;;;UAIJ,UAAU,CAAV,EAAa,SAAb,EAAwB,gB
AAxB,C;UACA,aAAc,gBAAO,gBAAP,C;;;MAK1B,OAAO,aAAc,W;K;ICtQzB,uC;MAU2D,OAAwB,CAAxB,2B

AAwB,mBAAS,SAAT,C;K;IAEnF,oC;MAKI,OAAQ,OAAW,mBAAL,SAAK,CAAX,C;K;IAGZ,6C;MAMI,IAAI,
cAAS,SAAb,C;QACI,iBAAsB,SAAY,Y;QAClC,IAAI,kBAAJ,C;UACS,SAAL,eAA+B,iBAAc,SAAd,E;;UAE/B,U
AAW,WAAI,SAAJ,C;;;K;IAUnB,6C;MAC4B,UAAjB,M;MAAP,OAAO,WAAiB,OAAZ,SAAY,YAAjB,4CAA+D
,W;K;IAI9E,iC;MACI,gBAAqB,sB;MACrB,iBAAsB,E;MACtB,kBAA+B,E;MAC/B,uBAAiC,C;K;uDAEjC,qB;M
ACc,qBAAV,SAAU,EAAc,EAAd,EAAkB,EAAlB,C;MACV,OAAO,aAAO,W;K;gDAGlB,qB;MAA6D,gBAAR,c;
MAAQ,c;;QzIu4Y7C,Q;QAAhB,wBAAgB,SAAhB,gB;UAAgB,cAAA,SAAhB,M;UAAsB,IAAc,OyIv4Y+B,czIu4
Y7C,C;YAAwB,aAAO,I;YAAP,e;;;QAC9C,aAAO,K;;;MyIx4Y8C,iB;K;sDAErD,wC;MACI,KAAK,qBAAL,SAA
K,EAAc,MAAd,EAAsB,SAAtB,CAAL,C;QAAyC,M;MAEzC,YAAY,SAAK,M;MACjB,OAAO,aAAP,C;QACI,K
AAM,qBAAN,KAAM,EAAc,MAAd,EAAsB,aAAtB,CAAN,C;UAA8C,M;QAC9C,QAAQ,KAAM,M;;K;sDAItB,
wC;MASgB,IAAiB,IAAjB,EA2BE,M;MAnCd,aAAO,gBAAO,MAAP,CAAe,gBAAO,SAAP,C;MACtB,gBAAgB,
SAAK,W;MACrB,IAAI,eAAQ,SAAR,CAAJ,C;QACI,aAAO,gBAAO,kCAAP,CAA2C,gBAAO,SAAP,CAAkB,gB
AAO,KAAP,C;QACpE,OAAO,K;;MAEH,cAAY,MAAK,SAAL,C;MAEpB,YAAY,CAAiB,OAAZ,SAAY,MAAjB
,2D;MACZ,IAAI,aAAJ,C;QvHyBG,SuHxBwB,WAAN,KAAM,EAAQ,SAAR,C;QAAvB,iBAAoD,KAAK,CAAT,
GAAY,CAAZ,GAAmB,KAAe,gBAAf,I;QACnE,IAAI,eAAc,CAAlB,C;UAAqB,aAAO,gBAAO,SAAP,CAAkB,g
BAAO,IAAP,C;QAC9C,IAAI,evG8MoC,YAAU,CuG9MlD,C;UACI,kBAAW,K;UACX,uBAAgB,U;;UAEhB,QA
AQ,wBAAiB,KAAjB,EAAwB,UAAxB,C;;QAEZ,IAAI,MvGgNuC,UAAS,CuGhNpD,C;UAEuB,U;UAAA,IAAI,e
AAc,CAAlB,C;YAAA,SAAqB,C;;YxG0+BpC,U;YADhB,YAAY,C;YACI,oBwG1+B+C,SxG0+B/C,C;YAAhB,O
AAgB,gBAAhB,C;cAAgB,sC;cAAM,IwG1+BgE,UxG0+BlD,oBwG1+BkD,MAAK,ExG0+BrE,C;gBAAwB,qB;;
YwG1+Bf,SAA4B,IxG2+BpD,KwG3+BoD,I;;UAA/C,yB;U1GyrCC,kB;UADb,YAAY,C;UACC,S0GxrCK,aAAN,
KAAM,C1GwrCL,W;UAAb,OAAa,gBAAb,C;YAAa,wB;Y0GvrCG,I1GurCU,oBAAmB,cAAnB,EAAmB,sBAAn
B,U0GvrCN,gBAAJ,C;cAA2B,aAAO,uB;YAClC,aAAO,gB1GsrCgC,I0GtrChC,CAAa,gBAAO,IAAP,C;;;UAGxB
,aAAO,gBAAO,KAAP,CAAc,gBAAO,IAAP,C;;;QAGzB,aAAO,gBAAO,SAAP,CAAkB,gBAAO,IAAP,C;;MAG7
B,iBAAiB,mC;MACjB,IpIyHoD,CoIzHhD,UpIyHiD,UoIzHrD,C;QACI,uBAAuB,SAAS,M;QACtB,8B;QAAV,OA
AU,gBAAV,C;UAAU,qB;UACJ,qBAAF,CAAE,EAAc,gBAAd,EAAgC,cAAhC,C;;;MAGV,OAAO,I;K;yDAGX,6
B;MAIwB,Q;MAHpB,mBAAwB,C;MACxB,gBAAqB,C;MACrB,mBAAwB,C;MACJ,OtHyIjB,MAAO,KsHzIgB,
eAAS,OAAT,GAAkB,oBAAlB,ItHyIhB,EsHzIiD,KAAM,OAAN,GAAe,UAAf,ItHyIjD,C;MsHzIV,eAAY,CAAZ,
oB;QACI,QAAQ,iBAAY,iBAAN,KAAM,CAAN,GAAkB,GAAlB,IAAN,C;QACR,IAAI,MAAK,2BAAkB,iBAAT
,eAAS,CAAT,GAAqB,GAArB,IAAT,CAAT,C;UAA6C,K;QAC7C,IAAI,MAAK,EAAT,C;UACI,8BAAgB,CAAh
B,I;UACA,eAAe,S;UACf,YAAY,G;;;MAGpB,IAAI,gBAAgB,CAApB,C;QAAuB,OAAO,K;MAC9B,OAAO,eAA
e,CAAf,IAAoB,iBAAY,iBAAN,KAAM,CAAN,IAAmB,YAAnB,GAAkC,CAAlC,KAAN,MAA+C,EAA1E,C;QA
CI,8BAAgB,CAAhB,I;MAGJ,OAAa,YAAN,KAAM,EAAS,YAAT,CAAN,IAA+B,cAAW,eAAe,CAAf,IAAX,uC
AA/B,C;K;;yHC/H+C,Y;MAAQ,W;K;IAEtE,gD;MACkB,UAMP,M;MANO,IAAI,aAAY,CAAhB,C;QACV,Y;;Q
AEA,UxBuZ8C,MAAW,KwBvZ/C,IxBuZ+C,EwBvZtC,QxBuZsC,C;QwBtZzD,OAAA,IAAO,OxB2UmC,MAAW
,KwB3UpC,KxB2UoC,CwB3UxC,GAAa,GAAnB,CAAP,GAAiC,GAAjC,GxBwV2C,WwBxVC,KxBwVD,C;;Mw
B5V/C,kB;MAMO,IxByUuC,MAAW,KwBzU1C,OxByU0C,CwBzU9C,GAAe,MAAnB,C;QAEmC,SAA9B,OAA
Y,SAAQ,QAAR,C;;QAGpB,exBoU0C,MAAW,KwBpUlC,OxBoUkC,C;QwBnUrD,qBAA8B,QAAY,axBgRC,MA
AW,MAvCV,YwBzOqB,QxByOrB,CAuCU,CwBhRA,GAAAwB,QAApC,C;QAC1C,SAAI,UAAU,CAAd,GAAiB,
MAAG,cAApB,GAAyC,c;;MAP7C,a;K;IAWJ,6C;MACI,OAAa,KAAY,gBAAe,OAAf,EAAwB,MAAK,4BAA2B,
QAA3B,CAAL,EAAxB,C;K;ICtBQ,4C;MAFrC,e;MAEsC,0B;MAFtC,iB;MAAAA,uB;K;IAAAA,mC;MAAAA,sC;O;M
AGI,uEAGY,GAHZ,C;MAIA,yEAGa,MAHb,C;MAIA,yEAGa,SAHb,C;MAIA,+DAGQ,KAHR,C;MAIA,+DAGQ
,MAHR,C;MAIA,2DAGM,MAHN,C;MAIA,yDAGK,OAHL,C;K;;IAxBA,gD;MAAAA,yB;MAAAA,wC;K;;IAIA,iD;
MAAA,yB;MAAA,yC;K;;IAIA,iD;MAAA,yB;MAAA,yC;K;;IAIA,4C;MAAA,yB;MAAA,oC;K;;IAIA,4C;MAAA
,yB;MAAA,oC;K;;IAIA,0C;MAAA,yB;MAAA,kC;K;;IAIA,yC;MAAA,yB;MAAA,iC;K;;IA3BJ,+B;MAAA,4Q;K
;;IAAA,oC;MAAA,a;aAAA,a;UAAA,6C;aAAA,c;UAAA,8C;aAAA,c;UAAA,8C;aAAA,S;UAAA,yC;aAAA,S;UA
AA,yC;aAAA,O;UAAA,uC;aAAA,M;UAAA,sC;;UAAA,6D;;K;;IAiCA,4D;MAGW,Q;MADP,0BAA2C,iBAAjB,
UAAW,cAAM,EAAU,UAAW,cAArB,C;MAEvC,0BAAsB,CAAtB,C;QAA2B,gBAAS,UAAW,cAAX,GAAmB,U
AAW,cAAvC,C;WAC3B,0BAAsB,CAAtB,C;QAA2B,gBAAS,UAAW,cAAX,GAAmB,UAAW,cAAvC,C;;QACn
B,Y;MAHZ,W;K;IAOJ,oE;MAGW,Q;MADP,0BAA2C,iBAAjB,UAAW,cAAM,EAAU,UAAW,cAArB,C;MAEvC

,0BAAsB,CAAtB,C;QAA2B,sBAA8C,uBAArC,UAAW,cAAX,GAAmB,UAAW,cAAO,CAA9C,C;WAC3B,0BAAsB,CAAtB,C;QAA2B,iBAA8C,uBAArC,UAAW,cAAX,GAAmB,UAAW,cAAO,CAA9C,C;;QACnB,Y;MAHZ,W;K;IAOJ,8D;MAGW,Q;MADP,0BAA2C,iBAAjB,UAAW,cAAM,EAAU,UAAW,cAArB,C;MAEvC,0BAAsB,CAAtB,C;QACI,YAAkD,uBAArC,UAAW,cAAX,GAAmB,UAAW,cAAO,C;QAClD,aAAa,eAAQ,KAAR,C;QAET,sBAAS,KAAT,GAAkB,KAAlB,E;UAA2B,a;aAC3B,uBAAQ,CAAR,C;;;;aAIR,0BAAsB,CAAtB,C;QAA2B,iBAA8C,uBAArC,UAAW,cAAX,GAAmB,UAAW,cAAO,CAA9C,C;;QACnB,Y;MAXZ,W;K;;;;ICvCJ,+B;MAAA,mC;MAWiB,wB;MANT,aAyB,OAAO,OAAQ,KAAI,WAAY,IAAG,OAAO,SAA5C,IAAyD,CAAC,CAAC,OAAO,SAAS,K;MADrG,sBAGQ,MAHR,GAIQ,iBAAa,OAAb,CAJR,GAMQ,qBACK,OADF,OAAO,IAAK,KAAI,WAAnB,GAAiC,IAAjC,GAAwC,UAAxC,4GAIO,+B;K;4CAGf,Y;MAA+C,OAAA,mBAAa,U;K;wDAC5D,oB;MAAqE,OAAA,mBAAa,qBAAY,QAAZ,C;K;8DAClF,wB;MAA8F,OAAA,mBAAa,2BAAkB,GAAlB,EAAuB,OAAvB,C;K;0DAE3G,8B;MACI,OAAA,mBAAa,uBAAc,QAAd,EAAwB,QAAxB,C;K;;;IAtBrB,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;IA+B2B,+B;MAAC,wB;K;qCAExB,Y;MAAwC,8CAAc,cAAQ,SAAtB,C;K;iDACxC,oB;MAEmB,IAAS,I;M1HmDrB,Q0HnDH,cAAQ,QAAO,eAAS,OAAT,QAAS,gBAAT,uBAAP,C;MACI,c5IlCT,EAAI,CAAJ,C;M4IkCkB,Y5IiElB,EAAI,CAAJ,C;M4InEH,OAEuC,aAAR,OAAQ,qCAAR,aAAiD,aAAN,KAAM,yCAAjD,C;K;uDAEnC,wB;MAEmB,IAAI,IAAJ,EACQ,M;mBADR,eAAI,OAAJ,GAAI,gBAAJ,uB;MAAV,S5ItCF,OAAI,CAAJ,C;M4IsCM,S5I6DN,OAAI,CAAJ,C;mB4I5DY,eAAQ,SAAR,OAAQ,gBAAR,yB;MAAV,S5IvCF,OAAI,CAAJ,C;M4IuCM,S5I4DN,OAAI,CAAJ,C;M4I3DH,OAAO,CAAK,OAAM,EAAN,IAAY,OAAM,EAAtB,GAA0B,gCAAS,KAAnC,GAAuD,aAAT,KAAK,EAAI,qCAAxD,cAAsG,aAAT,KAAK,EAAI,yCAAtG,C;K;mDAGX,8B;MAEK,IAAS,I;M1HuCP,Q0HvCF,eAAS,OAAT,QAAS,gBAAT,uB;MAA0C,c5I7CxC,EAAI,CAAJ,C;M4I6CiD,Y5IsDjD,EAAI,CAAJ,C;MUklBW,uB;MAAP,eAAuB,6B;MkItoBtB,oBAAoB,YAAY,U1BuPO,Y0BvPqB,6D1BuPrB,C0BvPnB,C;MAH5B,8CAIQ,CAAgB,aAAhB,EAAiD,SAAd,aAAc,CAAlB,GAA8B,QAAQ,QAAtC,GAAoD,GAAnF,CAJR,C;K;sCASJ,Y;MAAkC,qC;K;;IAKF,4C;MAAC,8B;K;6CAEjC,Y;MAA6B,OAAA,gBAAY,M;K;8CAEzC,Y;MAAwC,8CAAc,aAAd,C;K;0DACxC,oB;MAAwE,IAAS,I;MAAnB,OlIgCZ,akIhCa,iBAAS,QAAS,OAAT,QAAS,gBAAT,oCAAT,ClIgCb,4B;K;gEkI9BlD,wB;MACc,IAAI,IAAJ,EACQ,M;MADlB,UAAU,QAAI,OAAJ,GAAI,gBAAJ,oC;MACV,UAAU,QAAQ,SAAR,OAAQ,gBAAR,sC;MACV,OAAW,QAAO,GAAlB,GAAuB,gCAAS,KAAhC,GlI2B8C,akI3BH,MAAM,GlI2BH,4B;K;4DkIxBlD,8B;MAC8B,IAAS,I;MAAnC,8CAAc,YAAY,SAAS,OAAT,QAAS,gBAAT,wCAA6B,QAAS,0DAAlD,CAAAd,C;K;+CAEJ,Y;MAAkC,2C;K;;IAGtC,6B;MAAA,iC;K;yCAGI,Y;MAA6B,OAAe,U;K;0CAE5C,Y;MAAwC,8CAAc,aAAd,C;K;sDACxC,oB;MAAwE,IAAS,I;MAAnB,OlIYZ,akIZa,iBAAS,QAAS,OAAT,QAAS,gBAAT,oCAAT,ClIYb,4B;K;4DkIVlD,wB;MACc,IAAI,IAAJ,EACQ,M;MADlB,UAAU,QAAI,OAAJ,GAAI,gBAAJ,oC;MACV,UAAU,QAAQ,SAAR,OAAQ,gBAAR,sC;MACV,OAAW,QAAO,GAAlB,GAAuB,gCAAS,KAAhC,GlIO8C,akIPH,MAAM,GlIOH,4B;K;wDkIJlD,8B;MAC8B,IAAS,I;MAAnC,8CAAc,YAAY,SAAS,OAAT,QAAS,gBAAT,wCAA6B,QAAS,0DAAlD,CAAAd,C;K;2CAEJ,Y;MAAkC,+B;K;;;IAjBtC,yC;MAAA,wC;QAAA,uB;;MAAA,iC;K;IAoBA,4B;MAA8D,IAAO,QAApB,KAAoB,CAAP,C;QAAAgB,M AAM,gCAAyB,uCAAzB,C;MAAAnC,Y;K;ICl HjD,gD;MAQ+B,kBAApB,wBAAAc,IAAAd,C;MAA0B,I3HgEjC,a;M2HhEA,O3HiEO,W;K;I2H9DX,gD;MAQqD,kBAA1B,gBAAhB,sCAAgB,EAAAc,IAAAd,EAAAoB,IAApB,C;MAAAiC,sB3HoElD,W2HpEkD,C;MAAAxD,O3HqEO,W;K;I4HzFX,yC;MAEkD,8B;MAAAA,OCGN,aDHwB,yBAAa,QAAb,mCCGxB,C7G+xBgC,sB;K;I4GhyB5E,2C;M9IugIW,kBAAY,gB;MAoGH,Q;MAAhB,wB8IpmIqB,U9IomIrB,gB;QAAgB,c8IpmIK,U9IomIrB,M;QAAsB,IAAI,C8IpmIkB,sB9IomIP,O8IpmIO,C9IomItB,C;UAAyB,WAAY,WAAI,OAAJ,C;;M8IpmI3D,qB9IqmIO,W;M8IpmIP,IzIkNwD,CyIlNpD,czIkNqD,UyIlNzD,C;Q5GgKuC,U;Q4G/JnC,qB5G+JyD,OAAtB,+B4G/Jd,mB5G+Jc,uBAAsB,CAAO,W;QoGkO7C,kBAAhB,sB;QQ/XC,0C;QACA,IAAI,E5G8QoC,0BAAU,C4G9Q9C,CAAJ,C;UACI,2BAAO,GAAP,C;;QAEW,sCAAa,GAAb,C;QALnB,sB5H4DG,WoHoUqC,W;QQzXxC,OAAO,I;;MAGX,OAAO,K;K;IAGX,8C;MAOmB,c;;Q9I45YC,Q;QAAhB,wB8I55YI,U9I45YJ,gB;QAAgB,c8I55YZ,U9I45YJ,M;QAAsB,I8I55YD,sB9I45Ye,O8I55Yf,C9I45YC,C;YAAwB,aAAO,I;YAAP,e;;;QAC9C,aAAO,K;;;M8I75YP,e;QACI,kBAA6B,MAAX,UAAW,C;Q5GyIM,U;Q4GxIb,a5GwImC,OAAtB,+B4GxIvB,mB5GwIuB,uBAAsB,CAAO,W;Q4GxIX,kBC/BjB,aD+BD,MC/BC,C7Gg1C6C,uBAAzB,CAAyB,C;QbjmB9E,kBAAS,gB;QA2FA,U;QAAA,+B;QAAhB,OAAgB,gBAAhB,C;UAAgB,6B;UAAM,IyH3yB4C,4BzH2yB9B,SyH3yB8B,CzH2yB5C,C;YAAwB,WAAY,WAAI,SAAJ,C;;QyH3yBtD,sBAAmF,ezH4yBhF,WyH5yBgF,EAAa,GAAb,C;QACnF,OAAO,I;;MAGX,OAAO,K;K;IEnCP,iC;MAAQ,8BAAY,IAAK,UAAjB,IAA8B,uBAAY,IAAK,mB;K;IAO

vD,oC;MAAQ,8BAAY,IAAK,a;K;ICZ7B,4B;MAGI,OAAO,yBAAP,C;QACI,sBAAY,mCAAZ,C;;K;IAIR,uC;MA
OI,sBAAY,sCAAgB,gBAAe,IAAf,CAA5B,C;MACA,OAAO,S;K;ICbP,4B;MAAQ,mB;K;IACR,mC;MACI,eAAO
,K;K;IAKX,4B;MAAQ,mB;K;IACR,mC;MACI,eAAO,K;K;iHCoBf,sJ;MAEyC,qB;QAAA,QAAkB,I;MAAM,qB;
QAAA,QAAkB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,yB;QAAA,YAAsB,I;MAAM,kC;QAAA,qBAA+B,I;MA
AM,qC;QAAA,wBAAkC,K;MAAO,+C;QAAA,kCAA4C,K;MAAO,4C;QAAA,+BAAyC,K;MACtT,QAAQ,E;MA
CR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,WAAF,IAAi
B,S;MACjB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,uBAAF,IAA6B,qB;MAC7B,EAAE,iCAAF,IAAuC,+B;M
ACvC,EAAE,8BAAF,IAAoC,4B;MACpC,OAAO,C;K;+GAw0BX,wD;MAEwC,6B;QAAA,gBAAyB,E;MAAI,uB
;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/I,QAAQ,E;MACR,EAAE,
eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;M
AChB,OAAO,C;K;6EA6CX,4B;MAE6D,iBAAY,KAAZ,C;K;6EAE7D,mC;MAEoE,UAAY,KAAZ,IAAqB,K;K;6
EAuBzF,4B;MAE8D,iBAAY,KAAZ,C;K;6EAE9D,mC;MAEqE,UAAY,KAAZ,IAAqB,K;K;6EAuB1F,4B;MAEq
E,iBAAY,KAAZ,C;K;6EAErE,mC;MAE4E,UAAY,KAAZ,IAAqB,K;K;6EAuBjG,4B;MAE+D,iBAAY,KAAZ,C;
K;6EAE/D,mC;MAEsE,UAAY,KAAZ,IAAqB,K;K;6EAuB3F,4B;MAEgE,iBAAY,KAAZ,C;K;6EAEhE,mC;MAE
uE,UAAY,KAAZ,IAAqB,K;K;6EAuB5F,4B;MAE6D,iBAAY,KAAZ,C;K;6EAE7D,mC;MAEoE,UAAY,KAAZ,I
AAqB,K;K;6EAuBzF,4B;MAE8D,iBAAY,KAAZ,C;K;6EAE9D,mC;MAEqE,UAAY,KAAZ,IAAqB,K;K;6EAuB1
F,4B;MAEiE,iBAAY,KAAZ,C;K;6EAEjE,mC;MAEwE,UAAY,KAAZ,IAAqB,K;K;8EAuB7F,4B;MAEkE,iBAA
Y,KAAZ,C;K;6EAElE,mC;MAEyE,UAAY,KAAZ,IAAqB,K;K;6GC3oC9F,wD;MAEqC,6B;QAAA,gBAA+B,I;M
AAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpJ,QAAQ,E;MAC
R,EAAE,eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IA
AgB,Q;MAChB,OAAO,C;K;mIAiCX,+B;MAEgD,mC;QAAA,sBAAgC,K;MAC5E,QAAQ,E;MACR,EAAE,qBA
AF,IAA2B,mB;MAC3B,OAAO,C;K;4EC9CX,4B;MAEgE,iBAAY,KAAZ,C;K;4EAgChE,4B;MAEyE,iBAAY,KA
AZ,C;K;4EAiBzE,4B;MAEmE,iBAAY,KAAZ,C;K;4EAyYnE,4B;MAE0E,iBAAY,KAAZ,C;K;oIC7a1E,4H;MAE
8C,qB;QAAA,QAAiB,E;MAAI,6B;QAAA,gBAAgC,E;MAAW,iC;QAAA,oBAA2D,E;MAAW,iC;QAAA,oBAA2
D,E;MAAW,qC;QAAA,wBAmJvJ,U;;MAnJqO,+B;QAAA,kBAmJrO,U;;MAnJ6S,4B;QAAA,eAA+B,S;MAC3a,Q
AAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,eAAF,IAAqB,a;MACrB,EAAE,mBAAF,IAAyB,iB;MACzB,
EAAE,mBAAF,IAAyB,iB;MACzB,EAAE,uBAAF,IAA6B,qB;MAC7B,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,c
AAF,IAAoB,Y;MACpB,OAAO,C;K;wIAYX,mC;MAEgD,2B;QAAA,cAAuB,E;MAAI,0B;QAAA,aAAsB,E;MAC
7F,QAAQ,E;MACR,EAAE,aAAF,IAAmB,W;MACnB,EAAE,YAAF,IAAkB,U;MAClB,OAAO,C;K;8HAkEX,+D;
MAEqG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/K,QAAQ,E;M
ACR,EAAE,aAAF,IAAmB,W;MACnB,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,I
AAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;4HAwBX,iE;MAE0C,4B;QAAA,eAAwB,E;MAA
I,wB;QAAA,WAAyB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,
K;MAC/K,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,SAAF,IAA
e,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;sGAUqE,qB;MAAQ,O
AAW,U;K;sGAEnB,qB;MAAQ,OAAW,U;K;4GAEhB,qB;MAAQ,OAAc,a;K;wGAS1B,qB;MAAQ,OAAY,W;K;0
HAEX,qB;MAAQ,OAAqB,oB;K;kGASnD,qB;MAAQ,OAAS,Q;K;oGAEhB,qB;MAAQ,OAAU,S;K;sGAEjB,qB;
MAAQ,OAAW,U;K;wHAEV,qB;MAAQ,OAAoB,mB;K;wHAE5B,qB;MAAQ,OAAoB,mB;K;kHAE/B,qB;MAA
Q,OAAiB,gB;K;kHAEzB,qB;MAAQ,OAAiB,gB;K;oHASd,qB;MAAQ,OAAkB,iB;K;oHAE1B,qB;MAAQ,OAAk
B,iB;K;oHAE1B,qB;MAAQ,OAAkB,iB;K;wIAEhB,qB;MAAQ,OAA4B,2B;K;4FC1MnI,uD;MAE8B,oB;QAAA,
OAAgB,I;MAAM,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;Q
AAA,WAAqB,K;MAChJ,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,S
AAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;kGAuBX,sE;
MAEiC,6B;QAAA,gBAA8B,I;MAAM,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAA
oB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACvL,QAAQ,E;MACR,EAAE,eAAF,IAAq
B,a;MACrB,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YA
AF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;kGA8DX,8U;MAEiC,uB;QAAA,UAAgB,C;
MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SA

AiB,C;MAAG,uB;QAAA,UAAkB,C;MAAG,6B;QAAA,gBAA8B,I;MAAM,sB;QAAA,SAAkB,I;MAAM,uB;QA
AA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAA
O,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA
2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;Q
AAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,kC;QAAA,qBA+B,K;MAAO,oB;QAAA,OAAgB,I;M
AAM,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAq
B,K;MAC3wB,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O
;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,eAAF,IA
AqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MAChB,EA
AE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAA
wB,gB;MACxB,EAAE,YAAF,IAAkB,U;MAClB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB
,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IA0B,kB;MC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBA
AF,IAAsB,c;MACtB,EAAE,oBAAF,IA0B,kB;MC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;M
ACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;
wGAgDX,kQ;MAEoC,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAA
O,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAA
uB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBA0B,K;MAAO,+B;QAAA,kBA4B,K;MAAO,kC;Q
AAA,qBA+B,K;MAAO,6B;QAAA,gBA0B,K;MAAO,8B;QAAA,iBA2B,K;MAAO,kC;QAAA,qBA+B,K;
MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAu
B,K;MAAO,wB;QAAA,WAAqB,K;MAC7lB,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MAClB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IA0B,kB;MC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IA0B,kB;MC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;kGAsCX,iX;MAEiC,sB;QAAA,SAAkB,G;MAAK,sB;QAAA,SAAkB,G;MAAK,sB;QAAA,SAAkB,G;MAAK,yB;QAAA,YAAkB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UAAkB,C;MAAG,6B
;QAAA,gBAA8B,I;MAAM,sB;QAAA,SAAkB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;M
AAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mB
AA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBA0B,K;MAAO,+B;
QAAA,kBA4B,K;MAAO,kC;QAAA,qBA+B,K;MAAO,6B;QAAA,gBA0B,K;MAAO,8B;QAAA,iBA2B,K;
MAAO,kC;QAAA,qBA+B,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAA
oB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MCr2B,QAAQ,E;MCR,EAAE,QAAF,IAAc,M
;MCd,EAAE,QAAF,IAAc,M;MCd,EAAE,QAAF,IAAc,M;MCd,EAAE,WAAF,IAAiB,S;MCjB,EAAE,
SAAF,IAAe,O;MCf,EAAE,SAAF,IAAe,O;MCf,EAAE,SAAF,IAAe,O;MCf,EAAE,SAAF,IAAe,O;MCf,EA
AE,QAAF,IAAc,M;MCd,EAAE,SAAF,IAAe,O;MCf,EAAE,eAAF,IAAqB,a;MCrB,EAAE,QAAF,IAAc,M;M
ACd,EAAE,SAAF,IAAe,O;MCf,EAAE,UAAF,IAAgB,Q;MChB,EAAE,QAAF,IAAc,M;MCd,EAAE,SAAF,I
Ae,O;MCf,EAAE,kBAAF,IAAwB,gB;MCxB,EAAE,kBAAF,IAAwB,gB;MCxB,EAAE,YAAF,IAAkB,U;M
AClB,EAAE,gBAAF,IAAsB,c;MCtB,EAAE,eAAF,IAAqB,a;MCrB,EAAE,iBAAF,IAAuB,e;MCvB,EAAE,o
BAAF,IA0B,kB;MC1B,EAAE,eAAF,IAAqB,a;MCrB,EAAE,gBAAF,IAAsB,c;MCtB,EAAE,oBAAF,IA0
B,kB;MC1B,EAAE,MAAF,IAAY,I;MCZ,EAAE,QAAF,IAAc,M;MCd,EAAE,SAAF,IAAe,O;MCf,EAAE,
YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MChB,OAAO,C;K;kGA2CX,2B;MAEoC,0B;QAAA,aAAsB,C;MAAG,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACtM,QAAQ,E;MCR,EAAE,MAAF,IAAY,I;MCZ,EAAE,QAAF,IAAc,M;MCd,EAAE,SAAF,IAAe,O;MCf,EAAE,
YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MChB,OAAO,C;K;wGAmDX,4
S;MAEoC,mB;QAAA,MAAe,E;MAAI,oB;QAAA,OAAgB,E;MAAI,wB;QAAA,WAAiB,C;MAAG,sB;QAAA,SA

AmB,K;MAAO,2B;QAAA,cAAwB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,sB;Q
AAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B,K;
MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA,k
BAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,
kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAAoB,K;M
AAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjtB,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MA
CX,EAAE,MAAF,IAAY,I;MACZ,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,I
AAmB,W;MACnB,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,QAAF,IAAc,M;MACd,
EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YA
AF,IAAkB,U;MAClB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;M
ACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAA
E,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O
;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;8GAuBX,6D;MAEuC,oB;
QAAA,OAAgB,E;MAAI,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAAoB,K;MAAO,
0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC7K,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EA
AE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;M
AClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;wECnbX,4B;MAEyE,iBAAY,KAAZ,C;K;wEAEzE,2B;MAE
gG,iBAAY,IAAZ,C;K;wEAwBhG,oC;MAE+F,UAAY,KAAZ,IAAqB,M;K;wEAmFpH,2B;MAEqE,iBAAY,IAAZ,
C;K;wEAErE,kC;MAE2E,UAAY,IAAZ,IAAoB,K;K;wEAssC/F,4B;MAEyE,iBAAY,KAAZ,C;K;wEA0BzE,4B;M
AEyE,iBAAY,KAAZ,C;K;wEAsBzE,4B;MAEuE,iBAAY,KAAZ,C;K;wEAyBvE,4B;MAE6E,iBAAY,KAAZ,C;K;
2FA4C7E,gD;MAEiC,qB;QAAA,QAAiD,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,w
B;QAAA,WAAqB,K;MAClK,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE
,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;uEA+UX,4B;MAEuE,iBAAY,KAAZ,C;
K;wEAEvE,2B;MAE6F,iBAAY,IAAZ,C;K;wEAqN7F,4B;MAEyE,iBAAY,KAAZ,C;K;wEAEzE,oC;MAE2F,UA
AY,KAAZ,IAAqB,M;K;+FAuehH,wD;MAEmC,6B;QAAA,gBAA8B,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;
QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjJ,QAAQ,E;MACR,EAAE,eAAF,IAAqB,a;MACrB,EAAE,
SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;uGAuIX,mB
;MAEuC,uB;QAAA,UAAoB,K;MACvD,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;+HAyCX,iB;
MAEmD,qB;QAAA,QAAkB,I;MACjE,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;+FA0MX,sE;
MAEmC,oB;QAAA,OAAgB,I;MAAM,wB;QAAA,WA0+G4B,S;;MA1+GwB,kB;QAAA,KAAc,E;MAAI,wB;QA
AA,WAAoB,I;MAAM,sB;QAAA,SAAkB,S;MAAW,uB;QAAA,UAAoB,I;MAAM,qB;QAAA,QAAiB,I;MAAM,o
B;QAAA,OAAgB,I;MACnP,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,UAAF,IAAgB,Q;MAChB,EA
AE,IAAF,IAAU,E;MACV,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;
MACf,EAAE,OAAF,IAAa,K;MACb,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;qIAgDX,iB;MAEsD,qB;QAAA,Q
AAkB,I;MACpE,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;+GAkBX,qB;MAE2C,yB;QAAA,Y
AAmB,S;MAC1D,QAAQ,E;MACR,EAAE,SAAF,IAAe,S;MACf,OAAO,C;K;wEAkCX,4B;MAEqF,iBAAY,KAA
Z,C;K;yFAgCrF,4V;MAEgC,4B;QAAA,eAA8B,I;MAAM,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;M
AAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UAAk
B,C;MAAG,6B;QAAA,gBAA8B,I;MAAM,sB;QAAA,SAAkB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QAA
A,WAAqB,K;MAAO,sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAA
O,gC;QAAA,mBAA6B,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0
B,K;MAAO,+B;QAAA,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;Q
AAA,iBAA2B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SAAe,C;MAA
G,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC9yB,QAAQ,E;MACR,
EAAE,cAAF,IAAoB,Y;MACpB,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;
MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,eAAF,IA
AqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MAChB,EA
AE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,kBAAF,IAA

wB,gB;MACxB,EAAE,YAAF,IAAkB,U;MAClB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,eAAF,IAAqB,a;MACrB
,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,eAAF,IAAqB,a;MACrB,EAAE,gBA
AF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;M
ACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;
wEAwEX,2B;MAE+D,iBAAY,IAAZ,C;K;iGA2D/D,gD;MAEoC,qB;QAAA,QAAc,I;MAAM,uB;QAAA,UAAoB,
K;MAAO,0B;QAAA,aAAAuB,K;MAAO,wB;QAAA,WAAAqB,K;MAClI,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;M
ACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;
qGA2BX,yD;MAEsC,sB;QAAA,SAAkB,E;MAAI,sB;QAAA,SAAkB,E;MAAI,uB;QAAA,UAAoB,K;MAAO,0B;
QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC5J,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,
QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;M
AChB,OAAO,C;K;6GAuBX,oD;MAE0C,yB;QAAA,YAAsB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,
aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACjJ,QAAQ,E;MACR,EAAE,WAAF,IAAiB,S;MACjB,EAAE,SAAF,
IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;2FAoFX,kF;MAEi
C,uB;QAAA,UAAmB,E;MAAI,wB;QAAA,WAAoB,E;MAAI,sB;QAAA,SAAe,C;MAAG,qB;QAAA,QAAc,C;M
AAG,qB;QAAA,QAAc,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAq
B,K;MACjN,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,QAAF,IAAc
,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAA
F,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;iHAyBX,0D;MAEqE,sB;QAAA,SAAe,S;MAA
W,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACzK,QAAQ,E;MACR,
EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;
MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;wEAmXX,4B;MAEkE,iBAAY,KAAZ,C;K;wEAElE,2B;M
AEoE,iBAAY,IAAZ,C;K;wEAUpE,4B;MAEsE,iBAAY,KAAZ,C;K;wEAEtE,2B;MAEwE,iBAAY,IAAZ,C;K;wE
AaxE,4B;MAE+D,iBAAY,KAAZ,C;K;wEAE/D,2B;MAEiE,iBAAY,IAAZ,C;K;mGA0CjE,8G;MAEqC,gC;QAAA
,mBAooF8C,M;;MApoFe,gC;QAAA,mBAmpFT,S;;MAnpFyE,oC;QAAA,uBA8pFjE,S;;MA9pF6I,2B;QAAA,cAA
oB,S;MAAW,4B;QAAA,eAAqB,S;MAAW,6B;QAAA,gBAyqFlO,K;;MAxqFvE,QAAQ,E;MACR,EAAE,kBAAF,
IAAwB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,sBAAF,IAA4B,oB;MAC5B,EAAE,aAAF,IAAm
B,W;MACnB,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,eAAF,IAAqB,a;MACrB,OAAO,C;K;+FAwCX,mF;MAEm
C,oB;QAAA,OAAa,I;MAAM,sB;QAAA,SAAkB,E;MAAI,2B;QAAA,cAAuB,E;MAAI,sB;QAAA,SAAyC,I;MAA
M,qB;QAAA,QAA6B,E;MAAW,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB
,K;MACxQ,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,
W;MACnB,EAAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YA
AF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;6FA4BX,2B;MAEkC,+B;QAAA,kBAA4B,K;
MAC1D,QAAQ,E;MACR,EAAE,iBAAF,IAAuB,e;MACvB,OAAO,C;K;2FA2DX,iE;MAEiC,wB;QAAA,WAAqB
,K;MAAO,oB;QAAA,OAAAe,C;MAAG,sB;QAAA,SAAkB,E;MAAI,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aA
AuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/K,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,MAAF,
IAAY,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAA
E,UAAF,IAAgB,Q;MAChB,OAAO,C;K;yFA8FX,6B;MAEgC,oB;QAAA,OA+7E6C,S;;MA/7EL,2B;QAAA,cCl2
He,M;;MDm2HnF,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,aAAF,IAAmB,W;MACnB,OAAO,C;K;
wEAoDX,0B;MAE+D,iBAAY,GAAZ,C;K;wEAE/D,iC;MAEqE,UAAY,GAAZ,IAAmB,K;K;+FAoDxF,oF;MAE
mC,mB;QAAA,MAAe,I;MAAM,wB;QAAA,WAAoB,I;MAAM,wB;QAAA,WAAoB,I;MAAM,mB;QAAA,MAAe
,E;MAAI,2B;QAAA,cAAwB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,W
AAqB,K;MACvO,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,UA
AF,IAAgB,Q;MAChB,EAAE,KAAF,IAAW,G;MACX,EAAE,aAAF,IAAmB,W;MACnB,EAAE,SAAF,IAAe,O;M
ACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;iFAwNX,yC;MAE4B,uB;QA
AA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACtG,QAAQ,E;MACR,EAAE,SA
AF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;6FAwBX,iD;M
AEkC,sB;QAAA,SAAe,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAq
B,K;MACjI,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U

;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;uGASX,mB;MAEuC,uB;QAAA,UAAoB,K;MACvD,QAA
Q,E;MACR,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;6GAYX,kC;MAE0C,uB;QAAA,UAAoB,K;MAAO,oB;QA
AA,OAAiB,K;MAAO,uB;QAAA,UAAoB,K;MAC7G,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,MA
AF,IAAY,I;MACZ,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;wEAkEX,4B;MAE6D,iBAAY,KAAZ,C;K;wEAU7D
,4B;MAEsE,iBAAY,KAAZ,C;K;wEAEtE,2B;MAEwE,iBAAY,IAAZ,C;K;uGAsCxE,oH;MAEuC,yB;QAAA,YAA
sB,K;MAAO,0B;QAAA,aAAuB,S;MAAW,6B;QAAA,gBAA0B,S;MAAW,uB;QAAA,UAAoB,K;MAAO,iC;QAA
A,oBAA8B,S;MAAW,qC;QAAA,wBAAkC,S;MAAW,+B;QAAA,kBAAkC,S;MAC1R,QAAQ,E;MACR,EAAE,
WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MAClB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,SAAF,IAAe,O;
MACf,EAAE,mBAAF,IAAyB,iB;MACzB,EAAE,uBAAF,IAA6B,qB;MAC7B,EAAE,iBAAF,IAAuB,e;MACvB,O
AAO,C;K;mGAgFX,oB;MAEqC,wB;QAAA,WAAqB,K;MACtD,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MAC
hB,OAAO,C;K;wEA+MX,2B;MAEiE,iBAAY,IAAZ,C;K;2GAkCjE,c;MAEyC,kB;QAAA,KAAgB,S;MACrD,QA
AQ,E;MACR,EAAE,IAAF,IAAU,E;MACV,OAAO,C;K;2FAuMX,gB;MAGI,QAAQ,E;MACR,EAAE,MAAF,IA
AY,I;MACZ,OAAO,C;K;wEAgBX,4B;MAEiE,iBAAY,KAAZ,C;K;wEAEjE,oC;MAE4E,iBAAY,aAAZ,C;K;wE
AuT5E,4B;MAEmE,iBAAY,KAAZ,C;K;uFA2CnE,sB;MAE+B,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IAAa,G;MA
AK,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IAAa,G;MAC9F,QAAQ,E;MACR,EAAE,GAAF,IAAS,C;MACT,EAAE
,GAAF,IAAS,C;MACT,EAAE,GAAF,IAAS,C;MACT,EAAE,GAAF,IAAS,C;MACT,OAAO,C;K;qFA0CX,+B;M
AE8B,iB;QAAA,IAAa,G;MAAK,iB;QAAA,IAAa,G;MAAK,qB;QAAA,QAAiB,G;MAAK,sB;QAAA,SAAkB,G;
MACtG,QAAQ,E;MACR,EAAE,GAAF,IAAS,C;MACT,EAAE,GAAF,IAAS,C;MACT,EAAE,OAAF,IAAa,K;M
ACb,EAAE,QAAF,IAAc,M;MACd,OAAO,C;K;wEAOX,4B;MAEmE,iBAAY,KAAZ,C;K;yFAiHnE,oB;MAEgC,
wB;QAAA,WAy2B+C,M;;MAx2B3E,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;6FAeX,+B;
MAEkC,oB;QAAA,OAAgB,S;MAAW,mB;QAAA,MAAe,S;MAAW,wB;QAAA,WAq1BR,M;;MAp1B3E,QAAQ,
E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,KAAF,IAAW,G;MACX,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,
C;K;6GAwCX,yD;MAE0C,qB;QAAA,QAAiB,E;MAAI,uB;QAAA,UAAoB,K;MAAO,uB;QAAA,UAAoB,K;MA
AO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpK,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb
,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB
,Q;MAChB,OAAO,C;K;yGAiCX,mC;MAEwC,qB;QAAA,QA2wByD,Q;;MA3wBK,sB;QAAA,SA2wBL,Q;;MA3
wBoE,wB;QAAA,WA4vBtF,M;;MA3vB3E,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,
M;MACd,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;2FAYX,2B;MAEiC,mB;QAAA,MAuwB0C,Q;;MAvwBJ,
0B;QAAA,aAAsB,S;MACzF,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,YAAF,IAAkB,U;MAClB,O
AAO,C;K;+GAYX,0B;MAE2C,uB;QAAA,UAqvBgC,Q;;MArvBU,qB;QAAA,QAqvBV,Q;;MApvBvE,QAAQ,E;
MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;wEAgCX,4B;MAE+D,iBAAY,K
AAZ,C;K;qFAyaY,qB;MAAQ,OAAU,S;K;6FAEd,qB;MAAQ,OAAc,a;K;uFAEzB,qB;MAAQ,OAAW,U;K;iFASx
B,qB;MAAQ,OAAG,E;K;iFAEX,qB;MAAQ,OAAQ,O;K;uFAEb,qB;MAAQ,OAAW,U;K;uFAS3B,qB;MAAQ,O
AAW,U;K;mFAErB,qB;MAAQ,OAAS,Q;K;qFAEhB,qB;MAAQ,OAAU,S;K;yFAShB,qB;MAAQ,OAAY,W;K;uF
AErB,qB;MAAQ,OAAW,U;K;+FAEf,qB;MAAQ,OAAe,c;K;uFAE3B,qB;MAAQ,OAAW,U;K;uFAEnB,qB;MAA
Q,OAAW,U;K;mFASrB,qB;MAAQ,OAAS,Q;K;iFAElB,qB;MAAQ,OAAQ,O;K;6EAElB,qB;MAAQ,OAAM,K;K
;uFAET,qB;MAAQ,OAAW,U;K;qFASlB,qB;MAAQ,OAAU,S;K;qFAElB,qB;MAAQ,OAAU,S;K;6EASR,qB;MA
AQ,OAAM,K;K;mFAEX,qB;MAAQ,OAAS,Q;K;+EAEnB,qB;MAAQ,OAAO,M;K;+EAS/B,qB;MAAQ,OAAO,M
;K;iFAEd,qB;MAAQ,OAAQ,O;K;mFAEf,qB;MAAQ,OAAS,Q;K;mFAShB,qB;MAAQ,OAAQ,O;K;iFAEhB,qB;
MAAQ,OAAQ,O;K;iFAEhB,qB;MAAQ,OAAQ,O;K;mFASd,qB;MAAQ,OAAQ,O;K;+EAElB,qB;MAAQ,OAAM
,K;K;+EAEb,qB;MAAQ,OAAO,M;K;iFAEd,qB;MAAQ,OAAQ,O;K;mFAEf,qB;MAAQ,OAAS,Q;K;6EASd,qB;
MAAQ,OAAM,K;K;qFAEV,qB;MAAQ,OAAU,S;K;mFAEnB,qB;MAAQ,OAAS,Q;K;2FAEb,qB;MAAQ,OAAa,
Y;K;6FAEpB,qB;MAAQ,OAAc,a;K;mFAE3B,qB;MAAQ,OAAS,Q;K;6EAS1B,qB;MAAQ,OAAM,K;K;6EAEd,q
B;MAAQ,OAAM,K;K;qFAEV,qB;MAAQ,OAAU,S;K;+EASjB,qB;MAAQ,OAAO,M;K;mFAEb,qB;MAAQ,OAA
S,Q;K;+EASrB,qB;MAAQ,OAAO,M;K;iFAEd,qB;MAAQ,OAAQ,O;K;iFASjB,qB;MAAQ,OAAO,M;K;6FAER,q
B;MAAQ,OAAc,a;K;qFAE1B,qB;MAAQ,OAAU,S;K;iFASb,qB;MAAQ,OAAO,M;K;uFAEZ,qB;MAAQ,OAAU,
S;K;yFAS9B,qB;MAAQ,OAAY,W;K;+EAE1B,qB;MAAQ,OAAM,K;K;qFAEX,qB;MAAQ,OAAS,Q;K;iFAEnB,
qB;MAAQ,OAAO,M;K;+EASrB,qB;MAAQ,OAAO,M;K;6FAER,qB;MAAQ,OAAc,a;K;qFAS1B,qB;MAAQ,OA

AU,S;K;mFAEnB,qB;MAAQ,OAAS,Q;K;+EASX,qB;MAAQ,OAAO,M;K;mFAEb,qB;MAAQ,OAAS,Q;K;iFASn
B,qB;MAAQ,OAAO,M;K;qFAEZ,qB;MAAQ,OAAU,S;K;mFAEnB,qB;MAAQ,OAAS,Q;K;kFASJ,qB;MAAQ,O
AAQ,O;K;oFAEf,qB;MAAQ,OAAS,Q;K;8EAEpB,qB;MAAQ,OAAM,K;K;oFAEV,qB;MAAQ,OAAU,S;K;mFAS
zC,qB;MAAQ,OAAS,Q;K;mFAEjB,qB;MAAQ,OAAS,Q;K;qFAEhB,qB;MAAQ,OAAU,S;K;qFAElB,qB;MAAQ,
OAAU,S;K;wIEx+M7E,wM;MAEiD,qB;QAAA,QAAkB,I;MAAM,sB;QAAA,SAAmB,I;MAAM,2B;QAAA,cAA
wB,I;MAAM,yB;QAAA,YAAsB,I;MAAM,0B;QAAA,aAAuB,I;MAAM,0B;QAAA,aAAuB,I;MAAM,sB;QAAA,
SAAmB,I;MAAM,0B;QAAA,aAAuB,I;MAAM,0B;QAAA,aAAuB,I;MAAM,gC;QAAA,mBAA6B,I;MAAM,+B;
QAAA,kBAA4B,I;MAAM,gC;QAAA,mBAA6B,I;MAAM,uB;QAAA,UAAoB,I;MAAM,4B;QAAA,eAAyB,I;MA
AM,wB;QAAA,WAAqB,I;MAAM,uB;QAAA,UAAoB,I;MACrf,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,E
AAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IA
AkB,U;MAClB,EAAE,YAAF,IAAkB,U;MAClB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MAClB,
EAAE,YAAF,IAAkB,U;MAClB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kB
AAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;
MAChB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;wHAsDX,wM;MAEyC,qB;QAAA,QAAqB,S;MAAW,sB;QAA
A,SAAsB,S;MAAW,2B;QAAA,cAA4B,S;MAAW,yB;QAAA,YAA0B,S;MAAW,0B;QAAA,aAA6B,S;MAAW,0
B;QAAA,aAA6B,S;MAAW,sB;QAAA,SAAuB,S;MAAW,0B;QAAA,aAA0B,S;MAAW,0B;QAAA,aAA0B,S;MA
AW,gC;QAAA,mBAAoC,S;MAAW,+B;QAAA,kBAAmC,S;MAAW,gC;QAAA,mBAAoC,S;MAAW,uB;QAAA,
UAAwB,S;MAAW,4B;QAAA,eAA4B,S;MAAW,wB;QAAA,WAAoB,S;MAAW,uB;QAAA,UAAmB,S;MACtnB,
QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,E
AAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MAClB,EAAE,YAAF,IAAkB,U;MAClB,EAAE,QAAF,I
AAc,M;MACd,EAAE,YAAF,IAAkB,U;MAClB,EAAE,YAAF,IAAkB,U;MAClB,EAAE,kBAAF,IAAwB,gB;MA
CxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,c
AAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;sHAYX,kN;
MAEwC,wB;QAAA,WAA4C,S;MAAW,qB;QAAA,QAAiB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,2B;QAAA,c
AAuB,S;MAAW,yB;QAAA,YAAqB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,sB;Q
AAA,SAAkB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,gC;QAAA,mBAA4B,S;MAA
W,+B;QAAA,kBAA2B,S;MAAW,gC;QAAA,mBAA4B,S;MAAW,uB;QAAA,UAAmB,S;MAAW,4B;QAAA,eAA
wB,S;MAAW,wB;QAAA,WAAoB,S;MAAW,uB;QAAA,UAAmB,S;MAC9lB,QAAQ,E;MACR,EAAE,UAAF,IA
gB,Q;MAChB,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,E
AAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MAClB,EAAE,YAAF,IAAkB,U;MAClB,EAAE,QAAF,I
AAc,M;MACd,EAAE,YAAF,IAAkB,U;MAClB,EAAE,YAAF,IAAkB,U;MAClB,EAAE,kBAAF,IAAwB,gB;MA
CxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,c
AAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;0HAsDX,wM
;MAE0C,qB;QAAA,QAAiB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,2B;QAAA,cAAuB,S;MAAW,yB;QAAA,YA
AqB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,sB;QAAA,SAAkB,S;MAAW,0B;QAA
A,aAAsB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,gC;QAAA,mBAA4B,S;MAAW,+B;QAAA,kBAA2B,S;MAAW
,gC;QAAA,mBAA4B,S;MAAW,uB;QAAA,UAAmB,S;MAAW,4B;QAAA,eAAwB,S;MAAW,wB;QAAA,WAAo
B,S;MAAW,uB;QAAA,UAAmB,S;MACziB,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;
MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,YAAF,IAAkB,U;MAClB,EA
AE,YAAF,IAAkB,U;MAClB,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MAClB,EAAE,YAAF,IAAk
B,U;MAClB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;M
CxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,SA
AF,IAAe,O;MACf,OAAO,C;K;gHAyDX,wM;MAEqC,qB;QAAA,QAAc,S;MAAW,sB;QAAA,SAAe,S;MAAW,2
B;QAAA,cAAuB,S;MAAW,yB;QAAA,YAAqB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,0B;QAAA,aAAsB,S;MA
AW,sB;QAAA,SAAkB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,0B;QAAA,aAAsB,S;MAAW,gC;QAAA,mBAAA
6B,S;MAAW,+B;QAAA,kBAA4B,S;MAAW,gC;QAAA,mBAA6B,S;MAAW,uB;QAAA,UAAoB,S;MAAW,4B;
QAAA,eAAqB,S;MAAW,wB;QAAA,WAAoB,S;MAAW,uB;QAAA,UAAmB,S;MACxhB,QAAQ,E;MACR,EAA
E,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S

;MACjB,EAAE,YAAF,IAAkB,U;MAClB,EAAE,YAAF,IAAkB,U;MAClB,EAAE,QAAF,IAAc,M;MACd,EAAE,
YAAF,IAAkB,U;MAClB,EAAE,YAAF,IAAkB,U;MAClB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,iBAAF,IA
AuB,e;MACvB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,SAAF,IAAe,O;MACf,EAAE,cAAF,IAAoB,Y;MACpB
,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;8HAqBX,gD;MAEsE,uB;QAAA,UA
AoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAChJ,QAAQ,E;MACR,EAAE,OAAF,IAA
a,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAA
O,C;K;sIAoBX,gD;MAEgD,qB;QAAA,QAAiB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MA
AO,wB;QAAA,WAAqB,K;MACjJ,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,E
AAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;wHAwCX,wB;MAEyC,qB;QAAA,
QAAiB,K;MAAO,qB;QAAA,QAAiB,K;MAC9E,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,I
AAa,K;MACb,OAAO,C;K;kGAyBX,oB;MAE8B,mB;QAAA,MAAe,S;MAAW,mB;QAAA,MAAe,S;MACnE,QA
AQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;oHAYX,kC;MAEuC,q
B;QAAA,QAAiB,S;MAAW,qB;QAAA,QAAiB,S;MAAW,mB;QAAA,MAAe,S;MAAW,mB;QAAA,MAAe,S;MA
CpI,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,KAAF,IAAW,G;MACX
,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;gGAYX,oB;MAE6B,mB;QAAA,MAAY,S;MAAW,mB;QAAA,MA
AY,S;MAC5D,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;kHA
YX,kC;MAEsC,qB;QAAA,QAAc,S;MAAW,qB;QAAA,QAAc,S;MAAW,mB;QAAA,MAAY,S;MAAW,mB;QAA
A,MAAY,S;MACvH,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,KAAF,
IAAW,G;MACX,EAAE,KAAF,IAAW,G;MACX,OAAO,C;K;gIAeX,wB;MAE6C,qB;QAAA,QAAkB,S;MAAW,q
B;QAAA,QAAkB,S;MACxF,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,OAA
O,C;K;oIAeX,wB;MAE+C,qB;QAAA,QAAiB,S;MAAW,qB;QAAA,QAAiB,S;MACxF,QAAQ,E;MACR,EAAE,O
AAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,OAAO,C;K;4FAKX,Y;MAGI,QAAQ,E;MACR,OAAO,C;K;o
FAKX,Y;MAGI,QAAQ,E;MACR,OAAO,C;K;8FAKX,Y;MAGI,QAAQ,E;MACR,OAAO,C;K;kGASX,oB;MAE8
B,wB;QAAA,WAAkC,S;MAC5D,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;4FAUmE,qB;M
AAQ,OAAO,M;K;8FAEd,qB;MAAQ,OAAQ,O;K;4FASrB,qB;MAAQ,OAAO,M;K;0GAER,qB;MAAQ,OAAc,a;
K;8FAE7B,qB;MAAQ,OAAO,M;K;gGAEd,qB;MAAQ,OAAQ,O;K;8FASjB,qB;MAAQ,OAAO,M;K;gHAEL,qB;
MAAQ,OAAiB,gB;K;wGASrC,qB;MAAQ,OAAa,Y;K;0GAEpB,qB;MAAQ,OAAc,a;K;wGAEvB,qB;MAAQ,OA
Aa,Y;K;oFCroB7F,4B;MAE6E,iBAAY,KAAZ,C;K;iGASnB,qB;MAAQ,OAAS,Q;K;6FAEnB,qB;MAAQ,OAAO,
M;K;+FAEd,qB;MAAQ,OAAQ,O;K;iGASF,qB;MAAQ,OAAU,S;K;+FAEnB,qB;MAAQ,OAAS,Q;K;mGAS3B,q
B;MAAQ,OAAW,U;K;mGAEnB,qB;MAAQ,OAAW,U;K;6GC1D/E,mb;MAEmC,yB;QAAA,YAAkB,C;MAAG,q
B;QAAA,QAAiB,G;MAAK,sB;QAAA,SAAkB,G;MAAK,wB;QAAA,WAAmB,G;MAAI,kC;QAAA,qBAA6B,G;
MAAI,qB;QAAA,QAAc,C;MAAG,qB;QAAA,QAAc,C;MAAG,qB;QAAA,QAAc,C;MAAG,2B;QAAA,cAAuB,E;
MAAI,yB;QAAA,YAAsB,K;MAAO,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,uB;QAAA,UA
AgB,C;MAAG,uB;QAAA,UAAgB,C;MAAG,sB;QAAA,SAAiB,C;MAAG,uB;QAAA,UAAkB,C;MAAG,6B;QAA
A,gBAA8B,I;MAAM,sB;QAAA,SAAkB,I;MAAM,uB;QAAA,UAAoB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,
sB;QAAA,SAAmB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,gC;QAAA,mBAA6B,K;MAAO,gC;QAAA,mBAA6B
,K;MAAO,0B;QAAA,aAAuB,K;MAAO,8B;QAAA,iBAA2B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,+B;QAAA
,kBAA4B,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,6B;QAAA,gBAA0B,K;MAAO,8B;QAAA,iBAA2B,K;MAA
O,kC;QAAA,qBAA+B,K;MAAO,oB;QAAA,OAAgB,I;MAAM,sB;QAAA,SAAe,C;MAAG,uB;QAAA,UAAoB,K;
MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACl/B,QAAQ,E;MACR,EAAE,WAAF,IAAiB,S;
MACjB,EAAE,OAAF,IAAa,K;MACb,EAAE,QAAF,IAAc,M;MACd,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,oB
AAF,IAA0B,kB;MAC1B,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MA
Cb,EAAE,aAAF,IAAmB,W;MACnB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,
IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,S
AAF,IAAe,O;MACf,EAAE,eAAF,IAAqB,a;MACrB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,
EAAE,UAAF,IAAgB,Q;MAChB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,kBAAF,IAA
wB,gB;MACxB,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,YAAF,IAAkB,U;MAClB,EAAE,gBAAF,IAAsB,c;M
ACtB,EAAE,eAAF,IAAqB,a;MACrB,EAAE,iBAAF,IAAuB,e;MACvB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE

,eAAF,IAAqB,a;MACrB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,MAAF,IAA
Y,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,U
AAF,IAAgB,Q;MAChB,OAAO,C;K;6GC1BX,0C;MAEwC,oB;QAAA,OAAiB,I;MAAM,sB;QAAA,SAAmB,K;M
AAO,uB;QAAA,UAAoB,K;MAAO,uB;QAAA,UAAoB,K;MACpI,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MAC
Z,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;4EAmIX,
4B;MAEkE,iBAAY,KAAZ,C;K;4EAElE,qC;MAE2E,UAAY,KAAZ,IAAqB,O;K;4EAiBhG,4B;MAEuE,iBAAY,K
AAZ,C;K;4EAEvE,qC;MAE+E,UAAY,KAAZ,IAAqB,O;K;4EAiBpG,4B;MAEuE,iBAAY,KAAZ,C;K;4EAEvE,q
C;MAE+E,UAAY,KAAZ,IAAqB,O;K;4EAiGpG,4B;MAEoE,iBAAY,KAAZ,C;K;2EAEpE,qC;MAE4E,UAAY,K
AAZ,IAAqB,O;K;4EAkcjG,4B;MAE6E,iBAAY,KAAZ,C;K;4EAE7E,qC;MAEqF,UAAY,KAAZ,IAAqB,O;K;4E
AgP1G,4B;MAEqE,iBAAY,KAAZ,C;K;4EAErE,qC;MAE6E,UAAY,KAAZ,IAAqB,O;K;uFJ57BlG,+H;MAE8B,s
B;QAAA,SAAkB,S;MAAW,uB;QAAA,UAAmB,S;MAAW,oB;QAAA,OAAgB,S;MAAW,wB;QAAA,WAAoB,S;
MAAW,8B;QAAA,iBAA0B,S;MAAW,oB;QAAA,OAAqB,S;MAAW,2B;QAAA,cAAmC,S;MAAW,qB;QAAA,Q
AAuB,S;MAAW,wB;QAAA,WAA6B,S;MAAW,yB;QAAA,YAAqB,S;MAAW,yB;QAAA,YAAsB,S;MAAW,wB
;QAAA,WAAe,S;MAC5Z,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,M
AAF,IAAY,I;MACZ,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,gBAAF,IAAsB,c;MACtB,EAAE,MAAF,IAAY,I;M
ACZ,EAAE,aAAF,IAAmB,W;MACnB,EAAE,OAAF,IAAa,K;MACb,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,W
AAF,IAAiB,S;MACjB,EAAE,WAAF,IAAiB,S;MACjB,EAAE,QAAF,IAAc,Q;MACd,OAAO,C;K;yFA0CX,uC;M
AE+B,sB;QAAA,SAAiB,G;MAAK,0B;QAAA,aAAsB,I;MAAM,uB;QAAA,UAAmB,S;MAChG,QAAQ,E;MACR
,EAAE,QAAF,IAAc,M;MACd,EAAE,YAAF,IAAkB,U;MAClB,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;qFAUg
D,qB;MAAQ,OAAG,E;K;mFAEX,qB;MAAQ,OAAQ,O;K;iFAEjB,qB;MAAQ,OAAO,M;K;mFAEd,qB;MAAQ,O
AAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;mFAElB,qB;MAAQ,OAAQ,O;K;mFAEhB,qB;MAAQ,OAAQ,O;K;mF
AEhB,qB;MAAQ,OAAQ,O;K;qFASF,qB;MAAQ,OAAG,E;K;yFAER,qB;MAAQ,OAAW,U;K;mFAEtB,qB;MAA
Q,OAAQ,O;K;mFAEjB,qB;MAAQ,OAAO,M;K;qFAEd,qB;MAAQ,OAAQ,O;K;yFAEb,qB;MAAQ,OAAW,U;K;
mFAEtB,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;qFAEjB,qB;MAAQ,OAAS,Q;K;uFAEjB,qB;MA
AQ,OAAS,Q;K;mGAEV,qB;MAAQ,OAAgB,e;K;iGAEzB,qB;MAAQ,OAAe,c;K;qFAE9B,qB;MAAQ,OAAQ,O;
K;qFAEf,qB;MAAQ,OAAS,Q;K;iFAEnB,qB;MAAQ,OAAO,M;K;yFASzB,qB;MAAQ,OAAW,U;K;+FAEhB,qB;
MAAQ,OAAc,a;K;uFAE1B,qB;MAAQ,OAAU,S;K;iFAErB,qB;MAAQ,OAAO,M;K;iFASD,qB;MAAQ,OAAO,M
;K;iGAER,qB;MAAQ,OAAc,a;K;uFAE1B,qB;MAAQ,OAAU,S;K;yFAS9B,qB;MAAQ,OAAU,S;K;yFAEjB,qB;M
AAQ,OAAW,U;K;qFAErB,qB;MAAQ,OAAS,Q;K;yFAEf,qB;MAAQ,OAAW,U;K;+FAEhB,qB;MAAQ,OAAc,a;
K;qGAEnB,qB;MAAQ,OAAiB,gB;K;qFAS3B,qB;MAAQ,OAAS,Q;K;mFAElB,qB;MAAQ,OAAQ,O;K;uFAEf,q
B;MAAQ,OAAS,Q;K;mFASxB,qB;MAAQ,OAAQ,O;K;mFAEjB,qB;MAAQ,OAAO,M;K;yFAEZ,qB;MAAQ,OA
AU,S;K;qFAEpB,qB;MAAQ,OAAQ,O;K;qFAEf,qB;MAAQ,OAAS,Q;K;qGAET,qB;MAAQ,OAAiB,gB;K;+FKn
R/F,gB;MAEkC,oB;QAAA,OAAgB,E;MAC9C,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;+FAi
BX,8B;MAEkC,4B;QAAA,eAAqB,S;MAAW,oB;QAAA,OAAgB,E;MAC9E,QAAQ,E;MACR,EAAE,cAAF,IAAo
B,Y;MACpB,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;0EAUX,4B;MAE6D,iBAAY,KAAZ,C;K;+GC6B7D,sJ;M
AEsC,mB;QAAA,MA4GuD,M;;MA5GG,oB;QAAA,OAAgB,E;MAAI,oB;QAAA,OAAgB,E;MAAI,mB;QAAA,M
AAe,E;MAAI,qB;QAAA,QAAiB,S;MAAW,oB;QAAA,OAAgB,S;MAAW,qB;QAAA,QAAiB,S;MAAW,qB;QAA
A,QAAiB,S;MAAW,uB;QAAA,UAAmB,S;MAAW,yB;QAAA,YAAqB,S;MAAW,wB;QAAA,WAAqB,K;MAAO
,sB;QAAA,SAAmB,K;MAAO,wB;QAAA,WAAqB,K;MAAO,kC;QAAA,qBAA+B,K;MAAO,sB;QAAA,SAAmB,
K;MAAO,oB;QAAA,OAAa,I;MAAM,uB;QAAA,UAAsC,E;MAC/gB,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;M
ACX,EAAE,MAAF,IAAY,I;MACZ,EAAE,MAAF,IAAY,I;MACZ,EAAE,KAAF,IAAW,G;MACX,EAAE,OAAF,I
AAa,K;MACb,EAAE,MAAF,IAAY,I;MACZ,EAAE,OAAF,IAAa,K;MACb,EAAE,OAAF,IAAa,K;MACb,EAAE,
SAAF,IAAe,O;MACf,EAAE,WAAF,IAAiB,S;MACjB,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,QAAF,IAAc,M;M
ACd,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,oBAAF,IAA0B,kB;MAC1B,EAAE,QAAF,IAAc,M;MACd,EAAE,
MAAF,IAAY,I;MACZ,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;6GAWX,+B;MAEsE,oB;QAAA,OAAgB,S;MA
ClF,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAAE,MAAF,IAAY,I;MACZ,
OAAO,C;K;qHASX,e;MAEyC,mB;QAAA,MAAe,E;MACpD,QAAQ,E;MACR,EAAE,KAAF,IAAW,G;MACX,O
AAO,C;K;mHAyBX,+D;MAEqE,sB;QAAA,SAAkB,E;MAAI,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K

;MAAO,wB;QAAA,WAAqB,K;MACrK,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,QAAF,IAAc,M
;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;
K;iGAUwE,qB;MAAQ,OAAU,S;K;6FAEnB,qB;MAAQ,OAAS,Q;K;+FAEhB,qB;MAAQ,OAAU,S;K;2FASvB,qB
;MAAQ,OAAO,M;K;yFAEhB,qB;MAAQ,OAAM,K;K;yFAEd,qB;MAAQ,OAAM,K;K;yGCrJ3F,uB;MAEsC,qB;
QAAA,QAAiB,S;MAAW,oB;QAAA,ORy9MW,S;;MQx9MzE,QAAQ,E;MACR,EAAE,OAAF,IAAa,K;MACb,EA
AE,MAAF,IAAY,I;MACZ,OAAO,C;K;6HAuCX,mF;MAEgD,oB;QAAA,OAAa,S;MAAW,sB;QAAA,SAAkB,S;
MAAW,2B;QAAA,cAAuB,S;MAAW,sB;QAAA,SAA2C,S;MAAW,qB;QAAA,QAA6B,S;MAAW,uB;QAAA,UA
AoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/S,QAAQ,E;MACR,EAAE,MAAF,IAA
Y,I;MACZ,EAAE,QAAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,QAAF,IAAc,M;MACd,EAAE,
OAAF,IAAa,K;MACb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MA
ChB,OAAO,C;K;uGA2DX,qC;MAEqC,mC;QAAA,sBAAgC,K;MAAO,oB;QAAA,OA4UD,Q;;MA3UvE,QAAQ,
E;MACR,EAAE,qBAAF,IAA2B,mB;MAC3B,EAAE,MAAF,IAAY,I;MACZ,OAAO,C;K;yGAmBX,yC;MAEsC,u
B;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAChH,QAAQ,E;MACR,EA
AE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;yGAsB
X,2B;MAGI,QAAQ,E;MACR,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,OAAO,C;K;+FA8BX,s
E;MAEoD,wB;QAAA,WAAoB,I;MAAM,wB;QAAA,WAAqB,K;MAAO,uB;QAAA,UAAoB,K;MAAO,0B;QAA
A,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpL,QAAQ,E;MACR,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF
,IAAgB,Q;MAChB,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MACl
B,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;6GAuBX,0D;MAE2D,sB;QAAA,SAAkB,M;MAAQ,uB;QAAA,U
AAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC/J,QAAQ,E;MACR,EAAE,SAAF,IA
Ae,O;MACf,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,
UAAF,IAAgB,Q;MAChB,OAAO,C;K;2GAaX,qC;MAE4D,sB;QAAA,SAAkB,S;MAAW,uB;QAAA,UAA0B,S;M
AC/G,QAAQ,E;MACR,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,QAAF,IAAc,M;MACd,EAAE,SAAF,IAAe,O;M
ACf,OAAO,C;K;uHAuCX,mF;MAE6C,oB;QAAA,OAAa,S;MAAW,sB;QAAA,SAAkB,S;MAAW,2B;QAAA,cA
AuB,S;MAAW,sB;QAAA,SAAmD,S;MAAW,qB;QAAA,QAA6B,S;MAAW,uB;QAAA,UAAoB,K;MAAO,0B;Q
AAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MACpT,QAAQ,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,Q
AAF,IAAc,M;MACd,EAAE,aAAF,IAAmB,W;MACnB,EAAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MA
Cb,EAAE,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;q
GA+BX,6D;MAEoC,4B;QAAA,eAAyB,K;MAAO,4B;QAAA,eAAyB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,yB
;QAAA,YAAqB,S;MACnJ,QAAQ,E;MACR,EAAE,cAAF,IAAoB,Y;MACpB,EAAE,cAAF,IAAoB,Y;MACpB,E
AAE,YAAF,IAAkB,U;MAClB,EAAE,WAAF,IAAiB,S;MACjB,OAAO,C;K;yGAkBX,4C;MAEsC,oB;QAAA,OA
AgB,S;MAAW,uB;QAAA,UAAoB,S;MAAW,wB;QAAA,WAAsB,S;MAAW,uB;QAAA,UAA8B,S;MAC3J,QAA
Q,E;MACR,EAAE,MAAF,IAAY,I;MACZ,EAAE,SAAF,IAAe,O;MACf,EAAE,UAAF,IAAgB,Q;MAChB,EAAE,
SAAF,IAAe,O;MACf,OAAO,C;K;+FAkCmE,qB;MAAQ,OAAa,Y;K;6FAEtB,qB;MAAQ,OAAY,W;K;+FAEnB,q
B;MAAQ,OAAa,Y;K;6FAEtB,qB;MAAQ,OAAY,W;K;6FAEpB,qB;MAAQ,OAAY,W;K;6FAStC,qB;MAAQ,OA
AY,W;K;6FAEpB,qB;MAAQ,OAAY,W;K;uFAEvB,qB;MAAQ,OAAS,Q;K;qFAEnB,qB;MAAQ,OAAO,M;K;uF
ASX,qB;MAAQ,OAAS,Q;K;yFAEjB,qB;MAAQ,OAAS,Q;K;qGAEX,qB;MAAQ,OAAe,c;K;iFAEhC,qB;MAAQ,
OAAM,K;K;iGCharE,0E;MAEoC,gC;QAAA,mBAA6B,K;MAAO,sB;QAAA,SAAkB,C;MAAG,qB;QAAA,QAAi
B,C;MAAG,uB;QAAA,UAAoB,K;MAAO,0B;QAAA,aAAuB,K;MAAO,wB;QAAA,WAAqB,K;MAC3L,QAAQ,
E;MACR,EAAE,kBAAF,IAAwB,gB;MACxB,EAAE,QAAF,IAAc,M;MACd,EAAE,OAAF,IAAa,K;MACb,EAAE
,SAAF,IAAe,O;MACf,EAAE,YAAF,IAAkB,U;MAClB,EAAE,UAAF,IAAgB,Q;MAChB,OAAO,C;K;mFAU8E,q
B;MAAQ,OAAG,E;K;+FAEL,qB;MAAQ,OAAc,a;K;iFAE7B,qB;MAAQ,OAAO,M;K;yFAEX,qB;MAAQ,OAAW
,U;K;+EAEvB,qB;MAAQ,OAAO,M;K;+EAEf,qB;MAAQ,OAAO,M;K;oEnIjIvG,yB;MAAA,kF;MAAA,0B;MAA
A,uB;QAaI,IAAI,OAAO,CAAP,IAA8B,OAAO,KAAzC,C;UACI,MAAM,8BAAyB,wBAAqB,IAA9C,C;;QAEV,O
AAY,OAAL,IAAK,C;O;KAhBhB,C;0EAyCiC,qB;MAAQ,OAAA,SAAK,I;K;IoI7C9C,iC;K;;ICMA,4B;K;;IA6BA
,gD;K;;IC5BA,qC;K;;IAyBA,+B;K;;IC6DqC,uC;MACjC,uB;QAAA,UAAsB,E;MACtB,qB;QAAA,+C;MADA,sB;
MACA,kB;K;IAEA,4C;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,0C;MAAA,6C;O;MAKI,4E;MAGA,wE;K;;IAH
A,mD;MAAA,gC;MAAA,2C;K;;IAGA,iD;MAAA,gC;MAAA,yC;K;;IARJ,sC;MAAA,2F;K;;IAAA,2C;MAAA,a;a

AAA,S;UAAA,gD;aAAA,O;UAAA,8C;;UAAA,+D;;K;;;IA+ByB,4B;MACzB,8B;K;;IAGJ,qC;K;;IAyC6C,4C;MA
CzC,8B;K;;ICpKqC,sC;MACrC,8B;K;;ICD4C,8B;K;kDAI5C,mB;MAA6D,c;;QlJisD7C,Q;QADhB,IAAI,mCAAs
B,cAA1B,C;UAAqC,aAAO,K;UAAP,e;;QACrB,sB;QAAhB,OAAgB,cAAhB,C;UAAgB,2B;UAAM,IkJjsD6C,OlJi
sD/B,SkJjsD+B,UlJisD7C,C;YAAwB,aAAO,I;YAAP,e;;;QAC9C,aAAO,K;;;MkJlsDsD,iB;K;uDAE7D,oB;MACa,
c;;QlJyqDG,Q;QADhB,IAAI,ckJxqDA,QlJwqDA,iBkJxqDA,QlJwqDsB,UAA1B,C;UAAqC,aAAO,I;UAAP,e;;QA
CrB,OkJzqDZ,QlJyqDY,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CkJzqDP,oBlJyqDkB,OkJzqDlB
,ClJyqDG,C;YAAyB,aAAO,K;YAAP,e;;;QAC/C,aAAO,I;;;MkJ1qDH,iB;K;2CAEJ,Y;MAAkC,qBAAQ,C;K;IAEq
B,qE;MAAA,qB;QAC3D,OAAI,OAAO,uBAAX,GAAiB,mBAAjB,GAA6C,SAAH,EAAG,C;O;K;4CADjD,Y;MA
AkC,4BAAa,IAAb,EAAmB,GAAnB,EAAwB,GAAxB,kBAA6B,wCAA7B,C;K;2CAIlC,Y;MAI4C,uBAAgB,IAA
hB,C;K;mDAE5C,iB;MAI4D,yBAAgB,IAAhB,EAAsB,KAAtB,C;K;;IC/BhE,8B;MAAA,e;MAAA,iB;MAAA,uB;
K;IAAA,4B;MAAA,+B;O;MACI,4C;MACA,kD;MACA,0C;MACA,8C;K;;IAHA,mC;MAAA,kB;MAAA,2B;K;;I
ACA,sC;MAAA,kB;MAAA,8B;K;;IACA,kC;MAAA,kB;MAAA,0B;K;;IACA,oC;MAAA,kB;MAAA,4B;K;;IAJJ,
wB;MAAA,sH;K;;IAAA,6B;MAAA,a;aAAA,O;UAAA,gC;aAAA,U;UAAA,mC;aAAA,M;UAAA,+B;aAAA,Q;U
AAA,iC;;UAAA,6D;;K;;IAOA,4B;MAKI,mD;MACA,2BAA4B,I;K;yCAE5B,Y;MAEiB,IAAN,I;MzJUX,IAAI,Ey
JXQ,mDzJWR,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAyB,OAAQ,WAAjC,C;;MyJZC,QAAM,oBAAN,M;a
ACH,M;UAAc,Y;UAAd,K;aACA,O;UAAe,W;UAAf,K;;UACQ,wC;UAHL,K;;MAAP,W;K;sCAOJ,Y;MAIW,Q;
MAHP,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MACtB,mD;MAEA,OAAO,2F;K;4DAGX,Y;MACI,iD;MACA,
kB;MACA,OAAO,kD;K;+CAeX,iB;MAII,2BAAY,K;MACZ,gD;K;sCAGJ,Y;MAII,+C;K;;ICjDkC,wB;MAoFtC,o
C;MApFgE,6B;K;sCAIhE,Y;MAAuC,0C;K;2CAEvC,mB;MAAwD,uB;;QpJoU3C,Q;QADb,YAAY,C;QACC,sB;
QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IoJrUmE,OpJqUrD,IoJrUqD,UpJqUnE,C;YACI,sBAAO,K;YAAP,wB;;U
ACJ,qB;;QAEJ,sBAAO,E;;;MoJzUiD,0B;K;+CAExD,mB;MAA4D,sB;;QpJ6V5D,eAAoB,0BAAa,SAAb,C;QACp
B,OAAO,QAAS,cAAhB,C;UACI,IoJ/VsE,OpJ+VxD,QAAS,WoJ/V+C,UpJ+VtE,C;YACI,qBAAO,QAAS,Y;YAA
hB,uB;;;QAGR,qBAAO,E;;;MoJnWqD,yB;K;0CAE5D,Y;MAA+C,+CAAiB,CAAjB,C;K;kDAE/C,iB;MAAyD,+C
AAiB,KAAjB,C;K;6CAEzD,8B;MAA8D,gCAAQ,IAAR,EAAc,SAAd,EAAyB,OAAzB,C;K;IAElC,wD;MAAgF,u
B;MAA/E,kB;MAAmC,4B;MAC5D,eAAyB,C;MAGrB,+DAAkB,gBAAlB,EAA6B,OAA7B,EAAsC,WAAK,KA
A3C,C;MACA,eAAa,UAAU,gBAAV,I;K;iDAGjB,iB;MACI,+DAAkB,KAAlB,EAAyB,YAAzB,C;MAEA,OAAO,
wBAAK,mBAAY,KAAZ,IAAL,C;K;4FAGY,Y;MAAQ,mB;K;;oCAGnC,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB
,OAAO,I;MAC3B,IAAI,2BAAJ,C;QAAuB,OAAO,K;MAE9B,OAAO,2DAAc,IAAd,EAAoB,KAApB,C;K;sCAG
X,Y;MAG+B,oEAAgB,IAAhB,C;K;IAE/B,2C;MAAAA,oB;MACI,eACsB,C;K;kDAEtB,Y;MAAkC,sBAAQ,gB;K;
+CAE1C,Y;MAEe,gB;MADX,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;MACX,iE;MAAX,OAAO,+B;K;;IAO0
B,sD;MAHzC,oB;MAGwD,iD;MAGhD,gEAAmB,KAAnB,EAA0B,WAAkB,KAA5C,C;MACA,eAAa,K;K;0DAG
jB,Y;MAAsC,sBAAQ,C;K;wDAE9C,Y;MAAgC,mB;K;uDAEhC,Y;MACI,IAAI,CAAC,kBAAL,C;QAAoB,MAA
M,6B;MAC1B,OAAO,yBAAI,mCAAJ,EAAI,YAAJ,E;K;4DAGX,Y;MAAoC,sBAAQ,CAAR,I;K;;IAGxC,kC;MA
AA,sC;K;iEACI,uB;MACI,IAAI,QAAQ,CAAR,IAAa,SAAS,IAA1B,C;QACI,MAAM,8BAA0B,YAAS,KAAT,gB
AAuB,IAAjD,C;;K;kEAId,uB;MACI,IAAI,QAAQ,CAAR,IAAa,QAAQ,IAAzB,C;QACI,MAAM,8BAA0B,YAAS,
KAAT,gBAAuB,IAAjD,C;;K;iEAId,oC;MACI,IAAI,YAAY,CAAZ,IAAAiB,UAAU,IAA/B,C;QACI,MAAM,8BAA
0B,gBAAa,SAAb,mBAAkC,OAAlC,gBAAkD,IAA5E,C;;MAEV,IAAI,YAAY,OAAhB,C;QACI,MAAM,gCAAyB
,gBAAa,SAAb,oBAAmC,OAA5D,C;;K;kEAId,sC;MACI,IAAI,aAAa,CAAb,IAAAkB,WAAW,IAAAjC,C;QACI,MA
AM,8BAA0B,iBAAc,UAAd,oBAAqC,QAArC,gBAAsD,IAAhF,C;;MAEV,IAAI,aAAa,QAAAjB,C;QACI,MAAM,g
CAAyB,iBAAc,UAAd,qBAAsC,QAA/D,C;;K;+DAId,a;MAEc,UACsB,M;MAFhC,iBAAe,C;MACL,mB;MAAV,
OAAU,cAAV,C;QAAU,mB;QACN,aAAW,MAAK,UAAL,SAAiB,6DAAiB,CAAlC,K;;MAEf,OAAO,U;K;6DAG
X,oB;MAIiB,Q;MAHb,IAAI,CAAE,KAAF,KAAU,KAAM,KAApB,C;QAA0B,OAAO,K;MAEjC,oBAAoB,KAA
M,W;MACb,mB;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,gBAAgB,aAAc,O;QAC9B,IAAI,cAAQ,SAAR,CAAJ,C;
UACI,OAAO,K;;;MAGf,OAAO,I;K;;;IAjDf,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;;ICnFwC,uB;MAyHxC,mC;
MAzCA,uBAC6B,I;MAmC7B,yBACsC,I;K;8CAnHtC,e;MACI,OAAO,6BAAc,GAAd,S;K;gDAGX,iB;MAAwE,g
BAAR,Y;MAAQ,c;;QrJwrDxD,Q;QADhB,IAAI,wCAAsB,mBAA1B,C;UAAqC,aAAO,K;UAAP,e;;QACrB,2B;Q
AAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IqJxrDwD,OrJwrD1C,OqJxrD6C,MAAH,QrJwrDxD,C;YAAwB,aA
AO,I;YAAP,e;;;QAC9C,aAAO,K;;;MqJzrDyD,iB;K;kDAEhE,iB;MAEI,IAAI,gCAAJ,C;QAA+B,OAAO,K;MACt

C,UAAU,KAAM,I;MAChB,YAAY,KAAM,M;MpKmNO,Q;MoKlNzB,epKkN4C,CAAnB,mDAAmB,YoKlNzB,
GpKkNyB,C;MoKhN5C,IAAI,eAAS,QAAT,CAAJ,C;QACI,OAAO,K;;MAIP,6B;MAAA,W;QpK4NqB,U;QoK5N
D,UpK4NoB,CAAnB,uDAAmB,oBoK5NP,GpK4NO,C;;MoK5N5C,W;QACI,OAAO,K;;MAGX,OAAO,I;K;mCA
IX,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,0BAAJ,C;QAAyB,OAAO,K;MAChC,IAAI,c
AAQ,KAAM,KAAlB,C;QAAwB,OAAO,K;MAEV,gBAAd,KAAM,Q;MAAQ,c;;QrJmoDT,Q;QADhB,IAAI,wCA
AsB,mBAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,CqJ
noDK,2BrJmoDM,OqJnoDN,CrJmoDT,C;YAAyB,aAAO,K;YAAP,e;;;QAC/C,aAAO,I;;;MqJpoDH,iB;K;sCAGJ,e
;MAAwC,Q;MAAA,4CAAc,GAAd,8B;K;qCAGxC,Y;MAK+B,OAAQ,SAAR,YAAQ,C;K;oCAEvC,Y;MAAkC,q
BAAQ,C;K;mFACnB,Y;MAAQ,OAAA,YAAQ,K;K;IAWnB,0E;MAAA,wC;MAAS,sB;K;8EACb,mB;MAAsD,+
CAAY,OAAZ,C;K;IAI3C,sG;MAAA,kD;K;8FACH,Y;MAAkC,OAAA,0BAAc,U;K;2FAChD,Y;MAAyB,OAAA,
0BAAc,OAAO,I;K;;wEAJtD,Y;MACI,oBAAoB,6BAAQ,W;MAC5B,+F;K;sHAMmB,Y;MAAQ,OAAA,qBAAiB,
K;K;;mFAb5D,Y;MACI,IAAI,4BAAJ,C;QACI,+E;;MAcJ,OAAO,mC;K;IAOwD,uD;MAAA,qB;QAAE,2CAAS,E
AAT,C;O;K;qCAAzE,Y;MAAkC,OAAQ,eAAR,YAAQ,EAAa,IAAb,EAAmB,GAAnB,EAAwB,GAAxB,kBAA6B
,iCAA7B,C;K;+CAE1C,iB;MAAuD,+BAAS,KAAM,IAAf,IAAsB,GAAtB,GAA4B,wBAAS,KAAM,MAAf,C;K;+
CAEnF,a;MAAwC,OAAI,MAAM,IAAV,GAAgB,YAAhB,GAAoC,SAAF,CAAE,C;K;IAWtD,4E;MAAA,wC;MA
AS,6B;K;gFACf,mB;MAAsE,iDAAc,OAAd,C;K;IAI3D,wG;MAAA,kD;K;gGACH,Y;MAAkC,OAAA,0BAAc,U;
K;6FAChD,Y;MAAyB,OAAA,0BAAc,OAAO,M;K;;0EAJtD,Y;MACI,oBAAoB,6BAAQ,W;MAC5B,iG;K;wHA
MmB,Y;MAAQ,OAAA,qBAAiB,K;K;;qFAb5D,Y;MACI,IAAI,8BAAJ,C;QACI,mF;;MAcJ,OAAO,qC;K;oDAMf,
e;MAA8D,gBAAR,Y;MAAQ,sB;;QrJmJ9C,Q;QAAA,2B;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IqJnJsD
,OrJmJxC,OqJnJ2C,IAAH,MrJmJtD,C;YAAwB,qBAAO,O;YAAP,uB;;;QAC9C,qBAAO,I;;;MqJpJ+C,yB;K;IAEt
D,iC;MAAA,qC;K;4DAEI,a;MAAiE,gC;MAAX,OAAU,CAAC,kBAAN,CAAM,0DAAmB,CAApB,KAA4B,oBA
AjC,CAAiC,8DAAqB,CAAjD,C;K;4DAChE,a;MAAyD,OAAU,SAAL,CAAO,IAAF,mBAAL,CAAY,MAAP,C;K;
0DACnE,oB;MACI,IAAI,gCAAJ,C;QAA+B,OAAO,K;MACtC,OAAO,OAAA,CAAE,IAAF,EAAS,KAAM,IAAf,
KAAsB,OAAA,CAAE,MAAF,EAAW,KAAM,MAAjB,C;K;;;IANrC,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;;IC
hIqC,uB;MAkBrC,mC;MAlB+D,6B;K;mCAE/D,iB;MAMI,IAAI,UAAU,IAAd,C;QAAoB,OAAO,I;MAC3B,IAAI,
0BAAJ,C;QAAsB,OAAO,K;MAC7B,OAAO,sDAAU,IAAV,EAAgB,KAAhB,C;K;qCAGX,Y;MAG+B,qEAAkB,I
AAlB,C;K;IAE/B,iC;MAAA,qC;K;gEACI,a;MAEoB,Q;MADhB,iBAAe,C;MACC,mB;MAAhB,OAAgB,cAAhB,
C;QAAgB,yB;QACC,U;QAAb,2BAAa,yEAAuB,CAApC,K;;MAEJ,OAAO,U;K;wDAGX,oB;MACI,IAAI,CAAE,
KAAF,KAAU,KAAM,KAApB,C;QAA0B,OAAO,K;MACjC,OAAO,CtK8OsG,qBsK9OxF,KtK8OwF,C;K;;;IsKzP
rH,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;;;MCghBA,kC;MA9hBA,cAAwB,C;MACxB,yB;MAEA,sBAAyB,C;;
kFAAzB,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;4CA8BA,uB;MAOI,IAAI,cAAc,CAAlB,C;QAAqB,MAAM,6B
AAsB,mBAAtB,C;MAC3B,IAAI,eAAe,kBAAY,OAA/B,C;QAAqC,M;MACrC,IAAI,uBAAgB,qDAApB,C;QACI,
qBAAc,gBAAyB,gBAAZ,WAAY,EAAc,EAAd,CAAzB,O;QACd,M;;MAGJ,kBAAkB,uDAAY,kBAAY,OAAxB,
EAA8B,WAA9B,C;MAClB,oBAAa,WAAb,C;K;0CAGJ,uB;MAII,kBAAkB,gBAAmB,WAAnB,O;M5J20BtB,U4J
10BI,kB5J00BJ,E4J10ByB,W5J00BzB,E4J10BsC,C5J00BtC,E4J10ByC,W5J00BzC,E4J10B+C,kBAAY,O5J00B3
D,C;MAAA,U4Jz0BI,kB5Jy0BJ,E4Jz0ByB,W5Jy0BzB,E4Jz0BsC,kBAAY,OAAZ,GAAmB,WAAnB,I5Jy0BtC,E4
Jz0B+D,C5Jy0B/D,E4Jz0BkE,W5Jy0BlE,C;M4Jx0BI,cAAO,C;MACP,qBAAc,W;K;yCAGlB,yB;MAGW,Q;MAA
P,OAAO,2BAAY,aAAZ,4D;K;yCAGX,iB;MAA2C,OAAI,SAAS,kBAAY,OAAzB,GAA+B,QAAQ,kBAAY,OAA
pB,IAA/B,GAA6D,K;K;yCAExG,iB;MAA2C,OAAI,QAAQ,CAAZ,GAAe,QAAQ,kBAAY,OAApB,IAAf,GAA6C
,K;K;2CAExF,iB;MACoD,0BAAY,cAAO,KAAP,IAAZ,C;K;yCAEpD,iB;MAA2C,OAAI,UAAqB,cAAZ,kBAAY,
CAAzB,GAAoC,CAApC,GAA2C,QAAQ,CAAR,I;K;yCAEtF,iB;MAA2C,OAAI,UAAS,CAAb,GAA4B,cAAZ,kB
AAY,CAA5B,GAA2C,QAAQ,CAAR,I;K;mCAEtF,Y;MAAkC,qBAAQ,C;K;iCAE1C,Y;MAGwB,IAAI,cAAJ,C;Q
AAe,MAAM,2BAAuB,sBAAvB,C;;QAnBlC,Q;QAmBa,OAnBb,2BAmBkG,WAnBlG,4D;;K;uCAqBX,Y;MAG+B
,Q;MAAA,IAAI,cAAJ,C;QAAA,OAAAe,I;;QAxBnC,U;QAwBoB,OAxBpB,6BAwByD,WAxBzD,gE;;MAwBoB,W
;K;gCAE/B,Y;MAGuB,IAAI,cAAJ,C;QAAe,MAAM,2BAAuB,sBAAvB,C;;QA7BjC,Q;QA6BY,OA7BZ,2BAQyC
,mBAAY,cAqB0D,sBArB1D,IAAZ,CARzC,4D;;K;sCA+BX,Y;MAG8B,Q;MAAA,IAAI,cAAJ,C;QAAA,OAAAe,I;;
QAlClC,U;QAkCmB,OAlCnB,6BAQyC,mBAAY,cA0BiB,sBA1BjB,IAAZ,CARzC,gE;;MAkCmB,W;K;0CAE9B,
mB;MAII,sBAAe,YAAO,CAAP,IAAf,C;MAEA,cAAO,mBAAY,WAAZ,C;MACP,mBAAY,WAAZ,IAAAoB,O;M

ACpB,wBAAQ,CAAR,I;K;yCAGJ,mB;MAII,sBAAe,YAAO,CAAP,IAAf,C;MAEA,mBA7CgD,mBAAY,cA6ClC,
SA7CkC,IAAZ,CA6ChD,IAAmC,O;MACnC,wBAAQ,CAAR,I;K;uCAGJ,Y;MAII,IAAI,cAAJ,C;QAAe,MAAM,2
BAAuB,sBAAvB,C;MA7Dd,Q;MA+DP,cA/DO,2BA+DmB,WA/DnB,4D;MAgEP,mBAAY,WAAZ,IAAoB,I;MA
CpB,cAAO,mBAAY,WAAZ,C;MACP,wBAAQ,CAAR,I;MACA,OAAO,O;K;6CAGX,Y;MAGqC,OAAI,cAAJ,G
AAe,IAAf,GAAyB,kB;K;sCAE9D,Y;MAII,IAAI,cAAJ,C;QAAe,MAAM,2BAAuB,sBAAvB,C;MAErB,wBAzEg
D,mBAAY,cAyEtB,sBAzEsB,IAAZ,C;MARzC,Q;MAkFP,cAlFO,2BAkFmB,iBAlFnB,4D;MAmFP,mBAAY,iBA
AZ,IAAiC,I;MACjC,wBAAQ,CAAR,I;MACA,OAAO,O;K;4CAGX,Y;MAGoC,OAAI,cAAJ,GAAe,IAAf,GAAyB
,iB;K;qCAE7D,mB;MAEI,mBAAQ,OAAR,C;MACA,OAAO,I;K;uCAGX,0B;MACI,oCAAa,4BAAmB,KAAnB,E
AA0B,SAA1B,C;MAEb,IAAI,UAAS,SAAb,C;QACI,mBAAQ,OAAR,C;QACA,M;aACG,IAAI,UAAS,CAAb,C;Q
ACH,oBAAS,OAAT,C;QACA,M;;MAGJ,sBAAe,YAAO,CAAP,IAAf,C;MA2BA,oBAjIgD,mBAAY,cAiI1B,KAjI
0B,IAAZ,C;MAmIhD,IAAI,QAAS,SAAD,GAAQ,CAAR,IAAe,CAA3B,C;QAEI,+BAA+B,mBAAY,aAAZ,C;QA
C/B,sBAAsB,mBAAY,WAAZ,C;QAEtB,IAAI,4BAA4B,WAAhC,C;UACI,mBAAY,eAAZ,IAA+B,mBAAY,WA
AZ,C;U5JgrB3C,U4J/qBY,kB5J+qBZ,E4J/qBiC,kB5J+qBjC,E4J/qB8C,W5J+qB9C,E4J/qBoD,cAAO,CAAP,I5J+q
BpD,E4J/qB8D,2BAA2B,CAA3B,I5J+qB9D,C;;UAAA,U4J7qBY,kB5J6qBZ,E4J7qBiC,kB5J6qBjC,E4J7qB8C,cA
AO,CAAP,I5J6qB9C,E4J7qBwD,W5J6qBxD,E4J7qB8D,kBAAY,O5J6qB1E,C;U4J5qBY,mBAAY,kBAAY,OAA
Z,GAAmB,CAAnB,IAAZ,IAAoC,mBAAY,CAAZ,C;U5J4qBhD,U4J3qBY,kB5J2qBZ,E4J3qBiC,kB5J2qBjC,E4J3
qB8C,C5J2qB9C,E4J3qBiD,C5J2qBjD,E4J3qBoD,2BAA2B,CAA3B,I5J2qBpD,C;;Q4JxqBQ,mBAAY,wBAAZ,IA
AwC,O;QACxC,cAAO,e;;QAGP,WArJ4C,mBAAY,cAqJ/B,SArJ+B,IAAZ,C;QAuJ5C,IAAI,gBAAgB,IAApB,C;U
5JkqBR,U4JjqBY,kB5JiqBZ,E4JjqBiC,kB5JiqBjC,E4JjqB8C,gBAAgB,CAAhB,I5JiqB9C,E4JjqBiE,a5JiqBjE,E4Jj
qBgF,I5JiqBhF,C;;UAAA,U4J/pBY,kB5J+pBZ,E4J/pBiC,kB5J+pBjC,E4J/pB8C,C5J+pB9C,E4J/pBiD,C5J+pBjD,
E4J/pBoD,I5J+pBpD,C;U4J9pBY,mBAAY,CAAZ,IAAiB,mBAAY,kBAAY,OAAZ,GAAmB,CAAnB,IAAZ,C;U5
J8pB7B,U4J7pBY,kB5J6pBZ,E4J7pBiC,kB5J6pBjC,E4J7pB8C,gBAAgB,CAAhB,I5J6pB9C,E4J7pBiE,a5J6pBjE,
E4J7pBgF,kBAAY,OAAZ,GAAmB,CAAnB,I5J6pBhF,C;;Q4J1pBQ,mBAAY,aAAZ,IAA6B,O;;MAEjC,wBAAQ,
CAAR,I;K;oDAGJ,mC;MAGkD,UAIxB,M;MANtB,eAAe,QAAS,W;MAEsB,OAAZ,kBAAY,O;MAA9C,iBAAc,a
AAd,wB;QACI,IAAI,CAAC,QAAS,UAAd,C;UAAyB,K;QACzB,mBAAY,KAAZ,IAAqB,QAAS,O;;MAEZ,oB;M
AAtB,mBAAc,CAAd,8B;QACI,IAAI,CAAC,QAAS,UAAd,C;UAAyB,K;QACzB,mBAAY,OAAZ,IAAqB,QAAS,
O;;MAGlC,wBAAQ,QAAS,KAAjB,I;K;0CAGJ,oB;MACI,IAAI,QAAS,UAAb,C;QAAwB,OAAO,K;MAC/B,sBA
Ae,IAAK,KAAL,GAAY,QAAS,KAArB,IAAf,C;MACA,8BAtLgD,mBAAY,cAsLvB,SAtLuB,IAAZ,CAsLhD,EA
A4C,QAA5C,C;MACA,OAAO,I;K;0CAGX,2B;MACI,oCAAa,4BAAmB,KAAnB,EAA0B,SAA1B,C;MAEb,IAAI
,QAAS,UAAb,C;QACI,OAAO,K;aACJ,IAAI,UAAS,SAAb,C;QACH,OAAO,oBAAO,QAAP,C;;MAGX,sBAAe,I
AAK,KAAL,GAAY,QAAS,KAArB,IAAf,C;MAEA,WArMgD,mBAAY,cAqMnC,SArMmC,IAAZ,C;MAsMhD,o
BAtMgD,mBAAY,cAsM1B,KAtM0B,IAAZ,C;MAuMhD,mBAAmB,QAAS,K;MAE5B,IAAI,QAAS,SAAD,GAA
Q,CAAR,IAAe,CAA3B,C;QAGI,kBAAkB,cAAO,YAAP,I;QAElB,IAAI,iBAAiB,WAArB,C;UACI,IAAI,eAAe,C
AAnB,C;Y5J0mBZ,U4JzmBgB,kB5JymBhB,E4JzmBqC,kB5JymBrC,E4JzmBkD,W5JymBlD,E4JzmB+D,W5Jym
B/D,E4JzmBqE,a5JymBrE,C;;Y4JvmBgB,4BAAe,kBAAY,OAA3B,I;YACA,sBAAsB,gBAAgB,WAAhB,I;YACt
B,kBAAkB,kBAAY,OAAZ,GAAmB,WAAnB,I;YAElB,IAAI,eAAe,eAAnB,C;c5JmmBhB,U4JlmBoB,kB5JkmBp
B,E4JlmByC,kB5JkmBzC,E4JlmBsD,W5JkmBtD,E4JlmBmE,W5JkmBnE,E4JlmByE,a5JkmBzE,C;;cAAA,U4Jhm
BoB,kB5JgmBpB,E4JhmByC,kB5JgmBzC,E4JhmBsD,W5JgmBtD,E4JhmBmE,W5JgmBnE,E4JhmByE,cAAO,W
AAP,I5JgmBzE,C;cAAA,U4J/lBoB,kB5J+lBpB,E4J/lByC,kB5J+lBzC,E4J/lBsD,C5J+lBtD,E4J/lByD,cAAO,WAA
P,I5J+lBzD,E4J/lB6E,a5J+lB7E,C;;;;;UAAA,U4J3lBY,kB5J2lBZ,E4J3lBiC,kB5J2lBjC,E4J3lB8C,W5J2lB9C,E4J3l
B2D,W5J2lB3D,E4J3lBiE,kBAAY,O5J2lB7E,C;U4J1lBY,IAAI,gBAAgB,aAApB,C;Y5J0lBZ,U4JzlBgB,kB5JylB
hB,E4JzlBqC,kB5JylBrC,E4JzlBkD,kBAAY,OAAZ,GAAmB,YAAnB,I5JylBlD,E4JzlBmF,C5JylBnF,E4JzlBsF,a5
JylBtF,C;;YAAA,U4JvlBgB,kB5JulBhB,E4JvlBqC,kB5JulBrC,E4JvlBkD,kBAAY,OAAZ,GAAmB,YAAnB,I5Jul
BlD,E4JvlBmF,C5JulBnF,E4JvlBsF,Y5JulBtF,C;YAAA,U4JtlBgB,kB5JslBhB,E4JtlBqC,kB5JslBrC,E4JtlBkD,C5J
slBlD,E4JtlBqD,Y5JslBrD,E4JtlBmE,a5JslBnE,C;;;Q4JnlBQ,cAAO,W;QACP,8BAAuB,mBAAY,gBAAgB,YAAh
B,IAAZ,CAAvB,EAAkE,QAAlE,C;;QAIA,2BAA2B,gBAAgB,YAAhB,I;QAE3B,IAAI,gBAAgB,IAApB,C;UACI,
IAAI,QAAO,YAAP,SAAuB,kBAAY,OAAvC,C;Y5J2kBZ,U4J1kBgB,kB5J0kBhB,E4J1kBqC,kB5J0kBrC,E4J1kB
kD,oB5J0kBlD,E4J1kBwE,a5J0kBxE,E4J1kBuF,I5J0kBvF,C;;Y4JxkBgB,IAAI,wBAAwB,kBAAY,OAAAxC,C;c5J

wkBhB,U4JvkBoB,kB5JukBpB,E4JvkByC,kB5JukBzC,E4JvkBsD,uBAAuB,kBAAY,OAAnC,I5JukBtD,E4JvkB+
F,a5JukB/F,E4JvkB8G,I5JukB9G,C;;c4JrkBoB,mBAAmB,OAAO,YAAP,GAAsB,kBAAY,OAAlC,I;c5JqkBvC,U4
JpkBoB,kB5JokBpB,E4JpkByC,kB5JokBzC,E4JpkBsD,C5JokBtD,E4JpkByD,OAAO,YAAP,I5JokBzD,E4JpkB8E
,I5JokB9E,C;cAAA,U4JnkBoB,kB5JmkBpB,E4JnkByC,kB5JmkBzC,E4JnkBsD,oB5JmkBtD,E4JnkB4E,a5JmkB5
E,E4JnkB2F,OAAO,YAAP,I5JmkB3F,C;;;;UAAA,U4J/jBY,kB5J+jBZ,E4J/jBiC,kB5J+jBjC,E4J/jB8C,Y5J+jB9C,
E4J/jB4D,C5J+jB5D,E4J/jB+D,I5J+jB/D,C;U4J9jBY,IAAI,wBAAwB,kBAAY,OAAxC,C;Y5J8jBZ,U4J7jBgB,kB
5J6jBhB,E4J7jBqC,kB5J6jBrC,E4J7jBkD,uBAAuB,kBAAY,OAAnC,I5J6jBlD,E4J7jB2F,a5J6jB3F,E4J7jB0G,kB
AAY,O5J6jBtH,C;;YAAA,U4J3jBgB,kB5J2jBhB,E4J3jBqC,kB5J2jBrC,E4J3jBkD,C5J2jBlD,E4J3jBqD,kBAAY,
OAAZ,GAAmB,YAAnB,I5J2jBrD,E4J3jBsF,kBAAY,O5J2jBlG,C;YAAA,U4J1jBgB,kB5J0jBhB,E4J1jBqC,kB5J
0jBrC,E4J1jBkD,oB5J0jBlD,E4J1jBwE,a5J0jBxE,E4J1jBuF,kBAAY,OAAZ,GAAmB,YAAnB,I5J0jBvF,C;;;Q4Jvj
BQ,8BAAuB,aAAvB,EAAsC,QAAtC,C;;MAGJ,OAAO,I;K;uCAGX,iB;MACI,oCAAa,2BAAkB,KAAlB,EAAyB,
SAAzB,C;MAjRN,Q;MAmRP,OAnRO,2BAQyC,mBAAY,cA2Q3B,KA3Q2B,IAAZ,CARzC,4D;K;uCAsRX,0B;
MACI,oCAAa,2BAAkB,KAAlB,EAAyB,SAAzB,C;MAEb,oBAjRgD,mBAAY,cAiR1B,KAjR0B,IAAZ,C;MARzC
,Q;MA0RP,iBA1RO,2BA0RsB,aA1RtB,4D;MA2RP,mBAAY,aAAZ,IAA6B,O;MAE7B,OAAO,U;K;0CAGX,mB;
MAAoD,0BAAQ,OAAR,MAAoB,E;K;yCAExE,mB;MAIsB,IAIA,IAJA,EAIuB,M;MAPzC,WA3RgD,mBAAY,c
A2RnC,SA3RmC,IAAZ,C;MA6RhD,IAAI,cAAO,IAAX,C;QACI,iBAAc,WAAd,UAAyB,IAAzB,U;UACI,IAAI,g
BAAW,mBAAY,KAAZ,CAAX,CAAJ,C;YAAmC,OAAO,QAAQ,WAAR,I;;aAE3C,IAAI,eAAQ,IAAZ,C;QACW,
kB;QAAuB,SAAZ,kBAAY,O;QAArC,qD;UACI,IAAI,gBAAW,mBAAY,OAAZ,CAAX,CAAJ,C;YAAmC,OAAO
,UAAQ,WAAR,I;;QAE9C,mBAAc,CAAd,YAAsB,IAAtB,Y;UACI,IAAI,gBAAW,mBAAY,OAAZ,CAAX,CAAJ,
C;YAAmC,OAAO,UAAQ,kBAAY,OAApB,GAA2B,WAA3B,I;;;MAIlD,OAAO,E;K;6CAGX,mB;MAIsC,UAOJ,
MAPI,EAOa,M;MAV/C,WA9SgD,mBAAY,cA8SnC,SA9SmC,IAAZ,C;MAgThD,IAAI,cAAO,IAAX,C;QACkC,k
B;QAA9B,iBAAc,OAAO,CAAP,IAAd,yB;UACI,IAAI,gBAAW,mBAAY,KAAZ,CAAX,CAAJ,C;YAAmC,OAA
O,QAAQ,WAAR,I;;aAE3C,IAAI,cAAO,IAAX,C;QACH,mBAAc,OAAO,CAAP,IAAd,aAA8B,CAA9B,Y;UACI,I
AAI,gBAAW,mBAAY,OAAZ,CAAX,CAAJ,C;YAAmC,OAAO,UAAQ,kBAAY,OAApB,GAA2B,WAA3B,I;;QA
EpB,uBAAZ,kBAAY,C;QAAiB,oB;QAA3C,wD;UACI,IAAI,gBAAW,mBAAY,OAAZ,CAAX,CAAJ,C;YAAmC,
OAAO,UAAQ,WAAR,I;;;MAIlD,OAAO,E;K;wCAGX,mB;MACI,YAAY,mBAAQ,OAAR,C;MACZ,IAAI,UAAS
,EAAb,C;QAAiB,OAAO,K;MACxB,sBAAS,KAAT,C;MACA,OAAO,I;K;4CAGX,iB;MACI,oCAAa,2BAAkB,K
AAlB,EAAyB,SAAzB,C;MAEb,IAAI,UAAS,sBAAb,C;QACI,OAAO,iB;aACJ,IAAI,UAAS,CAAb,C;QACH,OA
AO,kB;;MAGX,oBAhVgD,mBAAY,cAgV1B,KAhV0B,IAAZ,C;MARzC,Q;MAyVP,cAzVO,2BAyVmB,aAzVnB,
4D;MA2VP,IAAI,QAAQ,aAAS,CAArB,C;QAEI,IAAI,iBAAAiB,WAArB,C;U5JoeR,U4JneY,kB5JmeZ,E4JneiC,kB
5JmejC,E4Jne8C,cAAO,CAAP,I5Jme9C,E4JnewD,W5JmexD,E4Jne8D,a5Jme9D,C;;UAAA,U4JjeY,kB5JieZ,E4Jj
eiC,kB5JiejC,E4Jje8C,C5Jie9C,E4JjeiD,C5JiejD,E4JjeoD,a5JiepD,C;U4JheY,mBAAY,CAAZ,IAAiB,mBAAY,kB
AAY,OAAZ,GAAmB,CAAnB,IAAZ,C;U5Jge7B,U4J/dY,kB5J+dZ,E4J/diC,kB5J+djC,E4J/d8C,cAAO,CAAP,I5J+
d9C,E4J/dwD,W5J+dxD,E4J/d8D,kBAAY,OAAZ,GAAmB,CAAnB,I5J+d9D,C;;Q4J5dQ,mBAAY,WAAZ,IAAoB
,I;QACpB,cAAO,mBAAY,WAAZ,C;;QAGP,wBAjW4C,mBAAY,cAiWlB,sBAjWkB,IAAZ,C;QAmW5C,IAAI,iB
AAiB,iBAArB,C;U5JsdR,U4JrdY,kB5JqdZ,E4JrdiC,kB5JqdjC,E4Jrd8C,a5Jqd9C,E4Jrd6D,gBAAgB,CAAhB,I5Jq
d7D,E4JrdgF,oBAAoB,CAApB,I5JqdhF,C;;UAAA,U4JndY,kB5JmdZ,E4JndiC,kB5JmdjC,E4Jnd8C,a5Jmd9C,E4J
nd6D,gBAAgB,CAAhB,I5Jmd7D,E4JndgF,kBAAY,O5Jmd5F,C;U4JldY,mBAAY,kBAAY,OAAZ,GAAmB,CAA
nB,IAAZ,IAAoC,mBAAY,CAAZ,C;U5JkdhD,U4JjdY,kB5JidZ,E4JjdiC,kB5JidjC,E4Jjd8C,C5Jid9C,E4JjdiD,C5Ji
djD,E4JjdoD,oBAAoB,CAApB,I5JidpD,C;;Q4J9cQ,mBAAY,iBAAZ,IAAiC,I;;MAErC,wBAAQ,CAAR,I;MAEA,
OAAO,O;K;6CAGX,oB;MAAkE,0B;;QAa5C,wD;QART,aAAL,IAAK,U;QAAL,Y;UAA8B,SAAZ,kB5KoxOnB,
YAAQ,C;;Q4KpxOX,W;UACI,yBAAO,K;UAAP,2B;;QAEJ,WA1XgD,mBAAY,cA0XnC,SA1XmC,IAAZ,C;QA2
XhD,cAAc,W;QACd,eAAe,K;QAEf,IAAI,cAAO,IAAX,C;UACI,iBAAc,WAAd,UAAyB,IAAzB,U;YACI,cAAc,m
BAAY,KAAZ,C;YAGd,IAjBsE,CAAU,wBAiBlE,0EAjBkE,CAiBhF,C;cACI,mBAAY,gBAAZ,EAAY,wBAAZ,Y
AAyB,O;;cAEzB,WAAW,I;;UAGP,OAAZ,kBAAY,EAAK,IAAL,EAAW,OAAX,EAAoB,IAApB,C;;UAGE,oB;U
AAuB,SAAZ,kBAAY,O;UAArC,uD;YACI,gBAAc,mBAAY,OAAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,I
A/BsE,CAAU,wBA+BlE,kFA/BkE,CA+BhF,C;cACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,S;;cAEzB,WAA
W,I;;UAGnB,UAAU,mBAAY,OAAZ,C;UAEV,mBAAc,CAAd,YAAsB,IAAtB,Y;YACI,gBAAc,mBAAY,OAAZ,

C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA5CsE,CAAU,wBA4ClE,kFA5CkE,CA4ChF,C;cACI,mBAAY,OAA
Z,IAAuB,S;cACvB,UAAU,mBAAY,OAAZ,C;;cAEV,WAAW,I;;;;QAIvB,IAAI,QAAJ,C;UACI,YAAO,mBAAY,
UAAU,WAAV,IAAZ,C;QAEX,yBAAO,Q;;;MAvDuD,6B;K;6CAElE,oB;MAAkE,0B;;QAW5C,wD;QART,aAAL,
IAAK,U;QAAL,Y;UAA8B,SAAZ,kB5KoxOnB,YAAQ,C;;Q4KpxOX,W;UACI,yBAAO,K;UAAP,2B;;QAEJ,WA
1XgD,mBAAY,cA0XnC,SA1XmC,IAAZ,C;QA2XhD,cAAc,W;QACd,eAAe,K;QAEf,IAAI,cAAO,IAAX,C;UACI,
iBAAc,WAAd,UAAyB,IAAzB,U;YACI,cAAc,mBAAY,KAAZ,C;YAGd,IAf+E,wBAejE,0EAfiE,CAe/E,C;cACI,
mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,O;;cAEzB,WAAW,I;;UAGP,OAAZ,kBAAY,EAAK,IAAL,EAAW,OA
AX,EAAoB,IAApB,C;;UAGE,oB;UAAuB,SAAZ,kBAAY,O;UAArC,uD;YACI,gBAAc,mBAAY,OAAZ,C;YACd,
mBAAY,OAAZ,IAAqB,I;YAGrB,IA7B+E,wBA6BjE,kFA7BiE,CA6B/E,C;cACI,mBAAY,gBAAZ,EAAY,wBAA
Z,YAAyB,S;;cAEzB,WAAW,I;;UAGnB,UAAU,mBAAY,OAAZ,C;UAEV,mBAAc,CAAd,YAAsB,IAAtB,Y;YAC
I,gBAAc,mBAAY,OAAZ,C;YACd,mBAAY,OAAZ,IAAqB,I;YAGrB,IA1C+E,wBA0CjE,kFA1CiE,CA0C/E,C;cA
CI,mBAAY,OAAZ,IAAuB,S;cACvB,UAAU,mBAAY,OAAZ,C;;cAEV,WAAW,I;;;;QAIvB,IAAI,QAAJ,C;UACI,
YAAO,mBAAY,UAAU,WAAV,IAAZ,C;QAEX,yBAAO,Q;;;MArDuD,6B;K;2CAElE,qB;MASsB,IAII,IAJJ,EAK
M,MALN,EAaA,MAbA,EAauB,MAbvB,EAkBI,MAlBJ,EAmBM,MAnBN,EA+BI,M;MAvCb,aAAL,IAAK,U;MA
AL,Y;QAA8B,SAAZ,kB5KoxOnB,YAAQ,C;;M4KpxOX,W;QACI,OAAO,K;MAEX,WA1XgD,mBAAY,cA0XnC
,SA1XmC,IAAZ,C;MA2XhD,cAAc,W;MACd,eAAe,K;MAEf,IAAI,cAAO,IAAX,C;QACI,iBAAc,WAAd,UAAyB
,IAAzB,U;UACI,cAAc,mBAAY,KAAZ,C;UAGd,IAAI,UAAU,0EAAV,CAAJ,C;YACI,mBAAY,gBAAZ,EAAY,
wBAAZ,YAAyB,O;;YAEzB,WAAW,I;;QAGP,OAAZ,kBAAY,EAAK,IAAL,EAAW,OAAX,EAAoB,IAApB,C;;Q
AGE,oB;QAAuB,SAAZ,kBAAY,O;QAArC,uD;UACI,gBAAc,mBAAY,OAAZ,C;UACd,mBAAY,OAAZ,IAAqB,
I;UAGrB,IAAI,UAAU,kFAAV,CAAJ,C;YACI,mBAAY,gBAAZ,EAAY,wBAAZ,YAAyB,S;;YAEzB,WAAW,I;;Q
AGnB,UAAU,mBAAY,OAAZ,C;QAEV,mBAAc,CAAd,YAAsB,IAAtB,Y;UACI,gBAAc,mBAAY,OAAZ,C;UAC
d,mBAAY,OAAZ,IAAqB,I;UAGrB,IAAI,UAAU,kFAAV,CAAJ,C;YACI,mBAAY,OAAZ,IAAuB,S;YACvB,UA
AU,mBAAY,OAAZ,C;;YAEV,WAAW,I;;;;MAIvB,IAAI,QAAJ,C;QACI,YAAO,mBAAY,UAAU,WAAV,IAAZ,
C;MAEX,OAAO,Q;K;iCAGX,Y;MACI,WA7agD,mBAAY,cA6anC,SA7amC,IAAZ,C;MA8ahD,IAAI,cAAO,IAA
X,C;QACgB,OAAZ,kBAAY,EAAK,IAAL,EAAW,WAAX,EAAiB,IAAjB,C;;QACT,IvKpS6C,CAAC,cuKoS9C,C
;UACS,OAAZ,kBAAY,EAAK,IAAL,EAAW,WAAX,EAAiB,kBAAY,OAA7B,C;UACA,OAAZ,kBAAY,EAAK,I
AAL,EAAW,CAAX,EAAc,IAAd,C;;;MAEhB,cAAO,C;MACP,YAAO,C;K;2CAGX,iB;MAGe,IAAC,IAAD,EAcJ,
M;MAfP,WACW,eAAC,OAAI,KAAM,OAAN,IAAc,SAAlB,GAAwB,KAAxB,GAAmC,aAAa,KAAb,EAAoB,SA
ApB,CAApC,uB;MAEX,WA7bgD,mBAAY,cA6bnC,SA7bmC,IAAZ,C;MA8bhD,IAAI,cAAO,IAAX,C;Q5J2XJ,U
4J1XQ,kB5J0XR,E4J1X6B,I5J0X7B,EAD+F,CAC/F,E4J1XgD,W5J0XhD,E4J1XiE,I5J0XjE,C;;Q4JzXW,IvKpT6
C,CAAC,cuKoT9C,C;U5JyXX,U4JxXQ,kB5JwXR,E4JxX6B,I5JwX7B,E4JxXuD,C5JwXvD,E4JxXuE,W5JwXvE,
E4JxXwF,kBAAY,O5JwXpG,C;UAAA,U4JvXQ,kB5JuXR,E4JvX6B,I5JuX7B,E4JvXuD,kBAAY,OAAZ,GAAm
B,WAAnB,I5JuXvD,E4JvX6F,C5JuX7F,E4JvX2G,I5JuX3G,C;;;M4JrXI,IAAI,IAAK,OAAL,GAAY,SAAhB,C;Q
ACI,KAAK,SAAL,IAAa,I;;MAIjB,OAAO,qD;K;mCAGX,Y;MAEI,OAAO,qBAAQ,gBAAmB,SAAnB,OAAR,C;
K;+CAGX,iB;MAC0D,4BAAQ,KAAR,C;K;+CAC1D,Y;MAA0C,qB;K;IAE1C,gC;MAAA,oC;MACI,0BxHriBuC,
E;MwHsiBvC,sBAAiC,U;MACjC,4BAAuC,E;K;yDAEvC,oC;MAEI,kBAAkB,eAAe,eAAgB,CAA/B,K;MAClB,I
AAI,eAAc,WAAd,QAA4B,CAAhC,C;QACI,cAAc,W;MAClB,IAAI,eAAc,UAAd,QAA6B,CAAjC,C;QACI,cAAk
B,cAAc,UAAlB,GAAgC,UAAhC,GAAmD,U;MACrE,OAAO,W;K;;;IAZf,4C;MAAA,2C;QAAA,0B;;MAAA,oC;
K;qDAgBA,qB;MAEI,WAvegD,mBAAY,cAuenC,SAvemC,IAAZ,C;MAwehD,WAAe,kBAAa,cAAO,IAAxB,GA
A8B,WAA9B,GAAwC,cAAO,kBAAY,OAAnB,I;MACnD,UAAU,IAAV,EAAgB,cAAhB,C;K;;IA5iBJ,iD;MAAA,
oD;MAGwC,+B;MApB5C,sB;MAqBsB,Q;MACV,wBAAmB,CAAnB,C;QAAwB,4D;WACxB,sBAAkB,CAAlB,C
;QAAuB,uBAAa,eAAb,O;;QACf,MAAM,gCAAyB,uBAAoB,eAA7C,C;MAHlB,0B;MAJJ,Y;K;IAWA,kC;MAAA
,oD;MAGoB,+B;MA/BxB,sB;MAgCQ,sBAAc,qD;MAJlB,Y;K;IAOA,4C;MAAA,oD;MAG2C,+B;MAtC/C,sB;M
AuCQ,sBtJpB8D,YsJoBhD,QtJpBgD,C;MsJqB9D,aAAO,mBAAY,O;MACnB,IAAI,mB5KsrPD,YAAQ,C4KtrPX,
C;QAA2B,sBAAc,qD;MAN7C,Y;K;IC5BJ,4B;MAMoB,Q;M7Ky4rBA,U;MADhB,UAAe,C;MACf,uD;QAAgB,c
AAhB,iB;QACI,YAAgB,O6K34rBiB,O7K24rBjC,I;;M6K34rBJ,aAAa,iB7K64rBN,G6K74rBM,C;MACb,wBAAg
B,SAAhB,gB;QAAgB,gBAAA,SAAhB,M;QACW,SAAP,MAAO,EAAO,SAAP,C;;MAEX,OAAO,M;K;IAGX,0B;
MASiB,Q;MAFb,YAAY,iBAAa,gBAAb,C;MACZ,YAAY,iBAAa,gBAAb,C;MACZ,wBAAa,SAAb,gB;QAAa,W

AAA,SAAb,M;QACI,KAAM,WAAI,IAAK,MAAT,C;QACN,KAAM,WAAI,IAAK,OAAT,C;;MAEV,OAAO,UA
AS,KAAT,C;K;gGAGX,qB;MAWW,4B;MAAA,U;QAAqB,OAAL,S7KirPhB,YAAQ,C;;M6KjrPf,W;K;oFAGJ,m
C;MAUI,O7KoqPO,qBAAQ,C6KpqPf,GAAe,cAAf,GAAmC,S;K;IAGvC,iD;MAMI,IAAI,cAAS,KAAb,C;QAAoB
,OAAO,I;MAC3B,IAAI,qBAAgB,aAAhB,IAAiC,SAAK,OAAL,KAAa,KAAM,OAAxD,C;QAA8D,OAAO,K;MA
ErE,4C;QACI,SAAS,UAAK,CAAL,C;QACT,SAAS,MAAM,CAAN,C;QAET,IAAI,OAAO,EAAX,C;UACI,Q;eA
CG,IAAI,cAAc,UAAlB,C;UACH,OAAO,K;;QAIP,0BAAsB,kBAAtB,C;UAA4C,IAAI,CAAI,kBAAH,EAAG,EA
AkB,EAAlB,CAAR,C;YAA+B,OAAO,K;eAClF,8BAAsB,sBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAc,E
AAd,CAAR,C;YAA2B,OAAO,K;eAC9E,+BAAsB,uBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAc,EAAd,C
AAR,C;YAA2B,OAAO,K;eAC9E,6BAAsB,qBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAc,EAAd,CAAR,C
;YAA2B,OAAO,K;eAC9E,8BAAsB,sBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAc,EAAd,CAAR,C;YAA2
B,OAAO,K;eAC9E,+BAAsB,uBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAc,EAAd,CAAR,C;YAA2B,OAA
O,K;eAC9E,gCAAsB,wBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAc,EAAd,CAAR,C;YAA2B,OAAO,K;e
AC9E,8BAAsB,sBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAc,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,i
CAAsB,yBAAtB,C;UAA4C,IAAI,CAAI,cAAH,EAAG,EAAc,EAAd,CAAR,C;YAA2B,OAAO,K;eAE9E,qCAAsB
,6BAAtB,C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAc,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,sCAAsB,8BA
AtB,C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAc,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,oCAAsB,4BAAtB,
C;UAA4C,IAAI,CAAI,gBAAH,EAAG,EAAc,EAAd,CAAR,C;YAA2B,OAAO,K;eAC9E,qCAAsB,6BAAtB,C;UA
A4C,IAAI,CAAI,gBAAH,EAAG,EAAc,EAAd,CAAR,C;YAA2B,OAAO,K;eAEtE,IAAI,YAAM,EAAN,CAAJ,C;
UAAc,OAAO,K;;MAIrC,OAAO,I;K;IAGX,4C;MAKI,IAAI,iBAAJ,C;QAAkB,OAAO,M;MACzB,aAAa,CAAK,e
AAL,gBAAK,EAAa,SAAb,CAAL,GAA6C,CAA7C,QAAiD,CAAjD,I;MvC6SkB,kBAAxB,mBuC5SY,MvC4SZ,C
;MuC3SH,oDxK5BgD,gBwK4BhD,C;MADJ,O3JnCO,WoH+U6C,W;K;IuCvSxD,mE;MAEI,IAAY,SAAR,0BAAJ
,C;QACI,MAAO,gBAAO,OAAP,C;QACP,M;;MAEJ,SAAU,WAAI,SAAJ,C;MACV,MAAO,gBAAO,EAAP,C;M
AEP,4C;QACI,IAAI,MAAK,CAAT,C;UACI,MAAO,gBAAO,IAAP,C;;QAEX,cAAc,UAAK,CAAL,C;QAEV,IAD
E,OACF,S;UAAmB,MAAO,gBAAO,MAAP,C;aAC1B,mBAFE,OAEF,E;UAA2B,4BAAR,OAAQ,EAA4B,MAA5
B,EAAoC,SAApC,C;aAC3B,uBAHE,OAGF,E;UAAmB,MAAO,gBAAe,gBAAR,OAAQ,CAAf,C;aAC1B,wBAJE,
OAIF,E;UAAmB,MAAO,gBAAe,gBAAR,OAAQ,CAAf,C;aAC1B,sBALE,OAKF,E;UAAmB,MAAO,gBAAe,gB
AAR,OAAQ,CAAf,C;aAC1B,uBANE,OAMF,E;UAAmB,MAAO,gBAAe,gBAAR,OAAQ,CAAf,C;aAC1B,wBAP
E,OAOF,E;UAAmB,MAAO,gBAAe,gBAAR,OAAQ,CAAf,C;aAC1B,yBARE,OAQF,E;UAAmB,MAAO,gBAAe,
gBAAR,OAAQ,CAAf,C;aAC1B,uBATE,OASF,E;UAAmB,MAAO,gBAAe,gBAAR,OAAQ,CAAf,C;aAC1B,0BA
VE,OAUF,E;UAAmB,MAAO,gBAAe,gBAAR,OAAQ,CAAf,C;aAE1B,kBAZE,OAYF,c;UAAmB,MAAO,gBAAe
,kBAAR,OAAQ,CAAf,C;aAC1B,kBAbE,OAaF,e;UAAmB,MAAO,gBAAe,kBAAR,OAAQ,CAAf,C;aAC1B,kBA
dE,OAcF,a;UAAmB,MAAO,gBAAe,kBAAR,OAAQ,CAAf,C;aAC1B,kBAfE,OAeF,c;UAAmB,MAAO,gBAAe,k
BAAR,OAAQ,CAAf,C;;UAEP,MAAO,gBAAO,OAAQ,WAAf,C;;MAIlC,MAAO,gBAAO,EAAP,C;MACP,SAA
U,kBAAmB,iBAAV,SAAU,CAAnB,C;K;IxKjJd,yB;MAAA,6B;K;sCACI,Y;MAAkC,Y;K;0CAClC,Y;MAAsC,Y;
K;wCACtC,Y;MAAgC,Q;K;4CAChC,Y;MAAoC,S;K;mCACpC,Y;MAA+B,MAAM,6B;K;uCACrC,Y;MAAmC,
MAAM,6B;K;;;IAN7C,qC;MAAA,oC;QAAA,mB;;MAAA,6B;K;IASA,qB;MAAA,yB;MACI,+C;K;iCAEA,iB;M
AA4C,qCAAoB,KAAM,U;K;mCACtE,Y;MAA+B,Q;K;mCAC/B,Y;MAAkC,W;K;iFAEX,Y;MAAQ,Q;K;kCAC/
B,Y;MAAkC,W;K;yCAClC,mB;MAAmD,Y;K;8CAnD,oB;MAAmE,OAAA,QAAS,U;K;sCAE5E,iB;MAAwC,
MAAM,8BAA0B,iDAA8C,KAA9C,MAA1B,C;K;wCAC9C,mB;MAA8C,S;K;4CAC9C,mB;MAAkD,S;K;mCAEl
D,Y;MAA6C,kC;K;uCAC7C,Y;MAAqD,kC;K;+CACrD,iB;MACI,IAAI,UAAS,CAAb,C;QAAgB,MAAM,8BAA0
B,YAAS,KAAnC,C;MACtB,OAAO,2B;K;0CAGX,8B;MACI,IAAI,cAAa,CAAb,IAAkB,YAAW,CAAjC,C;QAAo
C,OAAO,I;MAC3C,MAAM,8BAA0B,gBAAa,SAAb,mBAAkC,OAA5D,C;K;wCAGV,Y;MAAiC,8B;K;;;IA5BrC,
iC;MAAA,gC;QAAA,e;;MAAA,yB;K;IA+BA,iC;MAA8D,6BAAkB,SAAlB,EAAoC,KAApC,C;K;IAE5B,8C;MA
AC,oB;MAA0B,0B;K;yFAClC,Y;MAAQ,OAAA,WAAO,O;K;0CACtC,Y;MAAkC,OAAA,WL4qP3B,YAAQ,C;K
;iDK3qPf,mB;MAA6C,OAAO,SAAP,WAAO,EAAS,OAAT,C;K;sDACpD,oB;MAAsE,c;;QgBkoDtD,Q;QADhB,I
AAI,chBjoDyD,QgBioDzD,iBhBjoDyD,QgBioDnC,UAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,OhBloD6C,Qg
BkoD7C,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;UAAM,IAAI,ChBloDkD,oBgBkoDvC,OhBloDuC,CgBkoDtD
,C;YAAyB,aAAO,K;YAAP,e;;;QAC/C,aAAO,I;;;MhBnoDsD,iB;K;2CAC7D,Y;MAAuC,OAAO,qBAAP,WAAO,

C;K;0CAC9C,Y;MAC+C,gBAAP,W;MAAA,OAAwB,cAAxB,GiBiKpC,SjBjKoC,GiBmKpC,SN63BoB,Q;K;;IX7
hC5B,qB;MAIsC,8B;K;IAEtC,4B;MAIqD,OAAI,QAAS,OAAT,GAAgB,CAApB,GAAgC,OAAT,QAAS,CAAhC,
GAA8C,W;K;mFAEnG,yB;MAAA,qD;MAAA,mB;QAK0C,kB;O;KAL1C,C;+FAOA,yB;MAAA,+D;MAAA,mB;
QAMwD,uB;O;KANxD,C;2FAQA,yB;MAAA,+D;MAAA,mB;QAMoD,uB;O;KANpD,C;IAQA,mC;MAKI,OAAI
,QAAS,OAAT,KAAiB,CAArB,GAAwB,gBAAxB,GAAyC,iBAAU,sBAAkB,QAAlB,EAAwC,IAAxC,CAAV,C;K
;IAE7C,iC;MAKI,OAAI,QAAS,OAAT,KAAiB,CAArB,GAAwB,gBAAxB,GAAyC,iBAAU,sBAAkB,QAAlB,EA
AwC,IAAxC,CAAV,C;K;IAE7C,gC;MAI2D,OAAI,eAAJ,GAAqB,OAAO,OAAP,CAArB,GAA0C,W;K;IAErG,m
C;MAImE,OAAS,cAAT,QAAS,C;K;gFAE5E,yB;MAaA,gE;MAbA,6B;QAyBI,WAAW,eAduE,IAcvE,C;QaCX,iB
AAc,CAAd,UbfkF,IaelF,U;UbA6B,eAf2D,IAevD,CaCtB,KbDsB,CAAJ,C;;QAfyC,OAgB/D,I;O;KA3BX,C;8FAa
A,yB;MAAA,gE;MAAA,6B;QAYI,WAAW,eAAa,IAAb,C;QaCX,iBAAc,CAAd,UbAO,IaAP,U;UbA6B,eAAI,Ka
CtB,KbDsB,CAAJ,C;;QAC7B,OAAO,I;O;KAdX,C;wFAiBA,yB;MiBzFA,+D;MjByFA,gC;QiBrF0B,gBAAf,gB;Q
jBsGkB,aa5FzB,W;Qb4FA,Oa3FO,SIXoC,Q;O;KjBqF/C,C;yFAyBA,yB;MiB3GA,4E;MAAA,gE;MjB2GA,0C;Qi
BvGI,qBjB4HyB,QiB5HzB,C;QAC8B,gBAAvB,ejB2HkB,QiB3HlB,C;QjB2H4B,aazHnC,W;QbyHA,OaxHO,SIH
4C,Q;O;KjBsGvD,C;IAkCI,mC;MAAQ,uBAAG,iBAAO,CAAP,IAAH,C;K;IAQR,qC;MAAQ,OAAA,SAAK,KAA
L,GAAY,CAAZ,I;K;4FAEZ,qB;MAK4D,QAAC,mB;K;kGAE7D,qB;MAWI,OAAO,qBAAgB,SAAK,U;K;sFAGh
C,yB;MAAA,qD;MAAA,4B;QAKgE,uCAAQ,W;O;KALxE,C;sFAOA,yB;MAAA,qD;MAAA,4B;QAKoD,uCAA
Q,W;O;KAL5D,C;sFAOA,mC;MASI,OAAI,mBAAJ,GAAe,cAAf,GAAmC,S;K;4FAGvC,+B;MAQoH,OAAA,SA
AK,qBAAY,QAAZ,C;K;IAGzH,uC;MAK+E,kBAAhB,0B;MAAwB,+B;MAAxB,Oa9MpD,W;K;IbiNX,yC;MAAk
D,QAAM,cAAN,C;aAC9C,C;UAD8C,OACzC,W;aACL,C;UAF8C,OAEzC,OAAO,sBAAK,CAAL,CAAP,C;;UAF
yC,OAGtC,S;;K;IAGZ,8D;MAgBkE,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACjG,WAAW,cAAX,E
AAiB,SAAjB,EAA4B,OAA5B,C;MAEA,UAAU,S;MACV,WAAW,UAAU,CAAV,I;MAEX,OAAO,OAAO,IAAd,
C;QACI,UAAW,GAAY,GAAN,IAAM,KAAK,C;QAC5B,aAAa,sBAAI,GAAJ,C;QACb,UAAU,cAAc,MAAd,EA
AsB,OAAtB,C;QAEV,IAAI,MAAM,CAAV,C;UACI,MAAM,MAAM,CAAN,I;aACL,IAAI,MAAM,CAAV,C;UA
CD,OAAO,MAAM,CAAN,I;;UAEP,OAAO,G;;MAEf,OAAO,EAAE,MAAM,CAAN,IAAF,K;K;IAGX,4E;MAe8E
,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MAC7G,WAAW,cAAX,EAAiB,SAAjB,EAA4B,OAA5B,C;M
AEA,UAAU,S;MACV,WAAW,UAAU,CAAV,I;MAEX,OAAO,OAAO,IAAd,C;QACI,UAAW,GAAY,GAAN,IA
AM,KAAK,C;QAC5B,aAAa,sBAAI,GAAJ,C;QACb,UAAU,UAAW,SAAQ,MAAR,EAAgB,OAAhB,C;QAErB,I
AAI,MAAM,CAAV,C;UACI,MAAM,MAAM,CAAN,I;aACL,IAAI,MAAM,CAAV,C;UACD,OAAO,MAAM,CA
AN,I;;UAEP,OAAO,G;;MAEf,OAAO,EAAE,MAAM,CAAN,IAAF,K;K;kGAGX,yB;MAAA,8D;MAAA,4D;MAs
BqC,8D;QAAA,qB;UAAE,qBAAc,iBAAS,EAAT,CAAd,EAA4B,WAA5B,C;S;O;MAtBvC,+D;QAkBI,yB;UAAA
,YAAiB,C;QACjB,uB;UAAA,UAAe,c;QAGf,+BAAa,SAAb,EAAwB,OAAxB,EAAiC,oCAAjC,C;O;KAtBJ,C;IA6
BA,mE;MAmBoC,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,c;MACnE,WAAW,cAAX,EAAiB,SAAjB,EAA
4B,OAA5B,C;MAEA,UAAU,S;MACV,WAAW,UAAU,CAAV,I;MAEX,OAAO,OAAO,IAAd,C;QACI,UAAW,G
AAY,GAAN,IAAM,KAAK,C;QAC5B,aAAa,sBAAI,GAAJ,C;QACb,UAAU,WAAW,MAAX,C;QAEV,IAAI,MA
AM,CAAV,C;UACI,MAAM,MAAM,CAAN,I;aACL,IAAI,MAAM,CAAV,C;UACD,OAAO,MAAM,CAAN,I;;U
AEP,OAAO,G;;MAEf,OAAO,EAAE,MAAM,CAAN,IAAF,K;K;IAGX,8C;MAMQ,gBAAY,OAAZ,C;QAAuB,M
AAM,gCAAyB,gBAAa,SAAb,mCAAkD,OAAlD,OAAzB,C;WAC7B,gBAAY,CAAZ,C;QAAAiB,MAAM,8BAA0B
,gBAAa,SAAb,yBAA1B,C;WACvB,cAAU,IAAV,C;QAAkB,MAAM,8BAA0B,cAAW,OAAX,gCAA2C,IAA3C,
OAA1B,C;K;IAchC,8B;MAEoC,MAAM,wBAAoB,8BAApB,C;K;IAE1C,8B;MAEoC,MAAM,wBAAoB,8BAAp
B,C;K;;;;wF0Gnb1C,yB;MzGgCA,wE;MyGhCA,uC;QAmBW,kBzGqBiD,oB;QyGM9C,Q;QAAA,OAAK,0B;QA
Af,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAAkB,sBAAY,GAAZ,C;UAClB,Wz
GyKJ,ayGzKgB,GzGyKhB,EyGvMyC,SA8BlB,CAAU,GAAV,EAAe,WAAf,EAA4B,CAA5B,EAA+B,uBAAuB,
CAAC,WAAY,mBAAY,GAAZ,CAAnE,CzGyKvB,C;;QyGvMA,OAgCO,W;O;KAnDX,C;4FAsBA,6C;MAwBc,
Q;MAAA,OAAA,SAAK,iB;MAAf,OAAU,cAAV,C;QAAU,mB;QACN,UAAU,sBAAM,CAAN,C;QACV,kBAAk
B,sBAAY,GAAZ,C;QAClB,WzGyKJ,ayGzKgB,GzGyKhB,EyGzKuB,UAAU,GAAV,EAAe,WAAf,EAA4B,CAA
5B,EAA+B,uBAAuB,CAAC,WAAY,mBAAY,GAAZ,CAAnE,CzGyKvB,C;;MyGvKA,OAAO,W;K;iFAGX,yB;M
AAA,gB;MAAA,8B;MzGtBA,wE;MyGsBA,6D;QAnCW,kBzGqBiD,oB;QyGM9C,Q;QAAA,OAAK,0B;QAAf,O
AAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAAkB,sBAAY,GAAZ,C;UA8BwE,U;UA7

B1F,WzGyKJ,ayGzKgB,GzGyKhB,EyG5IkC,UA7BD,GA6BC,EA7BoB,uBAAuB,CAAC,WAAY,mBAAY,GAA
Z,CA6BzC,GAAW,qBA7B3B,GA6B2B,EA7BT,CA6BS,CAAX,GAA6C,UA7BxD,WA6BwD,6DAA5D,EA7BiB,
CA6BjB,CzG4IlC,C;;QyG7IA,OA1BO,W;O;KAGX,C;kFA0BA,yB;MAAA,gB;MAAA,8B;MAAA,0E;QAlCc,Q;
QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBA6DQ,WA7D
U,WAAY,GAAZ,C;UA6DuF,U;UAAjG,WzG6GZ,ayGzKgB,GzGyKhB,EyG7GiD,UA5DhB,GA4DgB,EA5DK,u
BAAuB,CA4DjE,WA5D8E,mBAAY,GAAZ,CA4D1B,GAAW,qBA5D1C,GA4D0C,EA5DxB,CA4DwB,CAAX,G
AA6C,UA5DvE,WA4DuE,6DAA5D,EA5DE,CA4DF,CzG6GjD,C;;QyG9GA,OACY,W;O;KA7BhB,C;iFAgCA,y
B;MAAA,gB;MAAA,8B;MzGhFA,wE;MyGgFA,qD;QA7FW,kBzGqBiD,oB;QyGM9C,Q;QAAA,OAAK,0B;QA
Af,OAAU,cAAV,C;UAAU,mB;UACN,UAAU,sBAAM,CAAN,C;UACV,kBAAkB,sBAAY,GAAZ,C;UAkFiD,U;
UAjFnE,WzGyKJ,ayGzKgB,GzGyKhB,EyGxFgC,UAjFsB,uBAAuB,CAAC,WAAY,mBAAY,GAAZ,CAiFhD,kB
AA6B,UAjFjC,WAiFiC,6DAAvC,EAjFmB,CAiFnB,CzGwFhC,C;;QyGzFA,OA9EO,W;O;KA6DX,C;oFAoBA,yB
;MAAA,gB;MAAA,8B;MAAA,kE;QAtFc,Q;QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB;UACN,UAAU
,sBAAM,CAAN,C;UACV,kBA2GQ,WA3GU,WAAY,GAAZ,C;UA2GgE,U;UAA1E,WzG+DZ,ayGzKgB,GzGyK
hB,EyG/D+C,UA1GO,uBAAuB,CA0GjE,WA1G8E,mBAAY,GAAZ,CA0GjC,kBAA6B,UA1GhD,WA0GgD,6DA
AvC,EA1GI,CA0GJ,CzG+D/C,C;;QyGhEA,OACY,W;O;KAvBhB,C;qFA0BA,yB;MAAA,gB;MAAA,8B;MzG9H
A,wE;MyG8HA,uC;QA3IW,kBzGqBiD,oB;QyGM9C,Q;QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB;U
ACN,UAAU,sBAAM,CAAN,C;UACV,kBAAkB,sBAAY,GAAZ,C;UACC,oB;UAkIc,U;UAAjC,IAlIkD,uBAAuB,
CAAC,WAAY,mBAAY,GAAZ,CAkItF,C;YADA,mBAjI+C,C;;YAiI/C,mBACkB,UAlIW,GAkIX,EAAe,UAlIC,
WAkID,6DAAf,EAlI6B,CAkI7B,C;;UAlIlB,WzGyKJ,ayGzKgB,GzGyKhB,mB;;QyGzCA,OA9HO,W;O;KA2GX,
C;sFAwBA,yB;MAAA,gB;MAAA,8B;MAAA,oD;QAxIc,Q;QAAA,OAAK,0B;QAAf,OAAU,cAAV,C;UAAU,mB
;UACN,UAAU,sBAAM,CAAN,C;UACV,kBA6JQ,WA7JU,WAAY,GAAZ,C;UACC,oB;UA8Jc,U;UAAjC,IA9Jk
D,uBAAuB,CA4JjE,WA5J8E,mBAAY,GAAZ,CA8JtF,C;YADA,mBA7J+C,C;;YA6J/C,mBACkB,UA9JW,GA8JX
,EAAe,UA9JC,WA8JD,6DAAf,EA9J6B,CA8J7B,C;;UAFV,WzGaZ,ayGzKgB,GzGyKhB,mB;;QyGbA,OAAY,W;
O;KAvBhB,C;IA6BA,6C;MArKc,Q;MAAA,OAAK,0B;MAAf,OAAU,cAAV,C;QAAU,mB;QACN,UAAU,sBAA
M,CAAN,C;QACV,kBA+KG,WA/Ke,WAAY,GAAZ,C;QA2GgE,U;QAoE/E,WzGLP,ayGzKgB,GzGyKhB,EyGK
mC,CA9KmB,uBAAuB,CA8KtE,WA9KmF,mBAAY,GAAZ,CA0GjC,GAoErC,CApEqC,GAA6B,UA1GhD,WA0
GgD,6DAoEnD,IAAM,CAAN,IzGLnC,C;;MyGKA,OAAO,W;K;I+DnP0B,oC;MAAC,kB;MAAuB,kB;K;;wCAN7
D,Y;MAMsC,iB;K;wCANtC,Y;MAM6D,iB;K;0CAN7D,wB;MAAA,wBAMsC,qCANtC,EAM6D,qCAN7D,C;K;s
CAAA,Y;MAAA,OAMsC,mDANtC,IAM6D,wCAN7D,O;K;sCAAA,Y;MAAA,c;MAMsC,sD;MAAuB,sD;MAN7
D,a;K;oCAAA,iB;MAAA,4IAMsC,sCANtC,IAM6D,sCAN7D,I;K;wFjKEA,yB;MAAA,kC;MAAA,4C;MAAA,kD
;QAMuF,wC;O;MANvF,4CAOI,Y;QAAuC,8B;O;MAP3C,8E;MAAA,2B;QAMuF,2C;O;KANvF,C;IAcsC,2C;MA
AC,wC;K;0CACnC,Y;MAAqD,4BAAiB,wBAAjB,C;K;;IAIzD,yC;MAI4D,OAAI,oCAAJ,GAA2B,SAAK,KAAhC
,GAA0C,I;K;IAEtG,uD;MAI0E,OAAI,oCAAJ,GAA2B,SAAK,KAAhC,GAA0C,S;K;IAGpH,8B;MAMoB,Q;MAD
hB,aAAa,gB;MACG,2B;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;QACL,OAAP,MAAO,EAAO,OAAP,C;;MAEX,
OAAO,M;K;IAGX,4B;MAUiB,Q;MAHb,mBAAmB,mCAAwB,EAAxB,C;MACnB,YAAY,iBAAa,YAAb,C;MAC
Z,YAAY,iBAAa,YAAb,C;MACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,KAAM,WAAI,IAAK,MAAT,C;Q
ACN,KAAM,WAAI,IAAK,OAAT,C;;MAEV,OAAO,UAAS,KAAT,C;K;wFUxDX,qB;MAKqE,gB;K;IAErE,iC;M
AMoE,4BAAiB,SAAjB,C;K;uFAEpE,gC;MAKI,OAAgB,mBAAhB,C;QAAgB,8B;QAAM,UAAU,OAAV,C;;K;IA
MY,oC;MAAC,0B;MACnC,eAAoB,C;K;yCACpB,Y;MAAwC,OAAA,eAAS,U;K;sCACjD,Y;MAA6E,Q;MAAhC
,wBAAa,oBAAmB,mBAAnB,EAAmB,2BAAnB,QAAb,EAA0C,eAAS,OAAnD,C;K;;sFwJ5BjD,yB;MAAA,4E;M
AAA,gB;MAAA,8B;MAAA,+C;QAUiC,Q;QAA7B,OAA6B,wCAAqB,QAAS,aAA9B,0D;O;KAVjC,C;wFAYA,y
B;MAAA,4E;MAAA,gB;MAAA,8B;MAAA,+C;QAWiC,Q;QAA7B,OAA6B,wCAAqB,QAAS,aAA9B,0D;O;KA
XjC,C;sFAaA,+C;MAQI,SAAK,aAAI,QAAS,aAAb,EAAAmB,KAAAnB,C;K;ICnCT,8C;MAUI,IAAI,wCAAJ,C;QA
CI,OAAO,SAAK,4BAAqB,GAArB,C;MAET,4B;M1KuTI,Q;MALX,YAAY,oB0KlTa,G1KkTb,C;MACZ,IAAI,iB
AAiB,CAAC,4B0KnTG,G1KmTH,CAAtB,C;Q0KnTgC,MAAM,2BAAuB,wCAAvB,C;;Q1KuTlC,2BAAO,sE;;M
0KvTX,+B;K;IAGJ,8C;MAUQ,kBADE,SACF,kB;QADJ,OACkC,YAAT,SAAK,IAAI,EAAY,YAAZ,C;;QADlC,O
AEY,uBAAmB,SAAnB,EAAyB,YAAzB,C;K;IAGhB,gD;MAWQ,kBADE,SACF,yB;QADJ,OACyC,cAAT,SAAK
,IAAI,EAAY,YAAZ,C;;QADzC,OAEY,8BAA0B,SAA1B,EAAgC,YAAhC,C;K;;;;;;;IAc0B,4C;MAAC,wB;MAAo

C,0B;K;qEAApC,Y;MAAA,yB;K;0CACvC,iB;MAA4C,OAAI,OAAJ,QAAI,EAAO,KAAP,C;K;4CAChD,Y;MAA
+B,OAAI,SAAJ,QAAI,C;K;4CACnC,Y;MAAkC,OAAA,QAAI,W;K;0FACf,Y;MAAQ,OAAA,QAAI,K;K;2CACn
C,Y;MAAkC,OAAA,QAAI,U;K;qDACtC,e;MAA4C,OAAA,QAAI,mBAAY,GAAZ,C;K;uDAChD,iB;MAAgE,O
AAA,QAAI,qBAAc,KAAd,C;K;6CACpE,e;MAA+B,OAAA,QAAI,WAAI,GAAJ,C;K;0FACT,Y;MAAQ,OAAA,
QAAI,K;K;4FACH,Y;MAAQ,OAAA,QAAI,O;K;6FACJ,Y;MAAQ,OAAA,QAAI,Q;K;8DAEvD,e;MAAmD,gBA
AJ,Q;MAAI,4B;M1K+PxC,Q;MALX,YAAY,oB0K1PyD,G1K0PzD,C;MACZ,IAAI,iBAAiB,CAAC,4B0K3P+C,G
1K2P/C,CAAtB,C;QACI,2B0K5PwE,mB;;Q1K+PxE,2BAAO,sE;;M0K/PoC,+B;K;;IAGN,mD;MAAC,wB;MAA2
C,0B;K;4EAA3C,Y;MAAA,yB;K;iDAC1C,iB;MAA4C,OAAI,OAAJ,QAAI,EAAO,KAAP,C;K;mDAChD,Y;MA
A+B,OAAI,SAAJ,QAAI,C;K;mDACnC,Y;MAAkC,OAAA,QAAI,W;K;iGACf,Y;MAAQ,OAAA,QAAI,K;K;kDA
CnC,Y;MAAkC,OAAA,QAAI,U;K;4DACtC,e;MAA4C,OAAA,QAAI,mBAAY,GAAZ,C;K;8DAChD,iB;MAAgE,
OAAA,QAAI,qBAAc,KAAd,C;K;oDACpE,e;MAA+B,OAAA,QAAI,WAAI,GAAJ,C;K;iGACF,Y;MAAQ,OAAA
,QAAI,K;K;mGACH,Y;MAAQ,OAAA,QAAI,O;K;oGACU,Y;MAAQ,OAAA,QAAI,Q;K;sDAE5E,sB;MAAyC,O
AAA,QAAI,aAAI,GAAJ,EAAS,KAAT,C;K;uDAC7C,e;MAAkC,OAAA,QAAI,cAAO,GAAP,C;K;yDACtC,gB;M
AA2C,QAAI,gBAAO,IAAP,C;K;gDAC/C,Y;MAAuB,QAAI,Q;K;qEAE3B,e;MAAmD,gBAAJ,Q;MAAI,4B;M1K
0OxC,Q;MALX,YAAY,oB0KrOyD,G1KqOzD,C;MACZ,IAAI,iBAAiB,CAAC,4B0KtO+C,G1KsO/C,CAAtB,C;Q
ACI,2B0KvOwE,mB;;Q1K0OxE,2BAAO,sE;;M0K1OoC,+B;K;;I1KvFnD,oB;MAAA,wB;MACI,8C;K;gCAEA,iB
;MAA4C,oCAAsB,KAAM,U;K;kCACxE,Y;MAA+B,Q;K;kCAC/B,Y;MAAkC,W;K;gFAEX,Y;MAAQ,Q;K;iCAC
/B,Y;MAAkC,W;K;2CAElC,e;MAA+C,Y;K;6CAC/C,iB;MAAsD,Y;K;mCACtD,e;MAAwC,W;K;mFACY,Y;MA
AQ,6B;K;gFAC/B,Y;MAAQ,6B;K;kFACI,Y;MAAQ,8B;K;uCAEjD,Y;MAAiC,6B;K;;;IAjBrC,gC;MAAA,+B;QA
AA,c;;MAAA,wB;K;IAoBA,oB;MAMuE,Q;MAA7B,OAA6B,uE;K;IAEvE,wB;MAaI,OAAI,KAAM,OAAN,GAA
a,CAAjB,GAA0B,QAAN,KAAM,EAAM,qBAAc,YAAY,KAAM,OAAlB,CAAd,CAAN,CAA1B,GAA6E,U;K;kF
AEjF,yB;MAAA,oD;MAAA,mB;QAO8C,iB;O;KAP9C,C;8FASA,yB;MAAA,wE;MAAA,mB;QAQ4D,2B;O;KAR
5D,C;IAUA,+B;MAYiD,gBAA7C,qBAAoB,YAAY,KAAM,OAAlB,CAApB,C;MAAqD,wB;MAArD,OYJO,S;K;
wFZMX,yB;MAAA,4D;MAAA,mB;QAOsD,qB;O;KAPtD,C;IASA,4B;MAM8G,gBAAvC,eAAc,YAAY,KAAM,
OAAlB,CAAd,C;MAA+C,wB;MAA/C,OYrB5D,S;K;4FZuBX,yB;MAAA,wE;MAAA,mB;QAK8D,2B;O;KAL9D,
C;IAOA,8B;MAU+E,OAAM,QAAN,KAAM,EAAM,qBAAc,YAAY,KAAM,OAAlB,CAAd,CAAN,C;K;sFAErF,y
B;MgBfA,wE;MhBeA,gC;QgBXiC,gBAAtB,oB;QhB8BiB,aY9DxB,W;QZ8DA,OY7DO,SI+B2C,Q;O;KhBWtD,C
;uFA2BA,yB;MgBnCA,uE;MhBmCA,0C;QgB/ByC,gBAA9B,mBhBsDiB,QgBtDjB,C;QhBsD2B,aY7FlC,W;QZ6
FA,OY5FO,SIsCmD,Q;O;KhB+B9D,C;4FAqCA,qB;MAK+D,QAAC,mB;K;kGAEhE,qB;MAWI,OAAO,qBAAgB
,mB;K;sFAG3B,yB;MAAA,oD;MAAA,4B;QAM2D,uCAAQ,U;O;KANnE,C;sFAQA,mC;MASI,OAAI,mBAAJ,G
AAe,cAAf,GAAmC,S;K;yFAEvC,yB;MAyBA,kC;MAAA,8B;MAzBA,iC;QAgCiC,Q;QAxB2E,OAwBxD,CAAnB
,wDAAmB,oBAxBoE,GAwBpE,C;O;KAhCpD,C;+EAUA,yB;MAAA,kC;MAAA,8B;MAAA,iC;QAKiC,Q;QAA7
B,OAAgD,CAAnB,wDAAmB,YAAI,GAAJ,C;O;KALpD,C;+EAOA,iC;MAKI,sBAAI,GAAJ,EAAS,KAAT,C;K;4
FAGJ,yB;MAAA,kC;MAAA,8B;MAAA,iC;QAOiC,Q;QAA7B,OAAgD,CAAnB,wDAAmB,oBAAY,GAAZ,C;O;
KAPpD,C;gGASA,4B;MASsG,OAAA,SAAK,qBAAc,KAAd,C;K;kFAG3G,yB;MAAA,gD;MAAA,8B;MAAA,iC;
QASiC,Q;QAA7B,OAAuD,CAA1B,+DAA0B,eAAO,GAAP,C;O;KAT3D,C;6FAWA,qB;MAWoE,oB;K;6FAEpE,
qB;MAWoE,sB;K;kFAEpE,yB;MAAA,6B;MAAA,4B;QAIgE,qBAAK,aAAL,EAAU,eAAV,C;O;KAJhE,C;2FAM
A,wC;MAOiF,Q;MAAA,mCAAI,GAAJ,oBAAY,c;K;uGAG7F,yB;MAAA,gB;MAAA,8B;MAAA,+C;QAMe,Q;Q
ALX,YAAY,oBAAI,GAAJ,C;QACZ,IAAI,iBAAiB,CAAC,4BAAY,GAAZ,CAAAtB,C;UACI,OAAO,c;;UAGP,OA
AO,sE;;O;KANf,C;IAUA,oC;MAUkD,uCAAqB,GAArB,C;K;sFAElD,wC;MAWW,Q;MADP,YAAY,oBAAI,GA
AJ,C;MACL,IAAI,aAAJ,C;QACH,aAAa,c;QACb,sBAAI,GAAJ,EAAS,MAAT,C;QACA,a;;QAEA,Y;;MALJ,W;K
;wFASJ,qB;MAMwF,OAAA,iBAAQ,W;K;wFAEhG,qB;MAMgH,OAAA,iBAAQ,W;K;4FAExH,6C;Mem1BoB,Q
;MAAA,Of90BT,iBe80BS,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;Qf90Ba,We+0Bb,aAAgB,Of/0Be,Ie+0B/B,
Ef/0BsC,Se+0BZ,CAAe,OAAf,CAA1B,C;;Mf/0BhB,OAA6B,W;K;wFAGjC,6C;Me20BoB,Q;MAAA,Ofn0BT,iBe
m0BS,W;MAAhB,OAAgB,cAAhB,C;QAAgB,yB;Qfn0Ba,Weo0Bb,afp0B0B,Seo0BtB,CAAY,OAAZ,CAAJ,EAA
yC,Ofp0BC,Meo0B1C,C;;Mfp0BhB,OAA6B,W;K;IAGjC,kC;MAIyB,Q;MAArB,wBAAqB,KAArB,gB;QAAqB,a
AAA,KAArB,M;QAAK,IAAC,yBAAD,EAAM,2B;QACP,sBAAI,GAAJ,EAAS,KAAT,C;;K;IAIR,oC;MAIyB,Q;
MAAA,uB;MAArB,OAAqB,cAArB,C;QAAqB,wB;QAAhB,IAAC,yBAAD,EAAM,2B;QACP,sBAAI,GAAJ,EAA

S,KAAT,C;;K;IAIR,oC;MAIyB,Q;MAAA,uB;MAArB,OAAqB,cAArB,C;QAAqB,wB;QAAhB,IAAC,yBAAD,EA
AM,2B;QACP,sBAAI,GAAJ,EAAS,KAAT,C;;K;wFAIR,yB;MAAA,0D;MAAA,uE;MAAA,uC;QASW,kBAAY,m
BAAoB,YAAY,cAAZ,CAApB,C;Qe4xBH,Q;QAAA,Of90BT,iBe80BS,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB
;Uf90Ba,We+0Bb,aAAgB,Of/0Be,Ie+0B/B,Ef7xB2C,Se6xBjB,CAAe,OAAf,CAA1B,C;;Qf7xBhB,OAlD6B,W;O;
KAyCjC,C;oFAYA,yB;MAAA,0D;MAAA,uE;MAAA,uC;QAYW,kBAAU,mBAAoB,YAAY,cAAZ,CAApB,C;Qe
6wBD,Q;QAAA,Ofn0BT,iBem0BS,W;QAAhB,OAAgB,cAAhB,C;UAAgB,yB;Ufn0Ba,Weo0Bb,af9wByC,Se8wB
rC,CAAY,OAAZ,CAAJ,EAAyC,Ofp0BC,Meo0B1C,C;;Qf9wBhB,OAtD6B,W;O;KA0CjC,C;0FAeA,yB;MAAA,w
E;MAAA,uC;QAQkB,Q;QADd,aAAa,oB;QACC,OAAA,SA3FsE,QAAQ,W;QA2F5F,OAAc,cAAd,C;UAAc,uB;U
ACV,IAAI,UAAU,KAAM,IAAhB,CAAJ,C;YACI,MAAO,aAAI,KAAM,IAAV,EAAe,KAAM,MAArB,C;;;QAGf,
OAAO,M;O;KAbX,C;8FAgBA,yB;MAAA,wE;MAAA,uC;QAQkB,Q;QADd,aAAa,oB;QACC,OAAA,SA3GsE,Q
AAQ,W;QA2G5F,OAAc,cAAd,C;UAAc,uB;UACV,IAAI,UAAU,KAAM,MAAhB,CAAJ,C;YACI,MAAO,aAAI,
KAAM,IAAV,EAAe,KAAM,MAArB,C;;;QAGf,OAAO,M;O;KAbX,C;yFAiBA,6C;MAOoB,Q;MAAA,OAAA,SA
3HoE,QAAQ,W;MA2H5F,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAAI,UAAU,OAAV,CAAJ,C;UACI,WAAY,aA
AI,OAAQ,IAAZ,EAAiB,OAAQ,MAAzB,C;;;MAGpB,OAAO,W;K;qFAGX,yB;MAAA,wE;MAAA,uC;QAOW,k
BAAS,oB;QAfA,Q;QAAA,OA3HoE,iBAAQ,W;QA2H5F,OAAgB,cAAhB,C;UAAgB,yB;UACZ,IAcmC,SAd/B,C
AAU,OAAV,CAAJ,C;YACI,WAAY,aAAI,OAAQ,IAAZ,EAAiB,OAAQ,MAAzB,C;;;QAapB,OAVO,W;O;KAGX
,C;+FAUA,6C;MAOoB,Q;MAAA,OAAA,SApJoE,QAAQ,W;MAoJ5F,OAAgB,cAAhB,C;QAAgB,yB;QACZ,IAA
I,CAAC,UAAU,OAAV,CAAL,C;UACI,WAAY,aAAI,OAAQ,IAAZ,EAAiB,OAAQ,MAAzB,C;;;MAGpB,OAAO,
W;K;2FAGX,yB;MAAA,wE;MAAA,uC;QAOW,kBAAY,oB;QAfH,Q;QAAA,OApJoE,iBAAQ,W;QAoJ5F,OAAg
B,cAAhB,C;UAAgB,yB;UACZ,IAAI,CAckC,SAdjC,CAAU,OAAV,CAAL,C;YACI,WAAY,aAAI,OAAQ,IAAZ,E
AAiB,OAAQ,MAAzB,C;;;QAapB,OAVO,W;O;KAGX,C;IAUA,0B;MAQqB,IAAN,I;MADX,IAAI,oCAAJ,C;QA
CW,QAAM,cAAN,C;eACH,C;YAAK,iB;YAAL,K;eACA,C;YAAK,aAAU,8BAAJ,GAAkB,sBAAK,CAAL,CAAl
B,GAA+B,oBAAW,OAAhD,C;YAAL,K;;YACQ,0BAAM,qBAAoB,YAAY,cAAZ,CAApB,CAAN,C;YAHL,K;;Q
AAP,W;;MAMJ,OAAoC,oBAA7B,mBAAM,oBAAN,CAA6B,C;K;IAGxC,yC;MAIwB,SAApB,WAAoB,Y;MAA
pB,kB;K;IAEJ,4B;MAM6D,QAAM,gBAAN,C;aACzD,C;UADyD,OACpD,U;aACL,C;UAFyD,OAEpD,MAAM,U
AAK,CAAL,CAAN,C;;UAFoD,OAGjD,mBAAM,qBAAoB,YAAY,gBAAZ,CAApB,CAAN,C;;K;IAGZ,yC;MAI
wB,OAApB,WAAoB,Y;MAApB,kB;K;IAEJ,4B;MAM4D,OAA6B,oBAA7B,mBAAM,oBAAN,CAA6B,C;K;IAEz
F,yC;MAIwB,SAApB,WAAoB,Y;MAApB,kB;K;IAEJ,4B;MAMqD,QAAM,cAAN,C;aACjD,C;UADiD,OAC5C,U
;aACL,C;UAFiD,OgBlY8B,uB;;UhBkY9B,OAGzC,uB;;K;IAGZ,iC;MAMmE,4BAAc,SAAd,C;K;IAEnE,yC;MAK
I,WAAoB,0B;MAApB,kB;K;IAEJ,kC;MAOI,Q;MAAA,IAAI,SAAK,UAAT,C;QAAA,OAAAoB,MAAM,IAAN,C;;
QAAqC,kBAApB,qBAAc,SAAd,C;QAA4B,wBAAS,UAAT,EAAqB,WAArB,C;QAAjE,OYpiBO,W;;MZoiBP,W;
K;IAEJ,mC;MAOI,Q;MAAA,IAAI,SAAK,UAAT,C;QAAA,OAA0B,MAAN,KAAM,C;;QAAiC,kBAApB,qBAAc
,SAAd,C;QAA4B,4B;QAAnE,OY7iBO,W;;MZ6iBP,W;K;IAEJ,mC;MAOI,Q;MAAA,IAAI,SAAK,UAAT,C;QAA
A,OAA0B,QAAN,KAAM,C;;QAAiC,kBAApB,qBAAc,SAAd,C;QAA4B,0B;QAAnE,OYtjBO,W;;MZsjBP,W;K;I
AEJ,mC;MAOwB,kBAApB,qBAAc,SAAd,C;MAA4B,4B;MAA5B,OAA4C,oBY/jBrC,WZ+jBqC,C;K;IAEhD,iC;
MAOwB,kBAApB,qBAAc,SAAd,C;MAA4B,+B;MAA5B,OYxkBO,W;K;0FZ2kBX,2B;MAKI,sBAAI,IAAK,MA
AT,EAAgB,IAAK,OAArB,C;K;4FAGJ,yB;MAAA,gD;MAAA,mC;QAKI,kBAAO,KAAP,C;O;KALJ,C;4FAQA,y
B;MAAA,gD;MAAA,mC;QAKI,kBAAO,KAAP,C;O;KALJ,C;4FAQA,yB;MAAA,gD;MAAA,mC;QAKI,kBAAO,
KAAP,C;O;KALJ,C;4FAQA,0B;MAKI,yBAAO,GAAP,C;K;IAGJ,kC;MAOwB,kBAAf,aAAL,SAAK,C;MAsCL,6
B;MAtCA,OAA+C,oBY1nBxC,WZ0nBwC,C;K;IAEnD,mC;MAQwB,kBAAf,aAAL,SAAK,C;MAqCK,YAAL,gB
AAK,O;MArCV,OAAgD,oBYpoBzC,WZooByC,C;K;IAEpD,mC;MAQwB,kBAAf,aAAL,SAAK,C;MAoCK,YAA
L,gBAAK,O;MApCV,OAAgD,oBY9oBzC,WZ8oByC,C;K;IAEpD,mC;MAQwB,kBAAf,aAAL,SAAK,C;MAmC
K,YAAL,gBAAK,O;MAnCV,OAAgD,oBYxpBzC,WZwpByC,C;K;4FAEpD,0B;MAMI,uBAAO,GAAP,C;K;8FA
GJ,yB;MAAA,sD;MAAA,kC;QAMc,UAAV,SAAK,KAAK,EAAU,IAAV,C;O;KANd,C;8FASA,yB;MAAA,sD;M
AAA,kC;QAMc,UAAV,SAAK,KAAK,EAAU,IAAV,C;O;KANd,C;8FASA,yB;MAAA,sD;MAAA,kC;QAMc,UA
AV,SAAK,KAAK,EAAU,IAAV,C;O;KANd,C;IAUA,wC;MACsD,QAAM,cAAN,C;aAClD,C;UADkD,OAC7C,U;
aACL,C;UAFkD,gB;;UAAA,OAG1C,S;;K;oF2K5wBZ,yB;MAAA,8D;MAAA,8B;MAAA,qC;QAUiC,Q;QAA7B,
OAA2D,CAA9B,sEAA8B,eAAO,OAAP,C;O;KAV/D,C;wFAYA,yB;MAAA,8D;MAAA,8B;MAAA,sC;QASiC,Q;

QAA7B,OAA2D,CAA9B,sEAA8B,oBAAU,QAAV,C;O;KAT/D,C;wFAWA,yB;MAAA,8D;MAAA,8B;MAAA,s
C;QASiC,Q;QAA7B,OAA2D,CAA9B,sEAA8B,oBAAU,QAAV,C;O;KAT/D,C;4FAWA,8B;MAKI,SAAK,WAAI,
OAAJ,C;K;4FAGT,yB;MAAA,gD;MAAA,sC;QAKS,OAAL,SAAK,EAAO,QAAP,C;O;KALT,C;4FAQA,yB;MA
AA,gD;MAAA,sC;QAKS,OAAL,SAAK,EAAO,QAAP,C;O;KALT,C;4FAQA,yB;MAAA,gD;MAAA,sC;QAKS,O
AAL,SAAK,EAAO,QAAP,C;O;KALT,C;8FAQA,8B;MAKI,SAAK,cAAO,OAAP,C;K;8FAGT,yB;MAAA,sD;MA
AA,sC;QAKS,UAAL,SAAK,EAAU,QAAV,C;O;KALT,C;8FAQA,yB;MAAA,sD;MAAA,sC;QAKS,UAAL,SAA
K,EAAU,QAAV,C;O;KALT,C;8FAQA,yB;MAAA,sD;MAAA,sC;QAKS,UAAL,SAAK,EAAU,QAAV,C;O;KAL
T,C;IAQA,qC;MAIU,IAIe,I;MAHjB,kBADE,QACF,c;QAAiB,OAAO,yBAAO,QAAP,C;;QAEpB,aAAsB,K;QAC
T,0B;QAAb,OAAa,cAAb,C;UAAa,sB;UACT,IAAI,oBAAI,IAAJ,CAAJ,C;YAAe,SAAS,I;;QAC5B,OAAO,M;;K;I
AKnB,uC;MAKiB,Q;MADb,aAAsB,K;MACT,0B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,IAAI,oBAAI,IAAJ,CA
AJ,C;UAAe,SAAS,I;;MAE5B,OAAO,M;K;IAGX,uC;MAII,OAAO,yBAAgB,OAAT,QAAS,CAAhB,C;K;IAGX,i
D;MAKI,OAAI,oCAAJ,GAAwB,SAAxB,GAAkC,mB;K;IAEtC,0C;MAII,OAAO,4BAAmB,6BAAT,QAAS,CAAn
B,C;K;IAGX,0C;MAII,WAAoB,UAAT,QAAS,C;MACpB,O5KuEwD,C4KvEjD,I5KuEkD,U4KvElD,IAAqB,4BA
AU,IAAV,C;K;IAGhC,0C;MAII,OjLuoPO,EiLvoPA,QjL+jPA,YAAQ,CAwER,CiLvoPA,IAAyB,4BAAmB,OAA
T,QAAS,CAAnB,C;K;IAGpC,0C;MAII,OAAO,4BAAmB,6BAAT,QAAS,CAAnB,C;K;IAGX,0C;MAII,IjLynPO,
EiLznPH,QjLijPG,YAAQ,CAwER,CiLznPP,C;QACI,OAAO,4BAAmB,OAAT,QAAS,CAAnB,C;;QAEP,OAAO,
wB;K;IAGf,0C;MAII,WAAoB,UAAT,QAAS,C;MACpB,I5KuCwD,C4KvCpD,I5KuCqD,U4KvCzD,C;QACI,OA
AO,4BAAU,IAAV,C;;QAEP,OAAO,wB;K;IAGf,kC;MACI,a5KgCwD,CAAC,mB;M4K/BzD,iB;MACA,OAAO,
M;K;IAIX,2C;MAKkF,gCAAc,SAAd,EAAyB,IAAzB,C;K;IAElF,2C;MAKkF,gCAAc,SAAd,EAAyB,KAAzB,C;K
;IAElF,sE;MACI,iBAAa,KAAb,C;M/JvJgB,kB+JwJX,oB;MACD,OAAO,qBAAP,C;QACI,IAAI,UAAU,kBAAV,6
BAAJ,C;UACI,oB;UACA,WAAS,I;;MAGrB,OAAO,Q;K;oFAIX,4B;MAM6D,kCAAS,KAAT,C;K;IAE7D,gC;M
AKiD,IAAI,mBAAJ,C;QAAe,MAAM,2BAAuB,gBAAvB,C;;QAArB,OAAmE,2BAAS,CAAT,C;K;IAEpH,sC;MA
KwD,OAAI,mBAAJ,GAAe,IAAf,GAAyB,2BAAS,CAAT,C;K;IAEjF,+B;MAKgD,IAAI,mBAAJ,C;QAAe,MAA
M,2BAAuB,gBAAvB,C;;QAArB,OAAmE,2BAAS,2BAAT,C;K;IAEnH,qC;MAKuD,OAAI,mBAAJ,GAAe,IAAf,
GAAyB,2BAAS,2BAAT,C;K;IAEhF,2C;MAK8E,kCAAc,SAAd,EAAyB,IAAzB,C;K;IAE9E,2C;MAK8E,kCAAc,
SAAd,EAAyB,KAAzB,C;K;IAE9E,wE;MAEgB,UAGS,MAHT,EAcY,MAdZ,EAc6B,M;MAfzC,IAAI,uCAAJ,C;
QACI,OAAoC,cAA5B,sEAA4B,EAAc,SAAd,EAAyB,uBAAzB,C;MAExC,iBAAsB,C;MACD,oC;MAArB,qBAA
kB,CAAlB,mC;QACI,cAAc,sBAAK,SAAL,C;QACd,IAAI,UAAU,OAAV,MAAAsB,uBAA1B,C;UACI,Q;QAEJ,IA
AI,eAAc,SAAlB,C;UACI,sBAAK,UAAL,EAAAmB,OAAAnB,C;QAEJ,+B;;MAEJ,IAAI,aAAa,cAAjB,C;QACwB,oC
;QAAiB,mB;QAArC,oE;UACI,2BAAS,WAAT,C;QAEJ,OAAO,I;;QAEP,OAAO,K;;K;ICtSf,wB;K;kCAEI,Y;MA
A4B,sB;K;;IAMhC,wB;K;kCAEI,Y;MAA4B,mC;K;;IAMhC,yB;K;mCAEI,Y;MAA4B,uB;K;;IAMhC,uB;K;iCAEI
,Y;MAA4B,qB;K;;IAMhC,wB;K;kCAEI,Y;MAA4B,sB;K;;IAMhC,yB;K;mCAEI,Y;MAA4B,uB;K;;IAMhC,0B;K;
oCAEI,Y;MAA4B,wB;K;;IAMhC,2B;K;qCAEI,Y;MAA4B,yB;K;;ICzDc,wC;MAAkC,uB;MAAjC,0B;K;4FACpB,
Y;MAAQ,OAAA,eAAS,K;K;iDACxC,iB;MAAkC,mCAAS,0BAAoB,KAApB,CAAT,C;K;;IAGT,gC;MAAyC,8B;
MAAxC,0B;K;oFACH,Y;MAAQ,OAAA,eAAS,K;K;yCACxC,iB;MAAkC,mCAAS,0BAAoB,KAApB,CAAT,C;K
;mCAElC,Y;MAAuB,eAAS,Q;K;8CAChC,iB;MAAuC,OAAA,eAAS,kBAAS,0BAAoB,KAApB,CAAT,C;K;yCA
EhD,0B;MAA8C,OAAA,eAAS,aAAI,0BAAoB,KAApB,CAAJ,EAAgC,OAAhC,C;K;yCACvD,0B;MACI,eAAS,a
AAI,2BAAqB,KAArB,CAAJ,EAAiC,OAAjC,C;K;;IAIjB,+C;MACoB,Q;MAAA,kC;MAAhB,IAAa,CAAT,0BAAJ
,C;QAAA,OAA2B,8BAAY,KAAZ,I;;QAAuB,MAAM,8BAA0B,mBAAgB,KAAhB,2BAA0C,gBAAG,2BAAH,C
AA1C,OAA1B,C;K;IAE5D,gD;MACoB,Q;MAAA,qB;MAAhB,IAAa,CAAT,0BAAJ,C;QAAA,OAAsB,iBAAO,K
AAP,I;;QAAkB,MAAM,8BAA0B,oBAAiB,KAAjB,2BAA2C,gBAAG,cAAH,CAA3C,OAA1B,C;K;IAGlD,+B;M
AK+C,gCAAqB,SAArB,C;K;IAE/C,iC;MAM6D,wBAAa,SAAb,C;K;;;;IrKpC7D,oD;MAQuF,wC;K;IARvF,8CAS
I,Y;MAAuC,8B;K;IAT3C,gF;IsKa8G,wC;MAAA,mB;QAAE,kBAAS,aAAT,C;O;K;IAVhH,yB;MAUqG,oCAAS,s
BAAT,C;K;IAErG,2B;MASI,eAAe,6B;MACf,oBAA0B,+BAAN,KAAM,EAAwC,QAAxC,EAA+D,QAA/D,C;MA
C1B,OAAO,Q;K;IAc+B,yB;K;+CAoBtC,kC;MAOI,IAAI,uCAA0B,QAAS,UAAvC,C;QAAkD,M;MAClD,OAAO,
sBAAS,QAAS,WAAlB,e;K;+CAGX,kC;MAQqD,6BAAS,QAAS,WAAlB,e;K;;;;;;;;IAYzD,mC;MAA2C,wB;MAC
vC,eAAoB,C;MACpB,mBAA4B,I;MAC5B,sBAAyC,I;MACzC,gBAAoC,I;K;gDAEpC,Y;MACI,OAAO,IAAP,C;
QACI,QAAM,YAAN,C;eACI,C;YAAA,K;eACA,C;YACI,IAAI,kCAAe,UAAnB,C;cACI,eAAQ,C;cACR,OAAO,I

;;cAEP,sBAAe,I;;;YALvB,K;eAOA,C;YAAc,OAAO,K;eACrB,C;eAAA,C;YAAgC,OAAO,I;;YAC/B,MAAM,yB;;
QAGlB,eAAQ,C;QACR,WAAW,4B;QACX,gBAAW,I;QACX,I5HpFR,oBDgDQ,W6HoCY,kB7HpCZ,CChDR,C;
;K;6C4HwFA,Y;MACU,IASe,I;MATrB,QAAM,YAAN,C;aACI,C;aAAA,C;UAAsC,OAAO,qB;aAC7C,C;UACI,e
AAQ,C;UACR,OAAO,kCAAe,O;aAE1B,C;UACI,eAAQ,C;UACR,aACa,mF;UACb,mBAAY,I;UACZ,OAAO,M;;
UAEH,MAAM,yB;;K;uDAItB,Y;MACI,IAAI,CAAC,cAAL,C;QAAgB,MAAM,6B;;QAA8B,OAAO,W;K;2DAG/
D,Y;MAA4C,QAAM,YAAN,C;aACxC,C;UADwC,OAC1B,6B;aACd,C;UAFwC,OAExB,6BAAsB,sBAAtB,C;;U
AFwB,OAGhC,6BAAsB,uCAAoC,YAA1D,C;;K;IAOqC,4E;MAAA,oB;QACzC,wCAAW,C;QAAX,OACA,yB;O;
K;oDALR,+B;MACI,mBAAY,K;MACZ,eAAQ,C;MACR,OAA6C,0CAAtC,c;K;IAUsC,+E;MAAA,oB;QACzC,w
CAAW,C;QAAX,OACA,yB;O;K;yDANR,kC;MACI,IAAI,CAAC,QAAS,UAAd,C;QAAyB,M;MACzB,sBAAe,Q;
MACf,eAAQ,C;MACR,OAA6C,6CAAtC,c;K;2DAMX,kB;M7HNO,Q;MADP,e6HSI,M7HTJ,C;MACO,Q6HQH,
M7HRG,+D;M6HSH,eAAQ,C;K;kGAIR,Y;MAAQ,0C;K;;ItK/KhB,oD;MAQuF,wC;K;IARvF,8CASI,Y;MAAuC,
8B;K;IAT3C,gF;sFAAA,yB;MAAA,kC;MAAA,0C;MAAA,kD;QAQuF,wC;O;MARvF,4CASI,Y;QAAuC,8B;O;M
AT3C,8E;MAAA,2B;QAQuF,2C;O;KARvF,C;IAiBgE,+C;MAAA,mB;QAAE,sB;O;K;IALlE,kC;MAKuD,OAAk
B,2CAAT,+BAAS,E;K;IAEzE,8B;MAK6D,OAAI,QdksPtD,YAAQ,CclsP0C,GAAwB,eAAxB,GAAsD,WAAT,Q
AAS,C;K;IAEnH,yB;MAG8C,kC;K;IAE9C,yB;MAAA,6B;K;uCACI,Y;MAA6C,kC;K;2CAC7C,a;MAA4B,kC;K;
2CAC5B,a;MAA4B,kC;K;;;IAHhC,qC;MAAA,oC;QAAA,mB;;MAAA,6B;K;oFAMA,yB;MAAA,2D;MAAA,4B;
QAM4D,uCAAQ,e;O;KANpE,C;IAgB4F,mH;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,wC;MAAA,wD;MAAA,
kC;K;;;;kDAAA,Y;;;;;cACxF,eAAe,uBAAa,W;cAC5B,IAAI,QAAS,UAAb,C;gBACI,gB;gCAAA,sCAAS,QAAT,
O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBAEA,gB;gCAAA,sCAAS,iCAAT,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;
;;;;;cAJJ,W;;cAAA,W;;;;;;;;;;;;;;;;K;IADwF,gE;MAAA,yD;uBAAA,uG;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAP5
F,4C;MAOmF,gBAAS,uCAAT,C;K;IAgBb,4B;MAAE,OAAA,EAAG,W;K;IAP3E,8B;MAO8D,4BAAQ,cAAR,C;
K;IAUQ,8B;MAAE,OAAA,EAAG,W;K;IAR3E,8B;MAQ8D,4BAAQ,gBAAR,C;K;IAM1B,8B;MAAE,S;K;IAJtC,
wC;MAEgB,Q;MADZ,IAAI,8CAAJ,C;QACI,OAA4C,CAApC,2EAAoC,kBAAQ,QAAR,C;;MAEhD,OAAO,uBA
AmB,SAAnB,EAAyB,gBAAzB,EAAiC,QAAjC,C;K;IAGX,4B;MAYiB,Q;MAFb,YAAY,gB;MACZ,YAAY,gB;M
ACC,2B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,KAAM,WAAI,IAAK,MAAT,C;QACN,KAAM,WAAI,IAAK,O
AAT,C;;MAEV,OAAO,UAAS,KAAT,C;K;IAGX,+B;MAQqD,6BAAS,4BAAT,C;K;IAW0B,+G;MAAA,wC;MA
AA,6B;MAAA,yB;MAAA,0C;MAAA,4C;MAAA,0B;MAAA,kC;K;;;;mDAAA,Y;;;;;kCAC9D,0C;cACb,gB;;;;;cA
AA,IAAO,iBT2FkD,US3FzD,C;gBAAA,gB;;;;cACI,QAAQ,yBAAO,iBAAQ,iBAAO,KAAf,C;cACf,WAAkB,WA
AP,iBAAO,C;cAClB,YAAgB,IAAI,iBAAO,KAAf,GAAqB,iBAAO,aAAI,CAAJ,EAAO,IAAP,CAA5B,GAA8C,I;
cAC1D,gB;8BAAA,iCAAM,KAAN,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cAJJ,gB;;;cAMJ,W;;;;;;;;;;;;;;K;IAR+E,
4D;MAAA,yD;uBAAA,mG;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAT/E,uC;MASmE,gBAAY,kCAAZ,C;K;IAkB
hC,0D;MAE/B,wB;QAAA,WAAgC,I;MADhC,0B;MACA,0B;MACA,4B;K;IAGuC,0E;MAAA,oD;MACnC,gBA
Ae,iCAAS,W;MACxB,iBAAqB,E;MACrB,gBAAmB,I;K;oEAEnB,Y;MACI,OAAO,aAAS,UAAhB,C;QACI,WA
AW,aAAS,O;QACpB,IAAI,wCAAU,IAAV,MAAmB,sCAAvB,C;UACI,gBAAW,I;UACX,iBAAY,C;UACZ,M;;;
MAGR,iBAAY,C;K;8DAGhB,Y;MASW,Q;MARP,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,IAAI,mBAAa,CAAj
B,C;QACI,MAAM,6B;MACV,aAAa,a;MACb,gBAAW,I;MACX,iBAAY,E;MAEZ,OAAO,yE;K;iEAGX,Y;MACI,
IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,OAAO,mBAAa,C;K;;2CAhC5B,Y;MAAuC,yD;K;;IA2C3C,qD;MAAY,
0B;MAAmC,gC;K;IACJ,gF;MAAA,0D;MACnC,gBAAe,oCAAS,W;K;iEACxB,Y;MACI,OAAO,6CAAY,aAAS,O
AArB,C;K;oEAGX,Y;MACI,OAAO,aAAS,U;K;;8CAPxB,Y;MAAuC,4D;K;qDAWvC,oB;MACI,OAAO,uBAA4B
,eAA5B,EAAsC,kBAAtC,EAAmD,QAAnD,C;K;;IAUf,4D;MAAY,0B;MAAmC,gC;K;IACJ,8F;MAAA,wE;MAC
nC,gBAAe,2CAAS,W;MACxB,aAAY,C;K;wEACZ,Y;MAC0C,Q;MAAtC,OAAO,oDAAY,oBAAmB,iBAAnB,E
AAmB,yBAAnB,QAAZ,EAAyC,aAAS,OAAlD,C;K;2EAGX,Y;MACI,OAAO,aAAS,U;K;;qDARxB,Y;MAAuC,m
E;K;;IAkB3C,oC;MAAY,0B;K;IAC6C,wE;MACjD,gBAAe,gCAAS,W;MACxB,aAAY,C;K;6DACZ,Y;MAC2C,Q
;MAAvC,OAAO,iBAAa,oBAAmB,iBAAnB,EAAmB,yBAAnB,QAAb,EAA0C,aAAS,OAAnD,C;K;gEAGX,Y;M
ACI,OAAO,aAAS,U;K;;0CARxB,Y;MAAqD,wD;K;;IAmBzD,0D;MACI,4B;MACA,4B;MACA,4B;K;IAEuC,sE;
MAAA,gD;MACnC,iBAAgB,gCAAU,W;MAC1B,iBAAgB,gCAAU,W;K;4DAC1B,Y;MACI,OAAO,sCAAU,cA
AU,OAApB,EAA4B,cAAU,OAAtC,C;K;+DAGX,Y;MACI,OAAO,cAAU,UAAV,IAAuB,cAAU,U;K;;yCARhD,Y
;MAAuC,uD;K;;IAc3C,6D;MACI,0B;MACA,gC;MACA,0B;K;IAEuC,4E;MAAA,sD;MACnC,gBAAe,kCAAS,W;

MACxB,oBAAiC,I;K;+DAEjC,Y;MACI,IAAI,CAAC,2BAAL,C;QACI,MAAM,6B;MACV,OAAO,gCAAe,O;K;k
EAG1B,Y;MACI,OAAO,2B;K;+EAGX,Y;MACQ,Q;MAAJ,IAAI,iEAA2B,KAA/B,C;QACI,oBAAe,I;MAEnB,O
AAO,yBAAP,C;QACI,IAAI,CAAC,aAAS,UAAd,C;UACI,OAAO,K;;UAEP,cAAc,aAAS,O;UACvB,uBAAuB,wC
AAS,2CAAY,OAAZ,CAAT,C;UACvB,IAAI,gBAAiB,UAArB,C;YACI,oBAAe,gB;YACf,OAAO,I;;;;MAInB,OA
AO,I;K;;4CA9Bf,Y;MAAuC,0D;K;;IAoC9B,6I;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,4C;MAAA,kD;MAAA
,gD;MAAA,wB;MAAA,yB;MAAA,kC;K;;;;yDAAA,Y;;;;;kBAGyC,I;iCAFlC,C;cACI,sD;cAAhB,gB;;;;;cAAA,KA
AgB,yBAAhB,C;gBAAA,gB;;;;cAAgB,oC;cACZ,aAAa,6BAAU,oBAAmB,uBAAnB,EAAmB,+BAAnB,QAAV,E
AAuC,OAAvC,C;cACb,gB;8BAAA,sCAAS,4BAAS,MAAT,CAAT,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cAFJ,gB
;;;cAIJ,W;;;;;;;;;;;;;;K;IANS,0F;MAAA,yD;uBAAA,iI;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IADb,wD;MACI,gBA
AS,kDAAT,C;K;;;;IAoByB,qD;MACzB,0B;MACA,8B;MACA,0B;MC3TA,IAAI,ED+TQ,qBAAc,CC/TtB,CAAJ,
C;QACI,cD8T2B,+CAA4C,iB;QC7TvE,MAAM,gCAAyB,OAAQ,WAAjC,C;;MAFV,IAAI,EDgUQ,mBAAY,CCh
UpB,CAAJ,C;QACI,gBD+TyB,6CAA0C,e;QC9TnE,MAAM,gCAAyB,SAAQ,WAAjC,C;;MAFV,IAAI,EDiUQ,m
BAAY,iBCjUpB,CAAJ,C;QACI,gBDgUkC,0DAAuD,eAAvD,WAAmE,iB;QC/TrG,MAAM,gCAAyB,SAAQ,WA
AjC,C;;K;sFDkUa,Y;MAAQ,yBAAW,iBAAX,I;K;yCAE/B,a;MAAyC,OAAI,KAAK,YAAT,GAAgB,eAAhB,GA
AqC,gBAAY,eAAZ,EAAsB,oBAAa,CAAb,IAAtB,EAAsC,eAAtC,C;K;yCAC9E,a;MAAyC,OAAI,KAAK,YAAT,
GAAgB,IAAhB,GAA0B,gBAAY,eAAZ,EAAsB,iBAAtB,EAAkC,oBAAa,CAAb,IAAlC,C;K;IAEzC,8D;MAAA,w
C;MAEtB,gBAAe,2BAAS,W;MACxB,gBAAe,C;K;0DAEf,Y;MAEI,OAAO,gBAAW,kCAAX,IAAyB,aAAS,UAA
zC,C;QACI,aAAS,O;QACT,qC;;K;2DAIR,Y;MACI,a;MACA,OAAQ,gBAAW,gCAAZ,IAAyB,aAAS,U;K;wDAG
7C,Y;MACI,a;MACA,IAAI,iBAAY,gCAAhB,C;QACI,MAAM,6B;MACV,qC;MACA,OAAO,aAAS,O;K;;qCAvB
xB,Y;MAA0B,mD;K;;IAgCA,uC;MAC1B,0B;MACA,oB;MC3WA,IAAI,ED+WQ,gBAAS,CC/WjB,CAAJ,C;QA
CI,cD8WsB,yCAAsC,YAAtC,M;QC7WtB,MAAM,gCAAyB,OAAQ,WAAjC,C;;K;0CDgXV,a;MAAyC,OAAI,K
AAK,YAAT,GAAgB,eAAhB,GAAqC,gBAAY,eAAZ,EAAsB,CAAtB,EAAyB,YAAzB,C;K;0CAC9E,a;MAAyC,
OAAI,KAAK,YAAT,GAAgB,IAAhB,GAA0B,iBAAa,eAAb,EAAuB,CAAvB,C;K;IAE5B,gE;MACnC,YAAW,yB
;MACX,gBAAe,4BAAS,W;K;yDAExB,Y;MACI,IAAI,cAAQ,CAAZ,C;QACI,MAAM,6B;MACV,6B;MACA,OA
AO,aAAS,O;K;4DAGpB,Y;MACI,OAAO,YAAO,CAAP,IAAY,aAAS,U;K;;sCAZpC,Y;MAAuC,oD;K;;IAsB3C,g
D;MACI,0B;MACA,4B;K;IAEuC,0E;MAAA,oD;MACnC,gBAAe,iCAAS,W;MACxB,iBAAqB,E;MACrB,gBAA
mB,I;K;oEAEnB,Y;MACI,IAAI,aAAS,UAAb,C;QACI,WAAW,aAAS,O;QACpB,IAAI,wCAAU,IAAV,CAAJ,C;U
ACI,iBAAY,C;UACZ,gBAAW,I;UACX,M;;;MAGR,iBAAY,C;K;8DAGhB,Y;MAMiB,Q;MALb,IAAI,mBAAa,E
AAjB,C;QACI,iB;MACJ,IAAI,mBAAa,CAAjB,C;QACI,MAAM,6B;MACV,aACa,gF;MAGb,gBAAW,I;MACX,i
BAAY,E;MACZ,OAAO,M;K;iEAGX,Y;MACI,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,OAAO,mBAAa,C;K;;2C
AlC5B,Y;MAAuC,yD;K;;IA2Cb,uC;MAC1B,0B;MACA,oB;MC5bA,IAAI,ED+bQ,gBAAS,CC/bjB,CAAJ,C;QAC
I,cD8bsB,yCAAsC,YAAtC,M;QC7btB,MAAM,gCAAyB,OAAQ,WAAjC,C;;K;0CDgcV,a;MItXO,SJsXmC,eAAQ
,CAAR,I;MAAD,OAA4B,KAAK,CAAT,GAAY,yBAAZ,GAAuC,iBAAa,eAAb,EAAuB,EAAvB,C;K;0CACxG,a;
MIvXO,SJuXmC,eAAQ,CAAR,I;MAAD,OAA4B,KAAK,CAAT,GAAY,yBAAZ,GAAuC,gBAAY,eAAZ,EAAsB,
YAAtB,EAA6B,EAA7B,C;K;IAEjE,gE;MACnC,gBAAe,4BAAS,W;MACxB,YAAW,yB;K;2DAEX,Y;MAEI,OA
AO,YAAO,CAAP,IAAY,aAAS,UAA5B,C;QACI,aAAS,O;QACT,6B;;K;yDAIR,Y;MACI,a;MACA,OAAO,aAAS,
O;K;4DAGpB,Y;MACI,a;MACA,OAAO,aAAS,U;K;;sCAnBxB,Y;MAAuC,oD;K;;IA6B3C,gD;MACI,0B;MACA,
4B;K;IAGuC,0E;MAAA,oD;MACnC,gBAAe,iCAAS,W;MACxB,iBAAqB,E;MACrB,gBAAmB,I;K;gEAEnB,Y;
MACI,OAAO,aAAS,UAAhB,C;QACI,WAAW,aAAS,O;QACpB,IAAI,CAAC,wCAAU,IAAV,CAAL,C;UACI,gB
AAW,I;UACX,iBAAY,C;UACZ,M;;;MAGR,iBAAY,C;K;8DAGhB,Y;MAMqB,Q;MALjB,IAAI,mBAAa,EAAjB,
C;QACI,a;MAEJ,IAAI,mBAAa,CAAjB,C;QACI,aACa,gF;QACb,gBAAW,I;QACX,iBAAY,C;QACZ,OAAO,M;;
MAEX,OAAO,aAAS,O;K;iEAGpB,Y;MACI,IAAI,mBAAa,EAAjB,C;QACI,a;MACJ,OAAO,mBAAa,CAAb,IAA
kB,aAAS,U;K;;2CAlC1C,Y;MAAuC,yD;K;;IAuCN,+C;MAAAC,sB;MAAAiC,gC;K;0CACnE,Y;MAAuC,4BAAiB,a
AAO,WAAxB,EAAoC,kBAApC,C;K;;IAGP,+C;MAAuE,2B;MAAtE,sB;MAAiC,gC;MAClE,kBAAuB,c;K;6CAE
vB,Y;MACI,OAAO,aAAO,UAAd,C;QACI,WAAW,aAAO,O;QAClB,UAAU,mBAAY,IAAZ,C;QAEV,IAAI,eAA
S,WAAI,GAAAJ,CAAb,C;UACI,mBAAQ,IAAR,C;UACA,M;;;MAIR,W;K;;IAKgC,0D;MAAC,wC;MAAuC,kC;K;
IACrC,0E;MAAA,oD;MACnC,gBAAmB,I;MACnB,iBAAqB,E;K;oEAErB,Y;MACI,gBAAe,mBAAa,EAAjB,GA
AqB,+CAArB,GAA4C,2CAAa,4BAAb,C;MACvD,iBAAgB,qBAAJ,GAAAsB,CAAAtB,GAA6B,C;K;8DAG7C,Y;M

AMiB,Q;MALb,IAAI,iBAAY,CAAhB,C;QACI,iB;MAEJ,IAAI,mBAAa,CAAjB,C;QACI,MAAM,6B;MACV,aAA
a,8D;MAEb,iBAAY,E;MACZ,OAAO,M;K;iEAGX,Y;MACI,IAAI,iBAAY,CAAhB,C;QACI,iB;MACJ,OAAO,mB
AAa,C;K;;2CAxB5B,Y;MAAuC,yD;K;;IA6B3C,kC;MAWI,OAAW,iDAAJ,GAAwC,SAAxC,GAAkD,4BAAwB,S
AAxB,C;K;IAelB,uD;MAAA,qB;QAAE,6B;O;K;IAX7C,wC;MAWI,OAA2D,cAApD,sBAAkB,YAAlB,EAAgC,q
CAAhC,CAAoD,C;K;IAqBrC,iD;MAAA,mB;QAAE,mB;O;K;IAlB5B,gD;MAeI,OAAI,YAAJ,GACI,2BADJ,GA
GI,sBAAkB,+BAAlB,EAA4B,YAA5B,C;K;IAER,wD;MAcI,6BAAkB,YAAlB,EAAgC,YAAhC,C;K;IPxpBJ,oB;
MAAA,wB;MACI,8C;K;gCAEA,iB;MAA4C,oCAAmB,KAAM,U;K;kCACrE,Y;MAA+B,Q;K;kCAC/B,Y;MAAk
C,W;K;gFAEX,Y;MAAQ,Q;K;iCAC/B,Y;MAAkC,W;K;wCAClC,mB;MAAmD,Y;K;6CACnD,oB;MAAmE,OAA
A,QAAS,U;K;kCAE5E,Y;MAA6C,kC;K;uCAE7C,Y;MAAiC,6B;K;;;IAdrC,gC;MAAA,+B;QAAA,c;;MAAA,wB;
K;IAkBA,oB;MAIoC,6B;K;IAEpC,2B;MAMmD,OAAI,QAAS,OAAT,GAAgB,CAApB,GAAgC,MAAT,QAAS,C
AAhC,GAA6C,U;K;iFAEhG,yB;MAAA,mD;MAAA,mB;QAKwC,iB;O;KALxC,C;6FAOA,yB;MAAA,uE;MAAA
,mB;QAQsD,2B;O;KARtD,C;IAUA,kC;MAKiE,OAAS,aAAT,QAAS,EAAAa,qBAAc,YAAY,QAAS,OAArB,CAA
d,CAAb,C;K;uFAE1E,yB;MAAA,2D;MAAA,mB;QAGgD,qB;O;KAHhD,C;IAKA,+B;MAC2D,OAAS,aAAT,QA
AS,EAAa,eAAQ,YAAY,QAAS,OAArB,CAAR,CAAb,C;K;2FAEpE,yB;MAAA,uE;MAAA,mB;QAMwD,2B;O;K
ANxD,C;IAQA,iC;MAKmE,OAAS,aAAT,QAAS,EAAa,qBAAc,YAAY,QAAS,OAArB,CAAd,CAAb,C;K;IAE5E,
+B;MAMyD,OAAI,eAAJ,GAAqB,MAAM,OAAN,CAArB,GAAyC,U;K;IAElG,kC;MAQI,OAAgB,gBAAT,QAA
S,EAAgB,sBAAhB,C;K;sFAGpB,yB;MetBA,uE;MfsBA,gC;QelB8B,gBAAnB,oB;QfqCiB,aWhDxB,W;QXgDA,O
W/CO,SIUwC,Q;O;KfkBnD,C;wFA2BA,yB;Me1CA,wE;Mf0CA,0C;QetCsC,gBAA3B,mBf6DiB,Qe7DjB,C;Qf6D
2B,aW/ElC,W;QX+EA,OW9EO,SIiBgD,Q;O;KfsC3D,C;sFAgCA,yB;MAAA,mD;MAAA,4B;QAEkD,uCAAQ,U;
O;KAF1D,C;IAIA,wC;MAAgD,QAAM,cAAN,C;aAC5C,C;UAD4C,OACvC,U;aACL,C;UAF4C,OAEvC,MAAM,
oBAAW,OAAjB,C;;UAFuC,OAGpC,S;;K;IOrKZ,oD;MAQuF,wC;K;IARvF,8CASI,Y;MAAuC,8B;K;IAT3C,gF;Iu
KLA,yC;MtK4BI,IAAI,EsK3BI,OAAO,CAAP,IAAY,OAAO,CtK2BvB,CAAJ,C;QACI,csK3BI,aAAJ,GACI,yEA
DJ,GAGI,8C;QtKyBJ,MAAM,gCAAyB,OAAQ,WAAjC,C;;K;IsKnBM,mI;MAAA,mB;QAAE,wBAAiB,gCAAjB,
EAA6B,YAA7B,EAAmC,YAAnC,EAAyC,sBAAzC,EAAyD,mBAAzD,C;O;K;IAFtB,gF;MACI,oBAAoB,IAApB,
EAA0B,IAA1B,C;MACA,oCAAgB,6EAAhB,C;K;IAKyB,yL;MAAA,wC;MAAA,6B;MAAA,yB;MAAA,wC;MA
AA,wC;MAAA,gD;MAAA,sD;MAAA,4D;MAAA,wB;MAAA,0B;MAAA,uB;MAAA,0B;MAAA,wB;MAAA,qB;
MAAA,4B;MAAA,kC;K;;;;2DAAA,Y;;;;;cACrB,4BAAiC,eAAL,uBAAK,EAAa,IAAb,C;+BACvB,0BAAO,uBAA
P,I;cACV,IAAI,kBAAO,CAAX,C;oCACiB,iBAAa,qBAAb,C;kCACF,C;gBACD,6C;gBAAV,iB;;;sCAaa,gBAAc,q
BAAd,C;gBACH,+C;gBAAV,gB;;;;;;cAAA,KAAU,2BAAV,C;gBAAAA,gB;;;;cAAU,kC;cACN,mBAAO,WAAI,G
AAJ,C;cACP,IAAI,mBAAO,SAAX,C;gBACI,IAAI,mBAAO,KAAP,GAAc,uBAAlB,C;kBAA0B,sBAAS,mBAAO
,kBAAuB,uBAAvB,C;kBAA8B,gB;;;kBAAxE,gB;;;;gBADJ,gB;;;;;cAGI,gB;8BAAA,iCAAU,8BAAJ,GAAiB,mB
AAjB,GAA6B,iBAAU,mBAAV,CAAnC,O;kBAAA,2C;uBAAA,yB;cAAA,Q;;cACA,mBAAO,qBAAY,uBAAZ,C
;cAJX,gB;;;cAFJ,gB;;;cASA,IAAI,iCAAJ,C;gBACI,gB;;;gBADJ,iB;;;;;cACI,IAAO,mBAAO,KAAd,IAAqB,uBAA
rB,C;gBAAA,gB;;;;cACI,gB;8BAAA,iCAAU,8BAAJ,GAAiB,mBAAjB,GAA6B,iBAAU,mBAAV,CAAnC,O;kBA
AA,2C;uBAAA,yB;cAAA,Q;;cACA,mBAAO,qBAAY,uBAAZ,C;cAFX,gB;;;cAIA,IhL8K4C,CgL9KxC,mBhL8K
yC,UgL9K7C,C;gBAAyB,iB;gCAAA,iCAAM,mBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBAAzB,iB;;;;;cAj
CR,W;;cA4BI,iB;;;cA1BJ,iB;;;cAGI,KAAU,yBAAV,C;gBAAA,iB;;;;6BAAU,sB;cACN,IAAI,kBAAO,CAAX,C;g
BAAgB,oCAAQ,CAAR,I;gBAAW,iB;;;gBAA3B,iB;;;;;cACA,iBAAO,WAAI,YAAJ,C;cACP,IAAI,iBAAO,KAAP
,KAAe,uBAAnB,C;gBACI,iB;gCAAA,iCAAM,iBAAN,O;oBAAA,2C;yBAAA,yB;gBAAA,Q;;gBADJ,iB;;;;;cAEI,
IAAI,8BAAJ,C;gBAAAiB,iBAAO,Q;;gBAAa,oBAAS,iBAAU,uBAAV,C;cAC9C,kBAAO,c;cAHX,iB;;;cAHJ,iB;;;c
ASA,IhLiMgD,CgLjM5C,iBhLiM6C,UgLjMjD,C;gBACI,IAAI,qCAAkB,iBAAO,KAAP,KAAe,uBAArC,C;kBAA
2C,iB;kCAAA,iCAAM,iBAAN,O;sBAAA,2C;2BAAA,yB;kBAAA,Q;;kBAA3C,iB;;;;gBADJ,iB;;;;;cAdJ,W;;cAcI,
iB;;;cAZJ,iB;;;cAkCJ,W;;;;;;;;;;;;;;K;IArCyB,sI;MAAA,yD;uBAAA,6K;YAAA,S;iBAAA,Q;;iBAAA,uB;O;K;IAF
7B,6E;MACI,IAAI,CAAC,QAAS,UAAd,C;QAAyB,OAAO,2B;MAChC,OAAO,WAAkB,0EAAlB,C;K;IAwCwB,
6B;MAA8B,uB;MAA7B,kB;MAChC,mBAA6B,C;MAC7B,eAAyB,C;K;2CAEzB,8B;MACI,+DAAkB,SAAlB,EA
A6B,OAA7B,EAAsC,WAAK,KAA3C,C;MACA,mBAAiB,S;MACjB,eAAa,UAAU,SAAV,I;K;0CAGjB,iB;MACI
,+DAAkB,KAAlB,EAAyB,YAAzB,C;MAEA,OAAO,wBAAK,mBAAY,KAAZ,IAAAL,C;K;qFAGY,Y;MAAQ,mB
;K;;IASR,wC;MAAqD,uB;MAApD,sB;MtKrDxB,IAAI,EsKuDQ,cAAc,CtKvDtB,CAAJ,C;QACI,csKsD2B,wE;Qt

KrD3B,MAAM,gCAAyB,OAAQ,WAAjC,C;;MAFV,IAAI,EsKwDQ,cAAc,aAAO,OtKxD7B,CAAJ,C;QACI,gBsK
uDqC,wFAA+E,aAAO,O;QtKtD3H,MAAM,gCAAyB,SAAQ,WAAjC,C;;MsK2DV,kBAAuB,aAAO,O;MAC9B,o
BAA8B,C;MAE9B,sBAAyB,U;K;kFAAzB,Y;MAAA,0B;K,OAAA,gB;MAAA,0B;K;uCAGA,iB;MAGW,Q;MAF
P,+DAAkB,KAAlB,EAAyB,SAAzB,C;MAEA,OAAO,sBAmGmC,CAnG5B,iBAmG6B,GAnGV,KAmGU,IAAD,I
AAa,eAAb,IAnGnC,4D;K;kCAGX,Y;MAAe,qBAAQ,e;K;IAEgB,4D;MAAA,sC;MAAS,2B;MAC5C,eAAoB,oB;
MACpB,eAAoB,4B;K;8DAEpB,Y;MAKgB,Q;MAJZ,IAAI,iBAAS,CAAb,C;QACI,W;;QAGA,mBAAQ,sCAAO,Y
AAP,4DAAR,C;QACA,eAoFkC,CApF1B,YAoF2B,GApFb,CAoFa,IAAD,IAAa,+BAAb,I;QAnFlC,mC;;K;;oCAX
Z,Y;MAAuC,kD;K;2CAgBvC,iB;MAGiE,UAQ1C,MAR0C,EAe1C,MAf0C,EAqBtD,M;MAtBP,aACQ,KAAM,O
AAN,GAAa,IAAK,KAAtB,GAAkC,UAAN,KAAM,EAAO,IAAK,KAAZ,CAAlC,GAAyD,kD;MAE7D,WAAW,I
AAK,K;MAEhB,WAAW,C;MACX,UAAU,iB;MAEV,OAAO,OAAO,IAAP,IAAe,MAAM,eAA5B,C;QACI,OAA
O,IAAP,IAAe,wBAAO,GAAP,gE;QACf,mB;QACA,iB;;MAGJ,MAAM,C;MACN,OAAO,OAAO,IAAd,C;QACI,
OAAO,IAAP,IAAe,wBAAO,GAAP,gE;QACf,mB;QACA,iB;;MAEJ,IAAI,MAAO,OAAP,GAAc,IAAK,KAAvB,C
;QAA6B,OAAO,IAAK,KAAZ,IAAoB,I;MAEjD,OAAO,uD;K;mCAGX,Y;MACI,OAAO,qBAAQ,gBAAa,SAAb,
OAAR,C;K;4CAGX,uB;MAKI,kBAAoD,eAAjC,mBAAY,mBAAa,CAAzB,IAA8B,CAA9B,IAAiC,EAAa,WAAb,
C;MACpD,gBAAoB,sBAAc,CAAlB,GAA4B,UAAP,aAAO,EAAO,WAAP,CAA5B,GAAqD,qBAAQ,gBAAa,WA
Ab,OAAR,C;MACrE,OAAO,eAAW,SAAX,EAAsB,SAAtB,C;K;qCAGX,mB;MAII,IAAI,aAAJ,C;QACI,MAAM,
6BAAsB,qBAAtB,C;;MAGV,cA6B0C,CA7BnC,iBA6BoC,GA7BjB,SA6BiB,IAAD,IAAa,eAAb,IA7B1C,IAAmC,
O;MACnC,6B;K;+CAGJ,a;MtKhJA,IAAI,EsKoJQ,KAAK,CtKpJb,CAAJ,C;QACI,csKmJkB,wC;QtKlJlB,MAAM,
gCAAyB,OAAQ,WAAjC,C;;MAFV,IAAI,EsKqJQ,KAAK,StKrJb,CAAJ,C;QACI,gBsKoJqB,wEAA8D,S;QtKnJn
F,MAAM,gCAAyB,SAAQ,WAAjC,C;;MsKqJN,IAAI,IAAI,CAAR,C;QACI,YAAY,iB;QACZ,UAgBsC,CAhB5B5B,
KAgB6B,GAhBf,CAgBe,IAAD,IAAa,eAAb,I;QAdtC,IAAI,QAAQ,GAAZ,C;UACW,OAAP,aAAO,EAAK,IAAL,
EAAW,KAAX,EAAkB,eAAlB,C;UACA,OAAP,aAAO,EAAK,IAAL,EAAW,CAAX,EAAc,GAAd,C;;UAEA,OA
AP,aAAO,EAAK,IAAL,EAAW,KAAX,EAAkB,GAAlB,C;;QAGX,oBAAa,G;QACb,wBAAQ,CAAR,I;;K;qCAK
R,wB;MAC8C,QAAC,YAAO,CAAP,IAAD,IAAa,eAAb,I;K;;IA9G9C,0C;MAAA,oD;MAA6B,uBAAK,gBAAmB,
QAAnB,OAAL,EAAmC,CAAnC,C;MAA7B,Y;K;ICvFJ,0C;MAII,QAAQ,I;MACR,QAAQ,K;MACR,YAAY,kBA
AM,CAAC,OAAO,KAAP,IAAD,IAAiB,CAAjB,IAAN,C;MACZ,OAAO,KAAK,CAAZ,C;QACI,OpL+B4E,0BoL/
BrE,kBAAM,CAAN,CpLoR2B,KAAL,GAAiB,GArP8B,EoL/B1D,KpLoRgB,KAAL,GAAiB,GArP8B,CoL/BrE,I
AAP,C;UACI,a;;QACJ,OpL6B4E,0BoL7BrE,kBAAM,CAAN,CpLkR2B,KAAL,GAAiB,GArP8B,EoL7B1D,KpLk
RgB,KAAL,GAAiB,GArP8B,CoL7BrE,IAAP,C;UACI,a;;QACJ,IAAI,KAAK,CAAT,C;UACI,UAAU,kBAAM,CA
AN,C;UACV,kBAAM,CAAN,EAAW,kBAAM,CAAN,CAAX,C;UACA,kBAAM,CAAN,EAAW,GAAX,C;UACA
,a;UACA,a;;;MAGR,OAAO,C;K;IAGX,uC;MAGI,YAAY,aAAU,KAAV,EAAAiB,IAAjB,EAAuB,KAAvB,C;MAC
Z,IAAI,QAAO,QAAQ,CAAR,IAAP,CAAJ,C;QACI,UAAU,KAAV,EAAAiB,IAAjB,EAAuB,QAAQ,CAAR,IAAvB
,C;MACJ,IAAI,QAAQ,KAAZ,C;QACI,UAAU,KAAV,EAAAiB,KAAjB,EAAwB,KAAxB,C;K;IAGR,0C;MAII,QA
AQ,I;MACR,QAAQ,K;MACR,YAAY,kBAAM,CAAC,OAAO,KAAP,IAAD,IAAiB,CAAjB,IAAN,C;MACZ,OA
AO,KAAK,CAAZ,C;QACI,OlLM6E,0BkLNtE,kBAAM,CAAN,ClLoP2B,KAAL,GAAiB,KA9O+B,EkLN3D,KlL
oPgB,KAAL,GAAiB,KA9O+B,CkLNtE,IAAP,C;UACI,a;;QACJ,OlLI6E,0BkLJtE,kBAAM,CAAN,ClLkP2B,KAA
L,GAAiB,KA9O+B,EkLJ3D,KlLkPgB,KAAL,GAAiB,KA9O+B,CkLJtE,IAAP,C;UACI,a;;QACJ,IAAI,KAAK,CA
AT,C;UACI,UAAU,kBAAM,CAAN,C;UACV,kBAAM,CAAN,EAAW,kBAAM,CAAN,CAAX,C;UACA,kBAAM
,CAAN,EAAW,GAAX,C;UACA,a;UACA,a;;;MAGR,OAAO,C;K;IAGX,yC;MAGI,YAAY,aAAU,KAAV,EAAAiB,
IAAjB,EAAuB,KAAvB,C;MACZ,IAAI,QAAO,QAAQ,CAAR,IAAP,CAAJ,C;QACI,YAAU,KAAV,EAAAiB,IAAj
B,EAAuB,QAAQ,CAAR,IAAvB,C;MACJ,IAAI,QAAQ,KAAZ,C;QACI,YAAU,KAAV,EAAAiB,KAAjB,EAAwB,
KAAxB,C;K;IAGR,0C;MAII,QAAQ,I;MACR,QAAQ,K;MACR,YAAY,kBAAM,CAAC,OAAO,KAAP,IAAD,IA
AiB,CAAjB,IAAN,C;MACZ,OAAO,KAAK,CAAZ,C;QACI,OnLnB8D,YmLmBvD,kBAAM,CAAN,CnLnBwE,K
AAjB,EmLmB5C,KnLnByE,KAA7B,CmLmBvD,IAAP,C;UACI,a;;QACJ,OnLrB8D,YmLqBvD,kBAAM,CAAN,
CnLrBwE,KAAjB,EmLqB5C,KnLrByE,KAA7B,CmLqBvD,IAAP,C;UACI,a;;QACJ,IAAI,KAAK,CAAT,C;UACI
,UAAU,kBAAM,CAAN,C;UACV,kBAAM,CAAN,EAAW,kBAAM,CAAN,CAAX,C;UACA,kBAAM,CAAN,EA
AW,GAAX,C;UACA,a;UACA,a;;;MAGR,OAAO,C;K;IAGX,yC;MAGI,YAAY,aAAU,KAAV,EAAAiB,IAAjB,EA
AuB,KAAvB,C;MACZ,IAAI,QAAO,QAAQ,CAAR,IAAP,CAAJ,C;QACI,YAAU,KAAV,EAAAiB,IAAjB,EAAuB,

QAAQ,CAAR,IAAvB,C;MACJ,IAAI,QAAQ,KAAZ,C;QACI,YAAU,KAAV,EAAiB,KAAjB,EAAwB,KAAxB,C;
K;IAGR,0C;MAII,QAAQ,I;MACR,QAAQ,K;MACR,YAAY,kBAAM,CAAC,OAAO,KAAP,IAAD,IAAiB,CAAjB
,IAAN,C;MACZ,OAAO,KAAK,CAAZ,C;QACI,OlK5C+D,akK4CxD,kBAAM,CAAN,ClK5C0E,KAAlB,EkK4C7
C,KlK5C2E,KAA9B,CkK4CxD,IAAP,C;UACI,a;;QACJ,OlK9C+D,akK8CxD,kBAAM,CAAN,ClK9C0E,KAAlB,
EkK8C7C,KlK9C2E,KAA9B,CkK8CxD,IAAP,C;UACI,a;;QACJ,IAAI,KAAK,CAAT,C;UACI,UAAU,kBAAM,C
AAN,C;UACV,kBAAM,CAAN,EAAW,kBAAM,CAAN,CAAX,C;UACA,kBAAM,CAAN,EAAW,GAAX,C;UAC
A,a;UACA,a;;;MAGR,OAAO,C;K;IAGX,yC;MAGI,YAAY,aAAU,KAAV,EAAiB,IAAjB,EAAuB,KAAvB,C;MA
CZ,IAAI,QAAO,QAAQ,CAAR,IAAP,CAAJ,C;QACI,YAAU,KAAV,EAAiB,IAAjB,EAAuB,QAAQ,CAAR,IAAv
B,C;MACJ,IAAI,QAAQ,KAAZ,C;QACI,YAAU,KAAV,EAAiB,KAAjB,EAAwB,KAAxB,C;K;IAKR,gD;MAI6E,
UAAU,KAAV,EAAiB,SAAjB,EAA4B,UAAU,CAAV,IAA5B,C;K;IAC7E,gD;MAC6E,YAAU,KAAV,EAAiB,SA
AjB,EAA4B,UAAU,CAAV,IAA5B,C;K;IAC7E,gD;MAC6E,YAAU,KAAV,EAAiB,SAAjB,EAA4B,UAAU,CAA
V,IAA5B,C;K;IAC7E,gD;MAC6E,YAAU,KAAV,EAAiB,SAAjB,EAA4B,UAAU,CAAV,IAA5B,C;K;IrK9I7E,0C
;MF0BI,IAAI,EEjBI,SAAU,OAAV,GAAiB,CFiBrB,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAyB,OAAQ,WA
AjC,C;;MElBV,OAAO,oBAAoB,CAApB,EAAuB,CAAvB,EAA0B,SAA1B,C;K;IAGX,8C;MACe,Q;MAAX,wBA
AW,SAAX,gB;QAAW,SAAA,SAAX,M;QACI,SAAS,GAAG,CAAH,C;QACT,SAAS,GAAG,CAAH,C;QACT,W
AAW,cAAc,EAAd,EAAkB,EAAlB,C;QACX,IAAI,SAAQ,CAAZ,C;UAAe,OAAO,I;;MAE1B,OAAO,C;K;sGAG
X,yB;MAAA,8D;MAAA,iC;QASI,OAAO,cAAc,SAAS,CAAT,CAAd,EAA2B,SAAS,CAAT,CAA3B,C;O;KATX,
C;sGAYA,sC;MASI,OAAO,UAAW,SAAQ,SAAS,CAAT,CAAR,EAAqB,SAAS,CAAT,CAArB,C;K;IAatB,6B;M
AWY,Q;MALR,IAAI,MAAM,CAAV,C;QAAa,OAAO,C;MACpB,IAAI,SAAJ,C;QAAe,OAAO,E;MACtB,IAAI,S
AAJ,C;QAAe,OAAO,C;MAGtB,OAA8B,iBAAtB,mDAAsB,EAAU,CAAV,C;K;IAaZ,6C;MAAA,uB;QAAU,2BA
AoB,CAApB,EAAuB,CAAvB,EAA0B,iBAA1B,C;O;K;IAVhC,8B;MF7CI,IAAI,EEsDI,SAAU,OAAV,GAAiB,CF
tDrB,CAAJ,C;QACI,cAda,qB;QAeb,MAAM,gCAAyB,OAAQ,WAAjC,C;;MEqDV,OAAO,eAAW,2BAAX,C;K;0
FAIX,yB;MAAA,sC;MAAA,oC;MAAA,uBAOe,yB;QArEf,8D;eAqEe,4B;UAAA,uB;YAAU,eAAsB,gB;YAAtB,
OA5Dd,cAAc,SA4DgB,CA5DhB,CAAd,EAA2B,SA4DM,CA5DN,CAA3B,C;W;S;OA4DI,C;MAPf,2B;QAOI,sB
AAW,0BAAX,C;O;KAPJ,C;0FASA,yB;MAAA,oC;MAQe,gE;QAAA,uB;UAAU,iBAAsB,kB;UAAtB,eAAkC,gB;
UAAlC,OA1Dd,UAAW,SAAQ,SA0DW,CA1DX,CAAR,EAAqB,SA0DC,CA1DD,CAArB,C;S;O;MAkDtB,uC;Q
AQI,sBAAW,sCAAX,C;O;KARJ,C;4GAUA,yB;MAAA,sC;MAAA,oC;MAAA,iCAOe,yB;QAxFf,8D;eAwFe,4B;
UAAA,uB;YAAU,eAAsB,gB;YAAtB,OA/Ed,cAAc,SA+EgB,CA/EhB,CAAd,EAA2B,SA+EM,CA/EN,CAA3B,C;
W;S;OA+EI,C;MAPf,2B;QAOI,sBAAW,oCAAX,C;O;KAPJ,C;8GASA,yB;MAAA,oC;MAUe,0E;QAAA,uB;UAA
U,iBAAsB,kB;UAAtB,eAAkC,gB;UAAlC,OA/Ed,UAAW,SAAQ,SA+EW,CA/EX,CAAR,EAAqB,SA+EC,CA/ED
,CAArB,C;S;O;MAqEtB,uC;QAUI,sBAAW,gDAAX,C;O;KAVJ,C;kFAYA,yB;MAAA,sC;MAAA,oC;MAAA,oB
AQe,yB;QA9Gf,8D;eA8Ge,yC;UAAA,uB;YACP,sBAAsB,WAAY,SAAQ,CAAR,EAAW,CAAX,C;YAClC,Q;YA
AA,IAAI,oBAAmB,CAAvB,C;cAAA,OAA0B,e;;cAAqB,eAAsB,gB;cAArE,OAvGG,cAAc,SAuG8C,CAvG9C,C
AAd,EAA2B,SAuGoC,CAvGpC,CAA3B,C;;YAsGH,W;W;S;OADO,C;MARf,sC;QAQI,sBAAW,kCAAX,C;O;K
ARJ,C;oFAaA,yB;MAAA,oC;MAQe,0E;QAAA,uB;UACP,sBAAsB,WAAY,SAAQ,CAAR,EAAW,CAAX,C;UA
ClC,Q;UAAA,IAAI,oBAAmB,CAAvB,C;YAAA,OAA0B,e;;YAAqB,iBAAsB,kB;YAAtB,eAAkC,gB;YAAjF,OA
xGG,UAAW,SAAQ,SAwGyC,CAxGzC,CAAR,EAAqB,SAwG+B,CAxG/B,CAArB,C;;UAuGd,W;S;O;MATR,kD
;QAQI,sBAAW,8CAAX,C;O;KARJ,C;sGAaA,yB;MAAA,sC;MAAA,oC;MAAA,8BAQe,yB;QAxIf,8D;eAwIe,m
D;UAAA,uB;YACP,sBAAsB,qBAAsB,SAAQ,CAAR,EAAW,CAAX,C;YAC5C,Q;YAAA,IAAI,oBAAmB,CAAv
B,C;cAAA,OAA0B,e;;cAAqB,eAAsB,gB;cAArE,OAjIG,cAAc,SAiI8C,CAjI9C,CAAd,EAA2B,SAiIoC,CAjIpC,C
AA3B,C;;YAgIH,W;W;S;OADO,C;MARf,sC;QAQI,sBAAW,4CAAX,C;O;KARJ,C;wGAaA,yB;MAAA,oC;MAQ
e,8F;QAAA,uB;UACP,sBAAsB,qBAAsB,SAAQ,CAAR,EAAW,CAAX,C;UAC5C,Q;UAAA,IAAI,oBAAmB,CA
AvB,C;YAAA,OAA0B,e;;YAAqB,iBAAsB,kB;YAAtB,eAAkC,gB;YAAjF,OAlIG,UAAW,SAAQ,SAkIyC,CAlIz
C,CAAR,EAAqB,SAkI+B,CAlI/B,CAArB,C;;UAiId,W;S;O;MATR,kD;QAQI,sBAAW,wDAAX,C;O;KARJ,C;kG
AcA,yB;MAAA,oC;MAOe,wE;QAAA,uB;UACP,sBAAsB,mBAAoB,SAAQ,CAAR,EAAW,CAAX,C;UAA1C,O
ACI,oBAAmB,CAAvB,GAA0B,eAA1B,GAA+C,mBAAW,CAAX,EAAc,CAAd,C;S;O;MATvD,wC;QAOI,sBAA
W,4CAAX,C;O;KAPJ,C;IAmBe,oD;MAAA,uB;QACP,sBAAsB,SAAU,SAAQ,CAAR,EAAW,CAAX,C;QAAhC,
OACI,oBAAmB,CAAvB,GAA0B,eAA1B,GAA+C,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IATlE,uC;MAO

I,sBAAW,kCAAX,C;K;IAYc,wE;MAAA,uB;QACV,sBAAsB,mBAAoB,SAAQ,CAAR,EAAW,CAAX,C;QAA1C,
OACI,oBAAmB,CAAvB,GAA0B,eAA1B,GAA+C,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IATlE,+C;MAO
I,sBAAc,4CAAd,C;K;IAcW,+C;MAAA,uB;QAEH,UAAM,CAAN,C;UADJ,OACe,C;aACX,c;UAFJ,OAEiB,E;aA
Cb,c;UAHJ,OAGiB,C;;UAHjB,OAIY,kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IAb/B,gC;MAQI,sBAAW,6B
AAX,C;K;4FASJ,yB;MAAA,4D;MAAA,wD;MAAA,mB;QAQqE,kBAAW,cAAX,C;O;KARrE,C;IAkBe,8C;MAA
A,uB;QAEH,UAAM,CAAN,C;UADJ,OACe,C;aACX,c;UAFJ,OAEiB,C;aACb,c;UAHJ,OAGiB,E;;UAHjB,OAIY,
kBAAW,SAAQ,CAAR,EAAW,CAAX,C;O;K;IAb/B,+B;MAQI,sBAAW,4BAAX,C;K;0FASJ,yB;MAAA,4D;MA
AA,sD;MAAA,mB;QAQoE,iBAAU,cAAV,C;O;KARpE,C;IAUA,wB;MAO4F,Q;MAA7B,OAA6B,4F;K;IAE5F,w
B;MAO4F,Q;MAA7B,OAA6B,4F;K;IAE5F,gC;MAM+D,IAEJ,IAFI,EAGJ,M;MAFvD,kBAD2D,SAC3D,sB;QAD
qD,OAC5B,SAAK,W;WAC9B,WAF2D,SAE3D,wC;QAFqD,OAEE,4F;WACvD,WAH2D,SAG3D,wC;QAHqD,O
AGE,gG;;QAHF,OAI7C,uBAAmB,SAAnB,C;K;IAIuB,wC;MAAC,4B;K;2CAChC,gB;MAAwC,OAAA,eAAW,S
AAQ,CAAR,EAAW,CAAX,C;K;4CACnD,Y;MACgC,sB;K;;IAGpC,kC;MAAA,sC;K;+CACI,gB;MAAoE,OAAE,
iBAAF,CAAE,EAAU,CAAV,C;K;gDACtE,Y;MAC8C,2C;K;;;IAHlD,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;IA
MA,kC;MAAA,sC;K;+CACI,gB;MAAoE,OAAE,iBAAF,CAAE,EAAU,CAAV,C;K;gDACtE,Y;MAC8C,2C;K;;;I
AHlD,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;8EsKzTA,4B;MAUI,OAAK,iBAAL,SAAK,EAAU,KAAV,C;K;IC
TT,iC;K;;;;oDAyDI,0C;MAiB+D,oB;QAAA,2C;aAjB/D,kG;K;;IAoBJ,uC;MAAA,e;MAAA,iB;MAAA,uB;K;IAA
A,qC;MAAA,wC;O;MASI,4E;MAMA,8E;MAOA,4E;MAOA,kE;K;;IApBA,mD;MAAA,2B;MAAA,2C;K;;IAMA,
oD;MAAA,2B;MAAA,4C;K;;IAOA,mD;MAAA,2B;MAAA,2C;K;;IAOA,8C;MAAA,2B;MAAA,sC;K;;IA7BJ,iC;
MAAA,+K;K;;IAAA,sC;MAAA,a;aAAA,c;UAAA,gD;aAAA,e;UAAA,iD;aAAA,c;UAAA,gD;aAAA,S;UAAA,2C
;;UAAA,oE;;K;;oFAqCA,mB;K;;;;;;;;;;;;;;;;;;IhImBiD,gD;MAAA,oB;QACzC,WAAW,sBAAmB,YAAF,CAAE,C
AAnB,C;QACX,cAAM,IAAN,C;QADA,OAEA,IAAK,a;O;K;;;;IAtHb,+B;K;;iFAUA,yB;MAAA,4B;MAAA,mC;
QAMI,6BDgDQ,WChDkB,KDgDlB,CChDR,C;O;KANJ,C;2GAQA,yB;MAAA,4B;MDgDQ,kD;MChDR,uC;QAO
I,6BDgDQ,WAAO,cChDW,SDgDX,CAAP,CChDR,C;O;KAPJ,C;+FAUA,yB;MAAA,kC;MAAA,mD;MAAA,yE;
QASI,sC;QAAA,4C;O;MATJ,iGAWY,Y;QAAQ,2B;OAXpB,E;MAAA,0DAaQ,kB;QACI,wBAAW,MAAX,C;O;
MAdZ,sF;MAAA,sC;QASI,0D;O;KATJ,C;IAiBA,gD;MAaI,4BAA0D,YAAzC,wCAA6B,UAA7B,CAAyC,CAA1
D,EAAyE,yBAAzE,C;K;IAEJ,4D;MAcI,4BAAoE,YAAnD,0CAA6B,QAA7B,EAAuC,UAAvC,CAAmD,CAApE,
EAAmF,yBAAnF,C;K;IAEJ,+C;MAU6C,YAAzC,wCAA6B,UAA7B,CAAyC,CAtEzC,oBDgDQ,WCsBsD,kBDtB
tD,CChDR,C;K;IAyEJ,2D;MAWuD,YAAnD,0CAA6B,QAA7B,EAAuC,UAAvC,CAAmD,CApFnD,oBDgDQ,WC
oCgE,kBDpChE,CChDR,C;K;IAuFJ,+C;MAYI,OAA6C,8BAAtC,c;K;8EAZX,yB;MAAA,oE;MAAA,6E;MAYiD,
gD;QAAA,oB;UACzC,WAAW,sBAAmB,YAAF,CAAE,CAAnB,C;UACX,cAAM,IAAN,C;UADA,OAEA,IAAK,
a;S;O;MAfb,sC;QAYW,mBAAsC,8BAAtC,6B;QAAP,OAAO,kD;O;KAZX,C;qGA0BI,yB;MAAA,2D;MAAA,mB
;QACI,MAAM,6BAAoB,0BAApB,C;O;KADV,C;;MiIzIA,yC;;IAAA,uC;MAAA,2C;K;;;IAAA,mD;MAAA,kD;Q
AAA,iC;;MAAA,2C;K;+EAkBA,wB;K;oDAaA,e;MAK2C,IAAI,IAAJ,EAGK,M;MAL5C,IAAI,+CAAJ,C;QAEI,O
AAW,GAAI,kBAAS,IAAK,IAAd,CAAR,GAA4B,cAAI,OAAJ,GAAI,iBAAQ,IAAR,CAAJ,yCAA5B,GAAyD,I;;
MAGpE,OAAW,8CAA4B,GAAhC,GAAqC,8EAArC,GAAoD,I;K;yDAI/D,e;MAGI,IAAI,+CAAJ,C;QACI,OAA
W,GAAI,kBAAS,IAAK,IAAd,CAAJ,IAA0B,GAAI,iBAAQ,IAAR,CAAJ,QAA9B,GAAyD,mCAAzD,GAAoF,I;;
MAE/F,OAAW,8CAA4B,GAAhC,GAAqC,mCAArC,GAAgE,I;K;;;;ICtChD,oD;MACf,cAAc,GAAI,kBAAS,OAA
Q,IAAjB,C;MAClB,IAAI,YAAY,mCAAhB,C;QADA,OACuC,O;;QAEnC,kBAAkB,oBAAQ,yCAAR,C;QAClB,I
AAI,mBAAJ,C;UAJJ,OAI6B,oBAAgB,OAAhB,EAAyB,OAAzB,C;;UACrB,WAAW,OAAQ,kBAAS,yCAAT,C;U
AL3B,OAMY,SAAS,mCAAb,GAAoC,oBAAgB,OAAhB,EAAyB,WAAzB,CAApC,GACI,oBAAgB,oBAAgB,IA
AhB,EAAsB,OAAtB,CAAhB,EAAgD,WAAhD,C;;;K;8CAdxB,mB;MAKI,OAAI,YAAY,mCAAhB,GAAuC,IAAv
C,GACI,OAAQ,cAAK,IAAL,EAAW,4BAAX,C;K;;;;;;qDAiCZ,e;MAEyB,Q;MADrB,OACI,OAAA,IAAK,IAAL,
EAAY,GAAZ,CAAJ,GAAqB,0EAArB,GAAoC,I;K;sDAExC,8B;MACI,iBAAU,OAAV,EAAmB,IAAnB,C;K;0D
AEJ,e;MACI,OAAI,OAAA,IAAK,IAAL,EAAY,GAAZ,CAAJ,GAAqB,mCAArB,GAAgD,I;K;;;IC1DP,8C;MAAC
,wB;K;kFAAA,Y;MAAA,yB;K;;IAiCe,wD;MAEjE,kC;MAEA,4BAAqC,mDAAJ,GAAkD,OAAQ,qBAA1D,GAA
0E,O;K;4DAE3G,mB;MAA6C,+BAAS,OAAT,C;K;6DAC7C,e;MAA8C,eAAQ,IAAR,IAAgB,8BAAe,G;K;;IAGjF
,+C;MAW2C,IAAI,IAAJ,EAGV,M;MAL7B,IAAI,+CAAJ,C;QAEI,OAAW,GAAI,kBAAS,SAAK,IAAd,CAAR,G
AA4B,cAAI,OAAJ,GAAI,iBAAQ,SAAR,CAAJ,yCAA5B,GAAyD,I;;MAGpE,OAAW,SAAK,IAAL,KAAa,GAAj

B,GAAsB,mFAAtB,GAAqC,I;K;IAGhD,6C;MAUI,IAAI,+CAAJ,C;QACI,OAAW,GAAI,kBAAS,SAAK,IAAd,C
AAJ,IAA0B,GAAI,iBAAQ,SAAR,CAAJ,QAA9B,GAAyD,mCAAzD,GAAoF,S;;MAE/F,OAAW,SAAK,IAAL,K
AAa,GAAjB,GAAsB,mCAAtB,GAAiD,S;K;IAG5D,iC;MAAA,qC;MAKI,4B;K;oDACA,Y;MAAiC,0C;K;kDAEj
C,e;MAAyD,W;K;mDACzD,8B;MAA4E,c;K;mDAC5E,mB;MAAwE,c;K;uDACxE,e;MAA8D,W;K;+CAC9D,Y;
MAAsC,Q;K;+CACtC,Y;MAAyC,8B;K;;;IAb7C,6C;MAAA,4C;QAAA,2B;;MAAA,qC;K;IAqB8B,wC;MAC1B,k
B;MACA,wB;K;4CAGA,e;MAGQ,Q;MAFJ,UAAU,I;MACV,OAAO,IAAP,C;QACI,YAAA,GAAI,UAAJ,aAAY,
GAAZ,W;UAAwB,W;;QACxB,WAAW,GAAI,O;QACf,IAAI,oCAAJ,C;UACI,MAAM,I;;UAEN,OAAO,iBAAK,
GAAL,C;;;K;6CAKnB,8B;MACI,iBAAU,WAAK,cAAK,OAAL,EAAc,SAAd,CAAf,EAAyC,cAAzC,C;K;iDAEJ,e
;UAGW,I;MAFP,+BAAQ,GAAR,U;QAAoB,OAAO,W;;MAC3B,cAAc,WAAK,kBAAS,GAAT,C;MAEf,gBAAY,
WAAZ,C;QAAoB,W;WACpB,gBAAY,mCAAZ,C;QAAqC,qB;;QAC7B,2BAAgB,OAAhB,EAAyB,cAAzB,C;MA
HZ,W;K;uCAOJ,Y;MAIc,IAAI,IAAJ,Q;MAHV,UAAU,I;MACV,WAAW,C;MACX,OAAO,IAAP,C;QACU,uBA
AI,OAAJ,GAAI,OAAJ,gC;QAAA,mB;UAAgC,OAAO,I;;QAA7C,MAAM,M;QACN,mB;;K;2CAIR,mB;MACI,+
BAAI,OAAQ,IAAZ,GAAoB,OAApB,C;K;8CAEJ,mB;MAQ4B,Q;MAPxB,UAAU,O;MACV,OAAO,IAAP,C;QA
CI,IAAI,CAAC,gBAAS,GAAI,UAAb,CAAL,C;UAA4B,OAAO,K;QACnC,WAAW,GAAI,O;QACf,IAAI,oCAAJ,
C;UACI,MAAM,I;;UAEN,OAAO,gBAAS,0EAAT,C;;;K;uCAKnB,iB;MACI,gBAAS,KAAT,KAAkB,yCAA4B,K
AAM,SAAN,KAAgB,aAA5C,IAAsD,KAAM,eAAY,IAAZ,CAA9E,C;K;yCAEJ,Y;MAA+B,OAAK,SAAL,WAA
K,CAAL,GAA0B,SAAR,cAAQ,CAA1B,I;K;IAGZ,uD;MACX,OAAI,GzJyHoC,YAAU,CyJzHlD,GAAmB,OAAQ
,WAA3B,GAA6C,GAAF,UAAQ,O;K;yCAF3D,Y;MACI,aAAM,kBAAK,EAAL,EAAS,+BAAT,CAAN,GAEI,G;
K;IAMO,8E;MAAA,6B;QAAyB,Q;QAAT,iBAAS,sBAAT,EAAS,8BAAT,UAAoB,O;QAAQ,W;O;K;+CAJ3D,Y;
MAOsB,Q;MANlB,QAAQ,a;MACR,eAAe,gBAA+B,CAA/B,O;MACf,gBAAY,CAAZ,C;MACA,kBAAK,kBAAL
,EAAW,oDAAX,C;M5KtFJ,IAAI,E4KuFM,YAAS,C5KvFf,CAAJ,C;QACI,cAdW,e;QAeX,MAAM,6BAAsB,OA
AQ,WAA9B,C;;M4KuFN,OAAO,+BAAW,qDAAX,C;K;IAGa,8C;MACpB,kD;MADqB,wB;K;IACrB,gD;MAAAA,
oD;MACI,4B;K;;;IADJ,4D;MAAAA,2D;QAAAA,0C;;MAAAA,oD;K;yDAIA,Y;MAA0C,gBAAT,a;M3L8/YrB,Q;MA
DhB,kB2L7/YmD,mC;M3L8/YnD,wBAAgB,SAAhB,gB;QAAgB,cAAA,SAAhB,M;QAAsB,cAAwB,yBAAa,OA
Ab,C;;M2L9/YT,O3L+/Y9B,W;K;;;I4LjrZX,oE;MA4BI,MAAM,wBAAoB,sEAApB,C;K;8GA5BV,yB;MAAA,2D
;MAAA,sC;QA4BI,MAAM,6BAAoB,sEAApB,C;O;KA5BV,C;IA0CoC,mC;MAAQ,4D;K;IAE5C,4C;MAAA,e;M
AAA,iB;MAAA,uB;K;IAAA,0C;MAAA,6C;O;MAK0C,oG;MAAqB,gF;MAAW,4E;K;;IAAhC,+D;MAAA,gC;M
AAA,uD;K;;IAAqB,qD;MAAA,gC;MAAA,6C;K;;IAAW,mD;MAAA,gC;MAAA,2C;K;;IAL1E,sC;MAAA,sJ;K;;I
AAA,2C;MAAA,a;aAAA,qB;UAAA,4D;aAAA,W;UAAA,kD;aAAA,S;UAAA,gD;;UAAA,qF;;K;;;;;ICrCA,sC;M
AG0F,2BAAgB,eAAhB,C;K;IAKE,6C;MAAA,mB;QAAE,sB;O;K;IAH9F,gC;MAGwG,gBAA5B,oBAAgB,2BAA
hB,C;M3KoElE,S2K1DH,K;MAVqE,O3KqEjE,S;K;I2KzB+B,0C;MAAgE,uB;MAA/D,wC;MAMvC,kBACkC,I;K
;4FAE9B,Y;MACI,QAAQ,e;MACR,IAAI,SAAJ,C;QAAe,OAAO,C;MACtB,IAAI,wB;MACJ,kBAAW,C;MACX,
OAAO,C;K;uFAIX,Y;MAAQ,OAAA,cAAQ,O;K;4CAEpB,iB;MACI,cAAc,c;MACd,+DAAkB,KAAlB,EAAyB,O
AAQ,OAAjC,C;MACA,OAAO,QAAQ,KAAR,C;K;+CAMX,mB;MAEI,IAAI,YAAY,IAAhB,C;QAAsB,OAAO,K
;MAE7B,aAAqB,UAAR,cAAQ,EAAU,OAAQ,QAAlB,C;MACrB,OAAO,WAAW,O;K;8CAGtB,mB;MAEI,IAAI,
YAAY,IAAhB,C;QAAsB,OAAO,E;MAE7B,cAc,OAAQ,Q;MACtB,aAAqB,UAAR,cAAQ,EAAU,OAAV,C;MA
CrB,OAAW,WAAW,OAAf,GAAwB,OAAxB,GAAqC,E;K;kDAGhD,mB;MAA4C,0BAAQ,OAAR,C;K;+CAE5C,
Y;MAEI,OAAO,kCAA8B,cAA9B,C;K;;ICpHf,gC;K;;ICAA,sC;K;;6ECAA,yB;MAAA,0B;MAAA,mC;QAGsD,O
AAiC,OAA3B,SAAL,GAAAuB,KAAS,C;O;KAHvF,C;2EAKA,yB;MAAA,0B;MAAA,mC;QAGqD,OAAgC,OAA1
B,SAAL,GAAsB,KAAS,C;O;KAHrF,C;6EAKA,yB;MAAA,0B;MAAA,mC;QAGsD,OAAiC,OAA3B,SAAL,GAA
uB,KAAS,C;O;KAHvF,C;6EAKA,yB;MAAA,0B;MAAA,4B;QAGqC,OAAqB,OAAP,CAAR,SAAe,C;O;KAH1D,
C;+EAMA,yB;MAAA,4B;MAAA,mC;QAGyD,OAAiC,QAA3B,SAAL,GAAuB,KAAS,C;O;KAH1F,C;6EAKA,y
B;MAAA,4B;MAAA,mC;QAGwD,OAAgC,QAA1B,SAAL,GAAsB,KAAS,C;O;KAHxF,C;+EAKA,yB;MAAA,4
B;MAAA,mC;QAGyD,OAAiC,QAA3B,SAAL,GAAuB,KAAS,C;O;KAH1F,C;+EAKA,yB;MAAA,4B;MAAA,4B
;QAGuC,OAAqB,QAAP,CAAR,SAAe,C;O;KAH5D,C;ICpCA,qC;K;;ICAA,mB;K;;IAOA,iB;K;;IAOA,2C;K;;IAO
A,wB;K;;IAQA,0B;K;;IAOA,sB;K;;IAOA,4B;K;;IAOA,6C;K;;IA+BuC,wE;MAEnC,uB;QAAAA,UAAsB,E;MACt
B,qB;QAAAA,8B;MACA,2B;QAAAA,qE;MACA,yB;QAAAA,YAAAqB,E;MAJrB,sB;MACA,sB;MACA,kB;MACA,8B
;MACA,0B;K;;IAGJ,iD;MAAAA,e;MAAAA,iB;MAAAA,uB;K;IAAAA,+C;MAAAA,kD;O;MAKI,wG;MAACA,wG;MAC

A,8F;K;;IAFA,iE;MAAA,qC;MAAA,yD;K;;IACA,iE;MAAA,qC;MAAA,yD;K;;IACA,4D;MAAA,qC;MAAA,oD;
K;;IAPJ,2C;MAAA,6K;K;;IAAA,gD;MAAA,a;aAAA,kB;UAAA,8D;aAAA,kB;UAAA,8D;aAAA,a;UAAA,yD;;U
AAA,6E;;K;;IAUA,wB;K;;ICnGA,mB;MAEI,UAAU,IAAI,CAAJ,I;MACV,OAAW,OAAO,CAAX,GAAc,GAAd,
GAAuB,MAAM,CAAN,I;K;IAGlC,qB;MACI,UAAU,SAAI,CAAJ,C;MACV,OAAW,kBAAO,CAAX,GAAc,GAA
d,GAAuB,QAAM,CAAN,C;K;IAGlC,mC;MAEI,OAAO,IAAI,IAAI,CAAJ,EAAO,CAAP,IAAY,IAAI,CAAJ,EAA
O,CAAP,CAAZ,IAAJ,EAA2B,CAA3B,C;K;IAGX,qC;MACI,OAAO,MAAI,MAAI,CAAJ,EAAO,CAAP,WAAY,
MAAI,CAAJ,EAAO,CAAP,CAAZ,CAAJ,EAA2B,CAA3B,C;K;IAGX,qD;MAkBI,WAAO,CAAP,C;QAD2E,OAC
3D,SAAS,GAAb,GAAkB,GAAlB,GAA2B,MAAM,iBAAiB,GAAjB,EAAsB,KAAtB,EAA6B,IAA7B,CAAN,I;W
ACvC,WAAO,CAAP,C;QAF2E,OAE3D,SAAS,GAAb,GAAkB,GAAlB,GAA2B,MAAM,iBAAiB,KAAjB,EAAw
B,GAAxB,EAA6B,CAAC,IAAD,IAA7B,CAAN,I;;QAC/B,MAAa,gCAAyB,eAAzB,C;K;IAGzB,uD;MAkBI,sBA
AO,CAAP,C;QAD+E,OAC/D,sBAAS,GAAT,MAAJ,GAAkB,GAAlB,GAA2B,aAAM,mBAAiB,GAAjB,EAAsB,
KAAtB,EAA6B,IAA7B,CAAN,C;WACvC,sBAAO,CAAP,C;QAF+E,OAE/D,sBAAS,GAAT,MAAJ,GAAkB,GA
AlB,GAA2B,QAAM,mBAAiB,KAAjB,EAAwB,GAAxB,EAA8B,IAAD,aAA7B,CAAN,C;;QAC/B,MAAa,gCAA
yB,eAAzB,C;K;IC7DzB,qB;MAAA,yB;K;0CAII,Y;MAO6D,uB;K;2HAE7D,yB;MAAA,+D;MAAA,kC;MAAA,0
F;MAAA,6F;MAAA,4E;QAUI,wC;QAAS,2C;O;MAVb,mEAWQ,wC;QAA6E,sBAAS,QAAT,EAAmB,QAAnB,E
AA6B,QAA7B,C;O;MAXrF,oG;MAAA,yC;QAUI,wDAA+B,YAA/B,C;O;KAVJ,C;uHAcA,yB;MAAA,+D;MAA
A,kC;MAAA,wF;MAAA,yF;MAAA,0E;QAcI,wC;QAAS,2C;O;MAdb,kEAeQ,wC;QAAuF,6BAAS,QAAT,EAAm
B,QAAnB,EAA6B,QAA7B,C;O;MAf/F,kG;MAAA,yC;QAcI,sDAA+B,YAA/B,C;O;KAdJ,C;;;IA3BJ,iC;MAAA,g
C;QAAA,e;;MAAA,yB;K;IAgDiC,sB;MAC7B,eAAwB,I;K;4CAExB,6B;MACoB,IAAT,I;MAAA,mB;MAAA,iB;
QAAS,MAAM,6BAAsB,cAAY,QAAS,aAArB,uCAAtB,C;;MAAtB,OAAO,I;K;4CAGX,oC;MACI,eAAa,K;K;;;;;k
DC9CjB,6B;;K;;;;;;;;iEA+CA,6B;;K;;ICrDuC,0C;MACvC,uBAAoB,Y;K;wDAEpB,wC;MAM6F,W;K;uDAE7F,w
C;K;oDAMA,6B;MACI,OAAO,oB;K;oDAGX,oC;MACI,eAAe,IAAK,gB;MACpB,IAAI,CAAC,0BAAa,QAAb,E
AAuB,QAAvB,EAAiC,KAAjC,CAAL,C;QACI,M;;MAEJ,uBAAa,K;MACb,yBAAY,QAAZ,EAAsB,QAAtB,EAA
gC,KAAhC,C;K;;4EC9BR,wC;MAqBI,OAAO,e;K;4EAGX,+C;MAuBI,cAAI,KAAJ,C;K;4EAIJ,wC;MAmBI,OA
AO,cAAI,OAAJ,C;K;4EAGX,+C;MAqBI,cAAI,OAAJ,EAAa,KAAb,C;K;IC/FJ,kB;MA6PI,4B;K;+BAtOA,Y;MA
OiC,6BAAS,EAAT,C;K;uCAEjC,iB;MAW2C,4BAAQ,CAAR,EAAW,KAAX,C;K;uCAE3C,uB;MAakB,Q;MAHd
,iBAAiB,IAAjB,EAAuB,KAAvB,C;MACA,QAAQ,QAAQ,IAAR,I;MACR,IAAI,IAAI,CAAJ,IAAS,MAAK,WAA
lB,C;QACc,IAAI,MAAM,CAAC,CAAD,IAAN,OAAY,CAAhB,C;UACN,eAAe,SAAS,CAAT,C;UACf,6BAAS,Q
AAT,C;;UAEA,K;;YAEI,WAAW,cAAU,KAAK,C;YAC1B,IAAI,OAAO,CAAP,I;;UACC,gBAAO,CAAP,IAAY,C
AAZ,GAAgB,CAAhB,SAAqB,CAArB,C;UACT,Q;;QATJ,c;QAWA,OAAO,OAAO,GAAP,I;;QAEP,OAAO,IAAP
,C;UACI,YAAU,c;UACV,IAAW,IAAP,qBAAkB,KAAtB,C;YAA6B,OAAO,K;;;K;gCAKhD,Y;MAOmC,OAAU,o
BAAV,cAAU,CAAS,WAAI,EAAJ,CAAnB,yBAA6B,cAA7B,E;K;wCAEnC,iB;MAW8C,iCAAY,KAAZ,C;K;wC
AE9C,uB;MAiBkB,Q;MAPd,mBAAiB,IAAjB,EAAuB,KAAvB,C;MACA,QAAQ,eAAQ,IAAR,C;MACR,IAAI,eA
AI,CAAR,C;QACI,O;QACA,IAAI,aAAO,CAAD,aAAN,GAAY,CAAZ,CAAJ,C;UACI,WAAW,CAAE,Q;UACb,
YAAa,qBAAO,EAAP,CAAW,Q;UAEpB,aAAQ,CAAR,C;YACI,eAAe,SAAS,IAAT,C;YAEf,OAAAmB,oBAAnB,s
BAAS,QAAT,CAAmB,CAAnB,iB;iBAEJ,cAAS,CAAT,C;YAEI,OAAU,oBAAV,cAAU,CAAV,iB;;YAEA,iBAAe
,SAAS,KAAT,C;YACf,OAAmB,oBAAnB,sBAAS,UAAT,CAAmB,CAAS,WAAI,EAAJ,CAA5B,KAAiD,oBAAV
,cAAU,CAAV,iBAAvC,C;;UAXR,U;;UAeA,K;;YAEI,WAAW,eAAW,oBAAK,CAAL,C;YACtB,IAAI,YAAO,CA
AP,C;;UACC,sBAAO,CAAP,MAAY,+BAAI,CAAJ,EAAZ,eAAqB,CAArB,C;UACT,MAAM,C;;QAEV,OAAO,S
AAO,GAAP,C;;QAEP,OAAO,IAAP,C;UACI,YAAU,e;UACV,IAAW,IAAP,0CAAkB,KAAlB,CAAJ,C;YAA6B,O
AAO,K;;;K;mCAKhD,Y;MAKyC,6BAAS,CAAT,MAAe,C;K;kCAExD,Y;MAKuC,uBAAgB,sBAAS,EAAT,CAA
hB,EAA8B,sBAAS,EAAT,CAA9B,C;K;0CAEvC,iB;MASoD,+BAAW,GAAX,EAAgB,KAAhB,C;K;0CAEpD,uB;
MAcY,Q;MAFR,mBAAiB,IAAjB,EAAuB,KAAvB,C;MACA,WAAW,QAAQ,I;MACX,IAAS,WAAL,IAAK,CAA
L,IAA0B,SAAL,IAAK,CAA1B,IAA8C,SAAN,KAAM,CAAlD,C;QACJ,SAAS,qBAAgB,QAAQ,CAAR,GAAY,O
AAO,CAAnC,C;QACT,cAAO,EAAP,GAAY,E;;QAEZ,cAAO,oBAAe,I;;MAJ1B,Y;MAMA,OAAW,KAAK,KAA
T,GAAsB,SAAN,KAAM,CAAtB,GAAsC,C;K;iCAGjD,Y;MAKqC,6BAAS,EAAT,IAA0B,Q;K;IAWK,oF;MAAA
,mB;QAAE,uBAAa,iBAAb,sBAAqC,eAArC,+BAAqE,aAAM,OAA3E,M;O;K;iDATtE,qC;MzLjLA,IAAI,EyL0Lq
B,CAAb,8BAAgB,KAAM,OzL1L9B,GyL0LiD,CAAX,0BAAc,KAAM,OzL1L1D,GyL0LsC,KzL1LtC,CAAJ,C;Q

ACI,cyLyLgE,kDzLzLlD,E;QACd,MAAM,gCAAyB,OAAQ,WAAjC,C;;MAFV,IAAI,EyL2LQ,aAAa,OzL3LrB,C
AAJ,C;QACI,gByL0LgC,mF;QzLzLhC,MAAM,gCAAyB,SAAQ,WAAjC,C;;MyL2LN,YAAY,CAAC,UAAU,SA
AV,IAAD,IAAwB,CAAxB,I;MAEZ,mBAAe,SAAf,C;MtLzEJ,iBAAc,CAAd,UsL0EW,KtL1EX,U;QsL2EQ,QAA
Q,c;QACR,MAAM,UAAN,IAAoB,OAAF,CAAE,C;QACpB,MAAM,aAAW,CAAX,IAAN,IAAgC,OAAV,CAAE,
KAAK,CAAG,C;QAChC,MAAM,aAAW,CAAX,IAAN,IAAiC,OAAX,CAAE,KAAK,EAAI,C;QACjC,MAAM,a
AAW,CAAX,IAAN,IAAiC,OAAX,CAAE,KAAK,EAAI,C;QACjC,0BAAY,CAAZ,I;;MAGJ,gBAAgB,UAAU,UA
AV,I;MAChB,SAAS,sBAAS,YAAY,CAAZ,IAAT,C;MACT,aAAU,CAAV,MAAkB,SAAlB,M;QACI,MAAM,aA
AW,CAAX,IAAN,IAAqC,OAAf,EAAG,MAAK,IAAI,CAAJ,IAAL,CAAY,C;;MAGzC,OAAO,K;K;yCACX,uD;
MAvB4C,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,KAAM,O;aARrF,0H;K;yCAiCA,iB;MAOyD,8BAAU,K
AAV,EAAiB,CAAjB,EAAoB,KAAM,OAA1B,C;K;yCAEzD,gB;MAKkD,8BAAU,cAAU,IAAV,CAAV,C;K;IAGl
D,0B;MAAA,8B;MAO2B,iB;MACvB,uBAAoC,uB;K;IAEpC,qC;MAAA,yC;MACI,4B;K;wDAEA,Y;MAAiC,mC
;K;;;IAHrC,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;8CAMA,Y;MAAkC,8C;K;gDAElC,oB;MAA4C,OAAA,oBA
Ac,kBAAS,QAAT,C;K;uCAC1D,Y;MAA8B,OAAA,oBAAc,U;K;+CAC5C,iB;MAAwC,OAAA,oBAAc,iBAAQ,K
AAR,C;K;+CACtD,uB;MAAmD,OAAA,oBAAc,iBAAQ,IAAR,EAAc,KAAd,C;K;wCAEjE,Y;MAAgC,OAAA,oB
AAc,W;K;gDAC9C,iB;MAA2C,OAAA,oBAAc,kBAAS,KAAT,C;K;gDACzD,uB;MAAuD,OAAA,oBAAc,kBAA
S,IAAT,EAAe,KAAf,C;K;2CAErE,Y;MAAsC,OAAA,oBAAc,c;K;0CAEpD,Y;MAAoC,OAAA,oBAAc,a;K;kDA
ClD,iB;MAAiD,OAAA,oBAAc,oBAAW,KAAX,C;K;kDAC/D,uB;MAA+D,OAAA,oBAAc,oBAAW,IAAX,EAAi
B,KAAjB,C;K;yCAE7E,Y;MAAkC,OAAA,oBAAc,Y;K;iDAEhD,iB;MAAsD,OAAA,oBAAc,mBAAU,KAAV,C;
K;iDACpE,gB;MAA+C,OAAA,oBAAc,mBAAU,IAAV,C;K;yDAC7D,qC;MACI,OAAA,oBAAc,mBAAU,KAAV
,EAAiB,SAAjB,EAA4B,OAA5B,C;K;;;IAtCtB,sC;MAAA,qC;QAAA,oB;;MAAA,8B;K;;IA0CJ,wB;MAauC,yBA
Aa,IAAb,EAAmB,IAAK,IAAI,EAA5B,C;K;IAEvC,wB;MAawC,yBAAa,IAAK,QAAlB,EAA2B,IAAK,YAAI,EA
AJ,CAAQ,QAAxC,C;K;IAGxC,mC;MAUI,IAAA,KAAM,UAAN,C;QAAmB,MAAM,gCAAyB,uCAAoC,KAA7D
,C;WACzB,IAAA,KAAM,KAAN,GAAa,UAAb,C;QAF8C,OAEhB,0BAAQ,KAAM,MAAd,EAAqB,KAAM,KAA
N,GAAa,CAAb,IAArB,C;WAC9B,IAAA,KAAM,MAAN,GAAc,WAAd,C;QAH8C,OAGf,0BAAQ,KAAM,MAA
N,GAAc,CAAd,IAAR,EAAyB,KAAM,KAA/B,IAAuC,CAAvC,I;;QAHe,OAItC,mB;K;IAGZ,oC;MAUI,IAAA,K
AAM,UAAN,C;QAAmB,MAAM,gCAAyB,uCAAoC,KAA7D,C;WACzB,IAAA,KAAM,KAAN,+C;QAFiD,OAEl
B,2BAAS,KAAM,MAAf,EAAsB,KAAM,KAAN,yBAAa,CAAb,EAAAtB,C;WAC/B,IAAA,KAAM,MAAN,+C;QA
HiD,OAGjB,2BAAS,KAAM,MAAN,8BAAc,CAAd,EAAT,EAA0B,KAAM,KAAhC,0BAAwC,CAAxC,E;;QAHi
B,OAIzC,oB;K;IAOZ,yB;MAAyC,YrFrTkB,YqFqTb,KrFrTa,CqFqTlB,I;K;IAEzC,4C;MAEI,OAAA,SAAK,KAA
K,EAAL,GAAU,QAAf,GAAyC,CAAX,CAAC,QAAD,IAAW,KAAI,E;K;IAEjD,uC;MzLtVI,IAAI,EyLsVuD,QA
AQ,IzLtV/D,CAAJ,C;QACI,cyLqVuE,+B;QzLpVvE,MAAM,gCAAyB,OAAQ,WAAjC,C;;K;IyLqVd,yC;MzLvVI,
IAAI,EyLuVyD,sBAAQ,IAAR,KzLvVzD,CAAJ,C;QACI,cyLsVyE,+B;QzLrVzE,MAAM,gCAAyB,OAAQ,WAAj
C,C;;K;IyLsVd,yC;MzLxVI,IAAI,EyLwV6D,QAAQ,IzLxVrE,CAAJ,C;QACI,cyLuV6E,+B;QzLtV7E,MAAM,gC
AAyB,OAAQ,WAAjC,C;;K;IyLwVd,yC;MAAyD,oCAA0B,IAA1B,qBAAiC,KAAjC,kB;K;ICrXzD,6B;MAOqC,
OtM6YE,SsM7YF,mBtM6YE,C;K;IsM3YvC,sC;MASgD,6BAAS,WAAT,EAAa,KAAb,C;K;IAEhD,4C;MAUI,qB
AAqB,IAArB,EAA2B,KAA3B,C;MAEA,iBAAiB,ItM+QgB,KsM/QhB,GAAiB,W;MAClC,kBAAkB,KtM8Qe,Ks
M9Qf,GAAkB,W;MAEpC,mBAAmB,0BAAQ,UAAR,EAAoB,WAApB,IAAqC,W;MACxD,OtMgXmC,SsMhX5
B,YtMgX4B,C;K;IsM7WvC,sC;MAWI,IAAA,KAAM,UAAN,C;QAAmB,MAAM,gCAAyB,uCAAoC,KAA7D,C;;
QACzB,ItMGkE,YsMHlE,KAAM,KtMG6E,KAAjB,EsMHrD,4BAAK,UtMG6E,KAA7B,CsMHlE,K;UAFiD,OAE
lB,sBAAS,KAAM,MAAf,EtMqBsB,SsMrBA,KAAM,KtMqBI,KAAK,GAAW,CsMrBb,WtMqBa,MAAX,IAAf,Cs
MrBtB,C;;UAC/B,ItMEkE,YsMFlE,KAAM,MtME6E,KAAjB,EsMFpD,4BAAK,UtME4E,KAA7B,CsMFlE,K;YA
HiD,OtMuBI,SsMpBrB,sBtMiCsB,SsMjCb,KAAM,MtMiCiB,KAAK,GAAY,CsMjC1B,WtMiC0B,MAAZ,IAAf,C
sMjCtB,EAA2B,KAAM,KAAjC,CtMoB+B,KAAK,GAAW,CsMpBN,WtMoBM,MAAX,IAAf,C;;YsMvBJ,OAIzC
,mB;;;K;IAGZ,8B;MAOuC,OrLoWG,UqLpWH,oBrLoWG,C;K;IqLlW1C,uC;MASmD,8BAAU,2BAAV,EAAe,K
AAf,C;K;IAEnD,6C;MAUI,sBAAsB,IAAtB,EAA4B,KAA5B,C;MAEA,iBAAiB,IrLkOkB,KqLlOlB,8B;MACjB,k
BAAkB,KrLiOiB,KqLjOjB,8B;MAElB,mBAAmB,2BAAS,UAAT,EAAqB,WAArB,+B;MACnB,OrLuUsC,UqLv
U/B,YrLuU+B,C;K;IqLpU1C,uC;MAWI,IAAA,KAAM,UAAN,C;QAAmB,MAAM,gCAAyB,uCAAoC,KAA7D,C
;;QACzB,IrL7CmE,aqL6CnE,KAAM,KrL7C+E,KAAlB,EqL6CtD,6BAAM,UrL7C8E,KAA9B,CqL6CnE,K;UAFo

D,OAEpB,uBAAU,KAAM,MAAhB,ErLhCuB,UqLgCA,KAAM,KrLhCK,KAAK,KAAW,CjBgR7C,UAAW,oBA
AL,CsMhPyB,WtMgPzB,MAAK,CAAL,iBAAN,CiBhR6C,MAAX,CAAhB,CqLgCvB,C;;UAChC,IrL9CmE,aqL8
CnE,KAAM,MrL9C+E,KAAlB,EqL8CrD,6BAAM,UrL9C6E,KAA9B,CqL8CnE,K;YAHoD,OrL9BG,UqLiCtB,uB
rLpBuB,UqLoBb,KAAM,MrLpBkB,KAAK,UAAY,CjBmQ/C,UAAW,oBAAL,CsM/Oc,WtM+Od,MAAK,CAAL,i
BAAN,CiBnQ+C,MAAZ,CAAhB,CqLoBvB,EAA4B,KAAM,KAAlC,CrLjCiC,KAAK,KAAW,CjBgR7C,UAAW,
oBAAL,CsM/OgC,WtM+OhC,MAAK,CAAL,iBAAN,CiBhR6C,MAAX,CAAhB,C;;YqL8BH,OAI5C,oB;;;K;IAG
Z,sC;MAQI,4BAAU,KlKg/FH,QkKh/FP,C;MACA,OAAO,K;K;IAGX,uC;MAKsD,OlK+iG3C,ekK/iG2C,4BAAU,
IAAV,ClK+iG3C,C;K;IkK7iGX,4D;MAOgD,yB;QAAA,YAAiB,C;MAAG,uB;QAAA,UAAe,KAAM,K;MACrF,4
BAAU,KlK69FH,QkK79FP,EAA+B,SAA/B,EAA0C,OAA1C,C;MACA,OAAO,K;K;IAIX,2C;M1LrHI,IAAI,EZ2
B8D,YsM0FD,KtM1FkB,KAAjB,EsM0FO,ItM1FsB,KAA7B,CsM0FD,I1LrH7D,CAAJ,C;QACI,c0LoH6E,+B;Q1
LnH7E,MAAM,gCAAyB,OAAAQ,WAAjC,C;;K;I0LoHd,4C;M1LtHI,IAAI,EKmC+D,aqLmFC,KrLnFiB,KAAlB,E
qLmFS,IrLnFqB,KAA9B,CqLmFC,I1LtHhE,CAAJ,C;QACI,c0LqHgF,+B;Q1LpHhF,MAAM,gCAAyB,OAAQ,W
AAjC,C;;K;I2LpBc,6C;MAsCxB,oC;MA/BA,iB;MANA,Y;MACA,Y;MACA,Y;MACA,Y;MACA,Y;MACA,sB;M
3LYA,IAAI,E2LLQ,CAAC,WAAK,QAAL,GAAU,QAAV,GAAe,QAAf,GAAoB,QAArB,MAA2B,C3LKnC,CAA
J,C;QACI,c2LNwC,wD;Q3LOxC,MAAM,gCAAyB,OAAQ,WAAjC,C;;MGoHV,iBAAc,CAAd,UwLxHW,ExLwH
X,U;QwLxHiB,c;;K;qCAGjB,Y;MAGI,QAAQ,Q;MACR,IAAI,IAAO,MAAO,C;MAClB,WAAI,Q;MACJ,WAAI,
Q;MACJ,WAAI,Q;MACJ,SAAS,Q;MACT,WAAI,E;MACJ,IAAK,IAAO,KAAM,CAAd,GAAsB,EAAtB,GAA8B,
MAAO,C;MACzC,WAAI,C;MACJ,gCAAU,MAAV,I;MACA,OAAO,IAAI,aAAJ,I;K;8CAGX,oB;MACI,OAAU,c
AAV,cAAU,EAAc,QAAd,C;K;IAEd,kC;MAAA,sC;MACI,4B;K;;;IADJ,8C;MAAA,6C;QAAA,4B;;MAAA,sC;K;;
IA7BA,gD;MAAA,sD;MACQ,yBAAK,KAAL,EAAY,KAAZ,EAAmB,CAAnB,EAAsB,CAAtB,EAA+B,CAAN,K
AAzB,EAAuC,SAAU,EAAX,GAAoB,UAAW,CAArE,C;MADR,Y;K;IlMbkB,wC;MA8BlB,iC;MA9BsD,2BAAg
B,KAAhB,EAAuB,YAAvB,EAAqC,CAArC,C;K;kFAC7B,Y;MAAQ,8B;K;yFACD,Y;MAAQ,6B;K;yFAKR,Y;M
AC5B,IAAI,cAAQ,sCAAK,UAAjB,C;QOyHyC,MAAM,6BPzHb,6EOyH2C,WAA9B,C;;MPxH/C,OAAO,+BAA
O,CAAP,E;K;2CAGX,iB;MAA8C,qBAAS,KAAT,IAAkB,SAAS,S;K;kCAEzE,Y;MAKkC,oBAAQ,S;K;iCAE1C,i
B;MACI,2CAAuB,kBAAa,KAAM,UAAnB,KACvB,eAAS,KAAM,MAAf,IAAwB,cAAQ,KAAM,KADf,CAAvB,
C;K;mCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,OAAK,UsBUS,ItBVd,UAAkB,SsBUJ,ItBVd,K;K;mCA
E5B,Y;MAAkC,2BAAE,UAAF,+BAAU,SAAV,C;K;IAElC,+B;MAAA,mC;MACI,aAC8B,cAAY,OAAF,CAAE,C
AAZ,EAAwB,OAAF,CAAE,CAAxB,C;K;;;IAFlC,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;;IAUiB,uC;MA8BjB,
gC;MA9BmD,0BAAe,KAAf,EAAsB,YAAtB,EAAoC,CAApC,C;K;iFAC3B,Y;MAAQ,iB;K;wFACD,Y;MAAQ,g
B;K;wFAKR,Y;MAC3B,IAAI,cAAQ,UAAZ,C;QOiFyC,MAAM,6BPjFd,6EOiF4C,WAA9B,C;;MPhF/C,OAAO,Y
AAO,CAAP,I;K;0CAGX,iB;MAA6C,qBAAS,KAAT,IAAkB,SAAS,S;K;iCAExE,Y;MAKkC,oBAAQ,S;K;gCAE1
C,iB;MACI,0CAAsB,kBAAa,KAAM,UAAnB,KACtB,eAAS,KAAM,MAAf,IAAwB,cAAQ,KAAM,KADhB,CAA
tB,C;K;kCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,MAAK,UAAL,QAAa,SAAb,I;K;kCAE5B,Y;MAAk
C,OAAE,UAAF,qBAAU,S;K;IAE5C,8B;MAAA,kC;MACI,aAC6B,aAAS,CAAT,EAAY,CAAZ,C;K;;;IAFjC,0C;
MAAA,yC;QAAA,wB;;MAAA,kC;K;;IAUkB,wC;MA8BlB,iC;MA9BsD,2BAAgB,KAAhB,EAAuB,YAAvB,K;K;
kFAC7B,Y;MAAQ,iB;K;yFACD,Y;MAAQ,gB;K;yFAKR,Y;MAC5B,IAAI,2CAAJ,C;QOyCyC,MAAM,6BPzCb,
6EOyC2C,WAA9B,C;;MPxC/C,OAAO,kCAAO,CAAP,E;K;2CAGX,iB;MAA8C,kCAAS,KAAT,UAAkB,sBAAS,
SAAT,M;K;kCAEhE,Y;MAKkC,kCAAQ,SAAR,K;K;iCAElC,iB;MACI,2CAAuB,kBAAa,KAAM,UAAnB,KACv
B,mBAAS,KAAM,MAAf,KAAwB,kBAAQ,KAAM,KAAd,CADD,CAAvB,C;K;mCAGJ,Y;MACI,OAAI,cAAJ,G
AAe,EAAf,GAAwB,iCAAM,eAAW,8BAAW,EAAX,CAAX,CAAN,MAAoC,cAAU,6BAAU,EAAV,CAAV,CAA
pC,CAA8D,Q;K;mCAE1F,Y;MAAkC,OAAE,UAAF,qBAAU,SAAV,W;K;IAElC,+B;MAAA,mC;MACI,aAC8B,q
B;K;;;IAFlC,2C;MAAA,0C;QAAA,yB;;MAAA,mC;K;;ImM9GkC,oD;MAA2C,uB;MAAjB,gB;MAC5D,sBAAgC,
I7KmCU,I;M6KlC1C,iBAAmC,YAAO,CAAX,GAAc,SAAS,IAAvB,GAAiC,SAAS,I;MACzE,cAA4B,cAA5B,GA
AqC,K7KiCK,I6KjC1C,GAAqD,mB;K;gDAErD,Y;MAAkC,qB;K;iDAElC,Y;MACI,YAAY,W;MACZ,IAAI,UAA
S,mBAAb,C;QACI,IAAI,CAAC,cAAL,C;UAAc,MAAa,6B;QAC3B,iBAAU,K;;QAGV,4BAAQ,SAAR,I;;MAEJ,O
AAa,OAAN,KAAM,C;K;;IAQgB,mD;MAAyC,sB;MAAjB,gB;MACzD,sBAAgC,I;MAChC,iBAAmC,YAAO,CA
AX,GAAc,SAAS,IAAvB,GAAiC,SAAS,I;MACzE,cAA4B,cAAJ,GAAa,KAAb,GAAwB,mB;K;+CAEhD,Y;MAA
kC,qB;K;+CAElC,Y;MACI,YAAY,W;MACZ,IAAI,UAAS,mBAAb,C;QACI,IAAI,CAAC,cAAL,C;UAAc,MAAa,

6B;QAC3B,iBAAU,K;;QAGV,4BAAQ,SAAR,I;;MAEJ,OAAO,K;K;;IAQuB,oD;MAA4C,uB;MAAlB,gB;MAC5D
,sBAAiC,I;MACjC,iBAAmC,uBAAO,CAAX,GAAc,sBAAS,IAAT,MAAd,GAAiC,sBAAS,IAAT,M;MAChE,cAA
6B,cAAJ,GAAa,KAAb,GAAwB,mB;K;gDAEjD,Y;MAAkC,qB;K;iDAElC,Y;MACI,YAAY,W;MACZ,IAAI,cAA
S,mBAAT,CAAJ,C;QACI,IAAI,CAAC,cAAL,C;UAAc,MAAa,6B;QAC3B,iBAAU,K;;QAGV,8BAAQ,SAAR,C;;
MAEJ,OAAO,K;K;;IC9DX,oD;MA6CA,uC;MAtCI,IAAI,SAAQ,CAAZ,C;QAAe,MAAa,gCAAyB,wBAAzB,C;M
AC5B,IAAI,SAAQ,WAAZ,C;QAA2B,MAAa,gCAAyB,wEAAzB,C;MAG5C,aAGyB,K;MAEzB,YAGuF,OAA/D,
0BAA0B,K9KeR,I8KflB,EAAsC,Y9KepB,I8KflB,EAAyD,IAAzD,CAA+D,C;MAEvF,YAGuB,I;K;yCAEvB,Y;M
AAwC,mCAAwB,UAAxB,EAA+B,SAA/B,EAAqC,SAArC,C;K;wCAExC,Y;MAMqC,OAAI,YAAO,CAAX,GAA
c,aAAQ,SAAtB,GAAgC,aAAQ,S;K;uCAE7E,iB;MACI,iDAA6B,kBAAa,KAAM,UAAnB,KAC7B,eAAS,KAAM,
MAAf,IAAwB,cAAQ,KAAM,KAAtC,IAA8C,cAAQ,KAAM,KAD/B,CAA7B,C;K;yCAGJ,Y;MACI,OAAI,cAAJ,
GAAe,EAAf,GAAwB,OAAM,OAAK,U9KPG,I8KOR,UAAkB,S9KPV,I8KOR,KAAN,SAAqC,SAArC,I;K;yCAE
5B,Y;MAAkC,OAAI,YAAO,CAAX,GAAc,oBAAE,UAAF,+BAAU,SAAV,eAAqB,SAAnC,GAA8C,oBAAE,UA
AF,qCAAgB,SAAhB,gBAA4B,CAAC,SAAD,IAA5B,C;K;IAEhF,qC;MAAA,yC;K;kEACI,sC;MAQ2F,2BAAgB,
UAAhB,EAA4B,QAA5B,EAAsC,IAAtC,C;K;;;IAT/F,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;;IAiBA,mD;MA6C
A,sC;MAtCI,IAAI,SAAQ,CAAZ,C;QAAe,MAAa,gCAAyB,wBAAzB,C;MAC5B,IAAI,SAAQ,WAAZ,C;QAA2B,
MAAa,gCAAyB,wEAAzB,C;MAG5C,aAGwB,K;MAExB,YAGuB,0BAA0B,KAA1B,EAAiC,YAAjC,EAA+C,IA
A/C,C;MAEvB,YAGuB,I;K;wCAEvB,Y;MAAuC,kCAAuB,UAAvB,EAA8B,SAA9B,EAAoC,SAApC,C;K;uCAE
vC,Y;MAMqC,OAAI,YAAO,CAAX,GAAc,aAAQ,SAAtB,GAAgC,aAAQ,S;K;sCAE7E,iB;MACI,gDAA4B,kBA
Aa,KAAM,UAAnB,KAC5B,eAAS,KAAM,MAAf,IAAwB,cAAQ,KAAM,KAAtC,IAA8C,cAAQ,KAAM,KADhC,
CAA5B,C;K;wCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,OAAM,MAAK,UAAL,QAAa,SAAb,IAAN,SA
A2B,SAA3B,I;K;wCAE5B,Y;MAAkC,OAAI,YAAO,CAAX,GAAAgB,UAAF,qBAAU,SAAV,cAAqB,SAAnC,GA
AgD,UAAF,2BAAgB,SAAhB,eAA4B,CAAC,SAAD,IAA5B,C;K;IAEhF,oC;MAAA,wC;K;iEACI,sC;MAQwF,0B
AAe,UAAf,EAA2B,QAA3B,EAAqC,IAArC,C;K;;;IAT5F,gD;MAAA,+C;QAAA,8B;;MAAA,wC;K;;IAiBA,oD;M
A6CA,uC;MAtCI,IAAI,gBAAJ,C;QAAgB,MAAa,gCAAyB,wBAAzB,C;MAC7B,IAAI,sCAAJ,C;QAA4B,MAAa,
gCAAyB,yEAAzB,C;MAG7C,aAGyB,K;MAEzB,YAGwB,4BAA0B,KAA1B,EAAiC,YAAjC,EAA+C,IAA/C,C;
MAExB,YAGwB,I;K;yCAExB,Y;MAAwC,mCAAwB,UAAxB,EAA+B,SAA/B,EAAqC,SAArC,C;K;wCAExC,Y;
MAMqC,OAAI,uBAAO,CAAX,GAAc,2BAAQ,SAAR,KAAd,GAAgC,2BAAQ,SAAR,K;K;uCAErE,iB;MACI,iD
AA6B,kBAAa,KAAM,UAAnB,KAC7B,mBAAS,KAAM,MAAf,KAAwB,kBAAQ,KAAM,KAAd,CAAxB,IAA8C
,kBAAQ,KAAM,KAAd,CADjB,CAA7B,C;K;yCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,iCAAM,iCAA
M,eAAW,8BAAW,EAAX,CAAX,CAAN,MAAoC,cAAU,6BAAU,EAAV,CAAV,CAApC,CAAN,MAAuE,cAAU,
6BAAU,EAAV,CAAV,CAAvE,CAAiG,Q;K;yCAE7H,Y;MAAkC,OAAI,uBAAO,CAAX,GAAgB,UAAF,qBAAU
,SAAV,yBAAqB,SAArB,WAAd,GAAgD,UAAF,2BAAgB,SAAhB,yBAA6B,SAAD,aAA5B,W;K;IAEhF,qC;MA
AA,yC;K;kEACI,sC;MAQ4F,2BAAgB,UAAhB,EAA4B,QAA5B,EAAsC,IAAtC,C;K;;;IAThG,iD;MAAA,gD;QA
AA,+B;;MAAA,yC;K;;;;6CClKA,iB;MAKkD,+BAAS,UAAT,UAAkB,wBAAS,iBAAT,M;K;oCAEpE,Y;MAKgC,
oCAAQ,iBAAR,K;K;;;;8CAuBhC,iB;MAKkD,+BAAS,UAAT,UAAkB,wBAAQ,iBAAR,K;K;qCAEpE,Y;MAKgC
,oCAAS,iBAAT,M;K;;ICxDiB,8C;MACjD,4B;MACA,0C;K;oEADA,Y;MAAA,2B;K;2EACA,Y;MAAA,kC;K;uC
AGA,iB;MACI,OAAO,0CAAgC,kBAAa,KAAM,UAAnB,KAC/B,mBAAS,KAAM,MAAf,KAAwB,0BAAgB,KA
AM,aAAtB,CADO,CAAhC,C;K;yCAIX,Y;MACI,OAAW,cAAJ,GAAe,EAAf,GAAuB,MAAW,SAAN,UAAM,CA
AX,QAAqC,SAAb,iBAAa,CAArC,I;K;yCAGlC,Y;MAAkC,OAAE,UAAF,qBAAU,iB;K;;IAGhD,kC;MAM6E,2B
AAgB,SAAhB,EAAsB,IAAtB,C;K;IAMjB,qD;MACxD,4B;MACA,0C;K;2EADA,Y;MAAA,2B;K;kFACA,Y;MA
AA,kC;K;8CAGA,iB;MACI,OAAO,iDAAuC,kBAAa,KAAM,UAAnB,KACtC,mBAAS,KAAM,MAAf,KAAwB,0
BAAgB,KAAM,aAAtB,CADc,CAAvC,C;K;gDAIX,Y;MACI,OAAW,cAAJ,GAAe,EAAf,GAAuB,MAAW,SAAN,
UAAM,CAAX,QAAqC,SAAb,iBAAa,CAArC,I;K;gDAGlC,Y;MAAkC,OAAE,UAAF,sBAAW,iB;K;;IAGjD,qC;
MAQiF,kCAAuB,SAAvB,EAA6B,IAA7B,C;K;;;0DAY7E,iB;MAA2C,qCAAiB,UAAjB,EAAwB,KAAxB,KAAk
C,8BAAiB,KAAjB,EAAwB,iBAAxB,C;K;iDAC7E,Y;MAAkC,QAAC,8BAAiB,UAAjB,EAAwB,iBAAxB,C;K;;I
AcR,gD;MAI3B,gBAAqB,K;MACrB,uBAA4B,Y;K;0FACD,Y;MAAQ,oB;K;iGACD,Y;MAAQ,2B;K;2DAE1C,g
B;MAA+D,YAAK,C;K;mDAEpE,iB;MAAgD,gBAAS,aAAT,IAAmB,SAAS,oB;K;0CAC5E,Y;MAAkC,SAAE,iB
AAU,oBAAZ,C;K;yCAElC,iB;MACI,OAAO,4CAA+B,kBAAa,KAAM,UAAnB,KAC9B,kBAAU,KAAM,SAAhB

,IAA0B,yBAAiB,KAAM,gBADnB,CAA/B,C;K;2CAIX,Y;MACI,OAAW,cAAJ,GAAe,EAAf,GAAuB,MAAY,SA
AP,aAAO,CAAZ,QAAuC,SAAd,oBAAc,CAAvC,I;K;2CAGlC,Y;MAAkC,OAAE,aAAF,qBAAW,oB;K;;IAGjD,o
C;MAOqF,6BAAkB,SAAlB,EAAwB,IAAxB,C;K;IAQrD,iD;MAI5B,gBAAqB,K;MACrB,uBAA4B,Y;K;2FACD,
Y;MAAQ,oB;K;kGACD,Y;MAAQ,2B;K;sDAE1C,gB;MAA8D,YAAK,C;K;oDAEnE,iB;MAAgD,gBAAS,aAAT,I
AAmB,QAAQ,oB;K;2CAC3E,Y;MAAkC,SAAE,gBAAS,oBAAX,C;K;0CAElC,iB;MACI,OAAO,6CAAgC,kBAA
a,KAAM,UAAnB,KAC/B,kBAAU,KAAM,SAAhB,IAA0B,yBAAiB,KAAM,gBADlB,CAAhC,C;K;4CAIX,Y;MA
CI,OAAW,cAAJ,GAAe,EAAf,GAAuB,MAAY,SAAP,aAAO,CAAZ,QAAuC,SAAd,oBAAc,CAAvC,I;K;4CAGlC,
Y;MAAkC,OAAE,aAAF,sBAAY,oB;K;;IAGlD,uC;MAO4E,8BAAmB,SAAnB,EAAyB,IAAzB,C;K;IAQ9C,+C;M
AI1B,gBAAqB,K;MACrB,uBAA4B,Y;K;yFACF,Y;MAAQ,oB;K;gGACD,Y;MAAQ,2B;K;0DAEzC,gB;MAA6D,
YAAK,C;K;kDAElE,iB;MAA+C,gBAAS,aAAT,IAAmB,SAAS,oB;K;yCAC3E,Y;MAAkC,SAAE,iBAAU,oBAA
Z,C;K;wCAElC,iB;MACI,OAAO,2CAA8B,kBAAa,KAAM,UAAnB,KAC7B,kBAAU,KAAM,SAAhB,IAA0B,yB
AAiB,KAAM,gBADpB,CAA9B,C;K;0CAIX,Y;MACI,OAAW,cAAJ,GAAe,EAAf,GAAuB,MAAY,SAAP,aAAO,
CAAZ,QAAuC,SAAd,oBAAc,CAAvC,I;K;0CAGlC,Y;MAAkC,OAAE,aAAF,qBAAW,oB;K;;IAGjD,oC;MAOkF,
4BAAiB,SAAjB,EAAuB,IAAvB,C;K;IASnD,gD;MAI3B,gBAAqB,K;MACrB,uBAA4B,Y;K;0FACF,Y;MAAQ,o
B;K;iGACD,Y;MAAQ,2B;K;qDAEzC,gB;MAA4D,YAAK,C;K;mDAEjE,iB;MAA+C,gBAAS,aAAT,IAAmB,QA
AQ,oB;K;0CAC1E,Y;MAAkC,SAAE,gBAAS,oBAAX,C;K;yCAElC,iB;MACI,OAAO,4CAA+B,kBAAa,KAAM,
UAAnB,KAC9B,kBAAU,KAAM,SAAhB,IAA0B,yBAAiB,KAAM,gBADnB,CAA/B,C;K;2CAIX,Y;MACI,OAA
W,cAAJ,GAAe,EAAf,GAAuB,MAAY,SAAP,aAAO,CAAZ,QAAuC,SAAd,oBAAc,CAAvC,I;K;2CAGlC,Y;MAA
kC,OAAE,aAAF,sBAAY,oB;K;;IAGlD,uC;MAOyE,6BAAkB,SAAlB,EAAwB,IAAxB,C;K;oFAGzE,8B;MAQI,0
BAAmB,2BAAS,OAAT,C;K;oFAEvB,8B;MASI,0BAAmB,2BAAS,OAAT,C;K;IAEvB,+C;MACI,IAAI,CAAC,U
AAL,C;QAAiB,MAAM,gCAAyB,iCAA8B,IAA9B,iBAAzB,C;K;ICxQ3B,gC;MAcW,Q;MADP,IAAI,CAAC,6BA
AW,KAAX,CAAL,C;QAAwB,MAAM,uBAAmB,sCtFjBzC,oBsFiByC,CAAnB,C;;MAC9B,OAAO,sD;K;IAMX,o
C;MAakC,Q;MAA9B,OAAW,6BAAW,KAAX,CAAJ,GAAuB,sDAAvB,GAAuC,I;K;;;;;;;ICvBhB,yC;MA2B9B,u
C;MA1BA,wB;MAIA,gB;MjMQA,IAAI,EiMDS,iBAAY,IAAb,MAAuB,iBAAvB,CjMCR,CAAJ,C;QACI,ciMDQ,
iBAAY,IAAhB,GACI,8CADJ,GAGI,sCAA0B,aAA1B,qC;QjMDR,MAAM,gCAAyB,OAAQ,WAAjC,C;;K;yCiM
KV,Y;MAAwC,Q;MAAA,oB;MACpC,iB;QAD8B,OACtB,G;WACR,oD;QAF8B,OAEF,SAAL,SAAK,C;WAC5B,
6C;QAH8B,OAGd,iBAAK,SAAL,C;WAChB,8C;QAJ8B,OAIb,kBAAM,SAAN,C;;QAJa,mC;K;IAOlC,qC;MAAA
,yC;MACI,YAGqC,oBAAgB,IAAhB,EAAsB,IAAtB,C;K;iGAQJ,Y;MAAQ,gB;K;4DAEzC,gB;MAOI,8DAAqC,I
AArC,C;K;gEAEJ,gB;MAMI,uDAA8B,IAA9B,C;K;4DAEJ,gB;MAMI,wDAA+B,IAA/B,C;K;;;IArCR,iD;MAAA,
gD;QAAA,+B;;MAAA,yC;K;;2CArCJ,Y;MAWI,oB;K;2CAXJ,Y;MAeI,gB;K;6CAfJ,0B;MAAA,2BAWI,8CAXJ,E
AeI,kCAfJ,C;K;yCAAA,Y;MAAA,c;MAWI,yD;MAIA,qD;MAfJ,a;K;uCAAA,iB;MAAA,4IAWI,4CAXJ,IAeI,oC
AfJ,I;K;ICLA,kC;MAAA,e;MAAA,iB;MAAA,uB;K;IAAA,gC;MAAA,mC;O;MAYI,4D;MAKA,8C;MAKA,gD;K
;;IAVA,2C;MAAA,sB;MAAA,mC;K;;IAKA,oC;MAAA,sB;MAAA,4B;K;;IAKA,qC;MAAA,sB;MAAA,6B;K;;IAt
BJ,4B;MAAA,mG;K;;IAAA,iC;MAAA,a;aAAA,W;UAAA,wC;aAAA,I;UAAA,iC;aAAA,K;UAAA,kC;;UAAA,6
D;;K;;6ECAA,yB;MAAA,4F;MAAA,2B;QASI,MAAM,mCAA8B,0EAA9B,C;O;KATV,C;ICkCA,+D;MAaW,Q;
MAAP,OAAO,8CAAO,KAAP,EAAc,UAAd,EAA0B,QAA1B,oC;K;IAGX,kC;MAIiB,Q;MAAb,wBAAa,KAAb,g
B;QAAa,WAAA,KAAb,M;QACI,yBAAO,IAAP,C;;MACJ,OAAO,S;K;mFAGX,qB;MAGwD,gCAAO,EAAP,C;K
;qFAExD,4B;MAG4E,OAAA,yBAAO,KAAP,CALpB,gBAAO,EAAP,C;K;qFAOxD,4B;MAGmE,OAAA,yBAAO
,KAAP,CAVX,gBAAO,EAAP,C;K;IAaxD,wD;MAEQ,sB;QAAqB,yBAAO,UAAU,OAAV,CAAP,C;WACrB,sD;
QAA4B,yBAAO,OAAP,C;WAC5B,2B;QAAmB,yBAAO,kBAAP,C;;QACX,yBAAe,SAAR,OAAQ,CAAf,C;K;ItL
7EhB,+B;MAY6B,kBAAlB,QAAQ,SAAR,EAAc,EAAd,C;MACH,IX0EE,WW1EE,GAAK,CAAT,C;QAAY,MA
AM,gCAAyB,oEAAzB,C;MADtB,OX4EO,W;K;IWvEX,wC;MAgBsC,IAA3B,I;MAAAA,qCAAiB,KAAjB,C;MAA
A,iB;QAA2B,MAAM,gCAAyB,8BAAO,SAAP,4CAA+C,KAAxE,C;;MAAxC,OAAO,I;K;IAGX,qC;MAY6B,kB
AAlB,QAAQ,SAAR,EAAc,EAAd,C;MAAP,OXmEqB,WWnEa,IAAM,CXmEjC,GAAqB,WAArB,GAA+B,I;K;I
WhE1C,8C;MAgBI,WAAW,KAAX,C;MAC4B,kBAArB,QAAQ,SAAR,EAAc,KAAd,C;MAAP,OX+CqB,WW/Cg
B,IAAM,CX+CpC,GAAqB,WAArB,GAA+B,I;K;IW5C1C,gC;MAWI,IAAY,CAAR,8BAAW,CAAf,C;QACI,OA
AO,YAAM,SAAN,C;;MAEX,MAAM,gCAAyB,SAAM,SAAN,4BAAzB,C;K;IAGV,yC;MAkBW,Q;MANP,IAAI,
EAAU,CAAV,sBAAa,EAAb,CAAJ,C;QACI,MAAM,gCAAyB,oBAAiB,KAAjB,4CAAzB,C;;MAEV,IAAI,YAA

O,CAAP,IAAY,aAAQ,KAAxB,C;QACI,MAAM,gCAAyB,WAAQ,SAAR,mDAAwD,KAAjF,C;;MAEH,IAAI,YA
AO,EAAX,C;QACH,mBAAM,SAAN,C;;QAEA,0BAAM,SAAN,IAAa,EAAb,C;;MAHJ,W;K;IAuFJ,8B;MAWsC,
+B;K;0EAEtC,4B;MAM8D,OAAK,oBAAL,SAAK,CAAL,GAAkB,K;K;IAEhF,gD;MAQoC,0B;QAAA,aAAsB,K;
MACtD,IAAI,cAAQ,KAAZ,C;QAAmB,OAAO,I;MAC1B,IAAI,CAAC,UAAL,C;QAAiB,OAAO,K;MAExB,gBA
AqB,cAAL,SAAK,C;MACrB,iBAAuB,cAAN,KAAM,C;MAEhB,yBAAa,U;MAAb,U;QAA2B,OFrMyB,oBEqMz
B,SFrMyB,CAAY,cAfrB,YAAY,CAAZ,CEoNhB,KFrMyB,oBEqMI,UFrMJ,CAAY,cAfrB,YAAY,CAAZ,C;;MEo
NlD,W;K;IAGJ,gC;MAGyC,QAAQ,cAAA,sCAAK,cAAL,EAAoB,sCAAK,cAAzB,CAAR,6B;K;IuL3OzC,6C;MA
e6B,4B;QAAA,eAAuB,G;MAChD,wCAAsB,EAAtB,EAA0B,YAA1B,C;K;IAEJ,mE;MAKwC,yB;QAAA,YAAoB
,E;MAAI,4B;QAAA,eAAuB,G;MrMEnF,IAAI,CmBwR+C,CAAC,QkLzR5C,YlLyR4C,CnBxRpD,C;QACI,cqMFi
C,wC;QrMGjC,MAAM,gCAAyB,OAAQ,WAAjC,C;;MqMFV,cAAY,gB;MAEC,yBAAS,mBAAS,YAAA,SAAU,
OAAV,EAAmB,OAAM,KAAzB,CAAT,I;MAAT,wBAAiD,kBAAkB,SAAlB,C;MA2E9D,gBAAgB,iBA3ET,OA2
ES,C;M/Lg7CT,kBAAoB,gB;MAwSd,gB;MADb,YAAY,C;MACC,O+LnyDN,O/LmyDM,W;kBAAb,OAAa,cAAb
,C;QAAa,sB;QA9RsB,U;QAAA,cA8RT,oBAAmB,cAAnB,EAAmB,sBAAnB,U;Q+LntDlB,kB;;YAHA,CAAC,Y
AAS,CAAT,IAAc,qBAAf,KAA4C,Q/LstDG,I+LttDH,C;UAC5C,a;;UAEA,4B;UA/E+B,uB;;YnLgHzB,kC;YAAA,
wBZkrDyC,IYlrDzC,C;YAAA,qB;YAAA,oB;YAAA,oB;YAAd,gE;cACI,ImLjHkD,CAAI,aAAH,UnLiHrC,YZirD
qC,IYjrDrC,YAAK,OAAL,EmLjHqC,CAAG,CnLiHtD,C;gBACI,sBAAO,O;gBAAP,wB;;;YAGR,sBAAO,E;;;Um
LrHH,iD;UAGI,gCAA2B,EAA3B,C;YAHJ,2BAGqC,I;iBACjC,IAAK,a/L8xD0C,I+L9xD1C,gBAAyB,uBAAzB,C
AAL,C;YAJJ,2B/LkyDmD,IO9kDsB,WwLhNI,0BAAuC,mBAAvC,IxLgNJ,C;;YwLpNzE,2BAKY,I;;UA0ER,iEl
MND,yBkMMC,4B/LmtD+C,I;;QA9RpB,8B;UAA6C,6B;;;M+LrgDhF,OAkFK,S/Lo7CE,W+Lp7CF,EAAO,mBA
Ac,kBAAd,CAAP,EAA0C,IAA1C,CACA,W;K;IAxET,+B;MAgByC,gCAAc,EAAd,C;K;IAEzC,6C;MAGgC,yB;Q
AAA,YAAoB,E;MAM3C,Q;MALL,cAAY,gB;M/LurBL,kBAAS,gB;MA2FA,U;MAAA,S+LhxBM,O/LgxBN,W;
MAAhB,OAAgB,gBAAhB,C;QAAgB,2B;QAAM,Ia7hB6B,CAAC,Qb6hBhB,Oa7hBgB,Cb6hB9B,C;UAAwB,WA
AY,WAAI,OAAJ,C;;M+L9wBrD,kB/L+wBE,W;MAmrBA,oBAAM,iBAAa,qCAAwB,EAAxB,CAAb,C;MAuEA,
U;MAAA,+B;MAAb,OAAa,gBAAb,C;QAAa,wB;QACT,aAAY,uBAAc,IAAd,E;;M+L5gDhB,sBAAsB,CAGjB,o
B/L0gDE,a+L1gDF,CAHiB,mBAGF,C;MAEP,yBAAS,mBAAS,YAAA,SAAU,OAAV,EAAmB,OAAM,KAAzB,
CAAT,I;MAAT,wBAAiD,kBAAkB,SAAlB,C;MAmC9D,gBAAgB,iBAnCT,OAmCS,C;M/Lg7CT,oBAAoB,gB;M
AwSd,kB;MADb,YAAY,C;MACC,S+L3vDN,O/L2vDM,W;MAAb,OAAa,gBAAb,C;QAAa,0B;QA9RsB,U;QAA
A,cA8RT,oBAAmB,cAAnB,EAAmB,sBAAnB,U;Q+LntDlB,kB;Q/Lq7C2B,c+Lx7C3B,CAAC,YAAS,CAAT,IAA
c,qBAAf,KAA4C,Q/LstDG,M+LttDH,C/Lw7CjB,G+Lv7C3B,I/Lu7C2B,G+Lr7C3B,oBAxCmG,Q/L2vDpD,M+L3
vDoD,kBAwCnG,YlMND,yBkMMC,4B/LmtD+C,MA9RpB,U;UAA6C,+B;;;M+L79ChF,OA0CK,S/Lo7CE,a+Lp7
CF,EAAO,mBAAc,kBAAd,CAAP,EAA0C,IAA1C,CACA,W;K;IAjCI,8C;MAAA,qB;QAEG,IAAG,QAAH,EAAG
,CAAH,C;UAEQ,IAAA,EAAG,OAAH,GAAY,cAAO,OAAnB,C;YAHZ,OAGyC,c;;YAHzC,OAIoB,E;;UAJpB,O
AOY,iBAAS,E;O;K;IAfjC,0C;MAKgC,sB;QAAA,SAAiB,M;MAC7C,OAYK,eAXA,OADL,uBACK,EAAI,4BAA
J,CAWA,EAAa,IAAb,C;K;IAET,gC;MAAwC,uB;;QnLkDtB,gC;QAAA,gC;QAAA,mB;QAAA,kB;QAAA,kB;QA
Ad,0D;UACI,ImLnD+C,CAAI,aAAH,UnLmDlC,iCAAK,KAAL,EmLnDkC,CAAG,CnLmDnD,C;YACI,sBAAO,
K;YAAP,wB;;;QAGR,sBAAO,E;;;Mf5CA,4B;MkMX6B,OAA8C,OAAM,EAAV,GAAc,gBAAd,GAA0B,E;K;IA
GpF,wC;MAAkB,W;K;IAC9B,oD;MAAA,uB;QAAkB,wBAAS,I;O;K;IAFvC,mC;MACI,IAAA,MlLgMgD,YAAU
,CkLhM1D,C;QAD4C,OACxB,wB;;QADwB,OAEpC,kC;K;mBAGZ,yB;M/L86CA,+D;MAwSA,wE;M+LttDA,sF;
QAKI,gBAAgB,2B;Q/Lg7CT,kBAAoB,gB;QAwSd,gB;QADb,YAAY,C;QACC,2B;QAAb,OAAa,cAAb,C;UAAa,
sB;UA9RsB,U;UAAA,cA8RT,oBAAmB,cAAnB,EAAmB,sBAAnB,U;U+LntDlB,kB;U/Lq7C2B,c+Lx7C3B,CAA
C,YAAS,CAAT,IAAc,qBAAf,KAA4C,Q/LstDG,I+LttDH,C/Lw7CjB,G+Lv7C3B,I/Lu7C2B,G+Lr7C3B,sC/LmtD
+C,I+LntD/C,alMND,yBkMMC,4B/LmtD+C,IA9RpB,U;YAA6C,6B;;;Q+Lz7ChF,OAMK,S/Lo7CE,W+Lp7CF,E
AAO,mBAAc,kBAAd,CAAP,EAA0C,IAA1C,CACA,W;O;KAbT,C;6E9EgSA,0B;MAGmE,OAAA,SAAK,gBAA
O,GAAP,C;K;qFAExE,yB;MAAA,yD;MAAA,gC;QAO2B,gBAAhB,oB;QAAsB,apHrU7B,W;QoHqUA,OpHpUO
,SoHoUqC,W;O;KAPhD,C;uFAUA,yB;MAAA,iE;MAAA,0C;QAQmC,gBAAxB,mBAAc,QAAd,C;QAA8B,apHh
VrC,W;QoHgVA,OpH/UO,SoH+U6C,W;O;KARxD,C;IAWA,oC;MAIiB,Q;MAAb,wBAAa,KAAb,gB;QAAa,WA
AA,KAAb,M;QACI,yBAAO,IAAP,C;;MACJ,OAAO,S;K;IAGX,oC;MAIiB,Q;MAAb,wBAAa,KAAb,gB;QAAa,
WAAA,KAAb,M;QACI,yBAAO,IAAP,C;;MACJ,OAAO,S;K;6EAGX,yB;MAAA,2D;MAAA,8C;QAI+F,MAAM,

8B;O;KAJrG,C;qFAMA,qB;MAG8D,gCAAO,EAAP,C;K;qFAE9D,4B;MAGkF,OAAA,yBAAO,KAAP,CALpB,g
BAAO,EAAP,C;K;qFAO9D,4B;MAG4E,OAAA,yBAAO,KAAP,CAVd,gBAAO,EAAP,C;K;qFAY9D,4B;MAGy
E,OAAA,yBAAO,KAAP,CAfX,gBAAO,EAAP,C;K;qFAiB9D,4B;MAG8E,OAAA,yBAAO,KAAP,CApBhB,gBA
AO,EAAP,C;K;qFAsB9D,4B;MAGyE,OAAA,yBAAO,KAAP,CAzBX,gBAAO,EAAP,C;K;qFA2B9D,4B;MAG4
E,OAAA,yBAAO,KAAP,CA9Bd,gBAAO,EAAP,C;K;I7Hrb9D,iC;MAK0C,iCAAqB,EAArB,C;K;IAE1C,0C;MA
QyC,IAAtB,I;MAAA,qBAAL,SAAK,EAAY,KAAZ,C;MAAL,iB;QAA2B,OAAO,I;;MAA5C,UAAU,I;MACV,IA
AI,MAAM,sCAAK,UAAX,IAAwB,MAAM,sCAAK,UAAvC,C;QAAkD,OAAO,I;MACzD,OAAW,OAAJ,GAAI,
C;K;IAGf,kC;MAK4C,kCAAsB,EAAtB,C;K;IAE5C,2C;MAQyC,IAAtB,I;MAAA,qBAAL,SAAK,EAAY,KAAZ,
C;MAAL,iB;QAA2B,OAAO,I;;MAA5C,UAAU,I;MACV,IAAI,MAAM,uCAAM,UAAZ,IAAyB,MAAM,uCAAM,
UAAzC,C;QAAoD,OAAO,I;MAC3D,OAAW,QAAJ,GAAI,C;K;IAGf,gC;MAKwC,gCAAoB,EAApB,C;K;IAExC,
yC;MAQI,WAAW,KAAX,C;MAEA,aAAa,SAAK,O;MAClB,IAAI,WAAU,CAAd,C;QAAiB,OAAO,I;MAExB,S;
MACA,c;MACA,S;MAEA,gBAAgB,qBAAK,CAAL,C;MAChB,IAAI,YAAY,EAAhB,C;QACI,IAAI,WAAU,CA
Ad,C;UAAiB,OAAO,I;QAExB,QAAQ,C;QAER,IAAI,cAAa,EAAjB,C;UACI,aAAa,I;UACb,QAAQ,W;eACL,IAA
I,cAAa,EAAjB,C;UACH,aAAa,K;UACb,QAAQ,W;;UAER,OAAO,I;;QAEX,QAAQ,C;QACR,aAAa,K;QACb,QA
AQ,W;;MAIZ,uBAAuB,S;MAEvB,qBAAqB,gB;MACrB,aAAa,C;MACb,aAAU,KAAV,MAAsB,MAAtB,M;QAC
I,YAAY,QAAQ,qBAAK,CAAL,CAAR,EAAiB,KAAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;QACtB,I
AAI,SAAS,cAAb,C;UACI,IAAI,mBAAkB,gBAAtB,C;YACI,iBAAiB,QAAQ,KAAR,I;YAEjB,IAAI,SAAS,cAAb,
C;cACI,OAAO,I;;;YAGX,OAAO,I;;;QAIf,6BAAU,KAAV,C;QAEA,IAAI,UAAS,QAAQ,KAAR,IAAT,CAAJ,C;U
AA4B,OAAO,I;QAEnC,kBAAU,KAAV,I;;MAGJ,OAAW,UAAJ,GAAgB,MAAhB,GAA4B,CAAC,MAAD,I;K;IA
GvC,iC;MAK0C,iCAAqB,EAArB,C;K;IAE1C,0C;MAQI,WAAW,KAAX,C;MAEA,aAAa,SAAK,O;MAClB,IAAI
,WAAU,CAAd,C;QAAiB,OAAO,I;MAExB,S;MACA,c;MACA,S;MAEA,gBAAgB,qBAAK,CAAL,C;MAChB,IA
AI,YAAY,EAAhB,C;QACI,IAAI,WAAU,CAAd,C;UAAiB,OAAO,I;QAExB,QAAQ,C;QAER,IAAI,cAAa,EAAjB
,C;UACI,aAAa,I;UACb,gC;eACG,IAAI,cAAa,EAAjB,C;UACH,aAAa,K;UACb,6B;;UAEA,OAAO,I;;QAEX,QA
AQ,C;QACR,aAAa,K;QACb,6B;;MAIJ,2C;MAEA,qBAAqB,gB;MACrB,e;MACA,aAAU,KAAV,MAAsB,MAAt
B,M;QACI,YAAY,QAAQ,qBAAK,CAAL,CAAR,EAAiB,KAAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,
I;QACtB,IAAI,uBAAS,cAAT,KAAJ,C;UACI,IAAI,uBAAkB,gBAAlB,CAAJ,C;YACI,iBAAiB,8BAAQ,KAAR,E;
YAEjB,IAAI,uBAAS,cAAT,KAAJ,C;cACI,OAAO,I;;;YAGX,OAAO,I;;;QAIf,6CAAU,KAAV,E;QAEA,IAAI,uB
AAS,8BAAQ,KAAR,EAAT,KAAJ,C;UAA4B,OAAO,I;QAEnC,6CAAU,KAAV,E;;MAGJ,OAAW,UAAJ,GAAgB
,MAAhB,GAA6B,MAAD,a;K;IAIvC,kC;MAAyD,MAAM,0BAAsB,6BAA0B,KAA1B,MAAtB,C;K;uEyBhI/D,yB
;MAAA,oC;MAAA,uC;QAII,iBAAiB,C;QACCjB,eAAe,mBAAS,CAAT,I;QACCf,iBAAiB,K;QAEjB,OAAO,cAAc,
QAArB,C;UACI,YAAgB,CAAC,UAAL,GAAiB,UAAjB,GAAiC,Q;UAC7C,YAAY,UAAU,iCAAK,KAAL,EAAV
,C;UAEZ,IAAI,CAAC,UAAL,C;YACI,IAAI,CAAC,KAAL,C;cACI,aAAa,I;;cAEb,0BAAc,CAAd,I;;YAEJ,IAAI,C
AAC,KAAL,C;cACI,K;;cAEA,sBAAY,CAAZ,I;;;QAIZ,OAAO,8BAAY,UAAZ,EAAwB,WAAW,CAAX,IAAxB,
C;O;KAzBX,C;yEA4BA,yB;MAAA,8B;MA5BA,oC;MA4BA,uC;QAIK,Q;QAAsB,kBAAtB,2D;QA5BD,iBAAiB,
C;QACjB,eAAe,qBAAS,CAAT,I;QACCf,iBAAiB,K;QAEjB,OAAO,cAAc,QAArB,C;UACI,YAAgB,CAAC,UAAL,
GAAiB,UAAjB,GAAiC,Q;UAC7C,YAsBwB,SAtBZ,CAAU,mCAAK,KAAL,EAAV,C;UAEZ,IAAI,CAAC,UAA
L,C;YACI,IAAI,CAAC,KAAL,C;cACI,aAAa,I;;cAEb,0BAAc,CAAd,I;;YAEJ,IAAI,CAAC,KAAL,C;cACI,K;;cA
EA,sBAAY,CAAZ,I;;;QAWZ,OAPO,gCAAY,UAAZ,EAAwB,WAAW,CAAX,IAAxB,CAOgC,W;O;KAJ3C,C;iF
AMA,yB;MAAA,mD;MAAA,oC;MAAA,uC;QAIuB,UAAL,MAAK,EAAL,MAAK,EAAL,M;QAAK,mBAAL,SA
AK,C;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;Y
ACI,OAAO,8BAAY,KAAZ,EAAmB,gBAAnB,C;QAEf,OAAO,E;O;KARX,C;mFAWA,yB;MAAA,8B;MAXA,m
D;MAAA,oC;MAWA,uC;QAIK,Q;QAAsB,kBAAtB,2D;QAAsB,oB;;UAXJ,kC;UAAA,qBAAL,WAAK,C;UAAL,
qB;UAAA,oB;UAAA,oB;UAAd,0D;YACI,IAAI,CAUyB,SAVxB,CAAU,mCAAK,KAAL,EAAV,CAAL,C;cACI,
mBAAO,gCAAY,KAAZ,EAAmB,kBAAnB,C;cAAP,qB;;UAER,mBAAO,E;;;QAOP,OAA4C,2B;O;KAJhD,C;6E
AMA,yB;MAAA,mD;MAAA,+C;MAAA,oC;MAAA,uC;QAIkB,Q;QAAA,OAAa,SAAR,YAAL,SAAK,CAAQ,C
AAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,IAAI,CAAC,UAAU,iCAAK,KAAL,EAAV,CAAL,C;YACI,OA
AO,8BAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;;QAEf,OAAO,E;O;KARX,C;+EAWA,yB;MAAA,8B;MAXA,m
D;MAAA,+C;MAAA,oC;MAWA,uC;QAIK,Q;QAAsB,kBAAtB,2D;QAAsB,kB;;UAXT,U;UAAA,SAAa,SAAR,Y

AAL,WAAK,CAAQ,CAAb,W;UAAd,OAAc,gBAAd,C;YAAc,yB;YACV,IAAI,CAUuB,SAVtB,CAAU,mCAAK,
KAAL,EAAV,CAAL,C;cACI,iBAAO,gCAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;cAAP,mB;;;UAER,iBAAO,E;
;;QAOP,OAA0C,yB;O;KAJ9C,C;IAMA,kC;MAhEI,iBAAiB,C;MACjB,eAAe,mBAAS,CAAT,I;MACf,iBAAiB,K;
MAEjB,OAAO,cAAc,QAArB,C;QACI,YAAgB,CAAC,UAAL,GAAiB,UAAjB,GAAiC,Q;QAC7C,YA6DgE,4BA
7D1C,iCAAK,KAAL,EA6D0C,E;QA3DhE,IAAI,CAAC,UAAL,C;UACI,IAAI,CAAC,KAAL,C;YACI,aAAa,I;;Y
AEb,0BAAc,CAAd,I;;UAEJ,IAAI,CAAC,KAAL,C;YACI,K;;YAEA,sBAAY,CAAZ,I;;;MAkDiD,OA9CtD,8BAA
Y,UAAZ,EAAwB,WAAW,CAAX,IAAxB,C;K;IAgDX,kC;MAzCK,Q;MAAsB,kBAAtB,2D;MA5BD,iBAAiB,C;
MACjB,eAAe,qBAAS,CAAT,I;MACf,iBAAiB,K;MAEjB,OAAO,cAAc,QAArB,C;QACI,YAAgB,CAAC,UAAL,
GAAiB,UAAjB,GAAiC,Q;QAC7C,YAkEoD,4BAlE9B,mCAAK,KAAL,EAkE8B,E;QAhEpD,IAAI,CAAC,UAAL
,C;UACI,IAAI,CAAC,KAAL,C;YACI,aAAa,I;;YAEb,0BAAc,CAAd,I;;UAEJ,IAAI,CAAC,KAAL,C;YACI,K;;YA
EA,sBAAY,CAAZ,I;;;MAuDqC,OAnD1C,gCAAY,UAAZ,EAAwB,WAAW,CAAX,IAAxB,CAOgC,W;K;IA8C3
C,uC;MAGsE,oB;;QA3C/C,gC;QAAA,gC;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,CA0CsE,4BA
1C3D,iCAAK,KAAL,EA0C2D,EA1C1E,C;YACI,mBAAO,8BAAY,KAAZ,EAAmB,gBAAnB,C;YAAP,qB;;;QAE
R,mBAAO,E;;;MAuC2D,uB;K;IAEtE,uC;MAlCK,Q;MAAsB,kBAAtB,2D;MAAsB,oB;;QAXJ,kC;QAAA,wBAA
L,WAAK,C;QAAL,qB;QAAA,oB;QAAA,oB;QAAd,0D;UACI,IAAI,CA+C0D,4BA/C/C,mCAAK,KAAL,EA+C+
C,EA/C9D,C;YACI,mBAAO,gCAAY,KAAZ,EAAmB,kBAAnB,C;YAAP,qB;;;QAER,mBAAO,E;;;MA4C+C,OAr
CV,2B;K;IAuChD,qC;MAGoE,kB;;QApClD,Q;QAAA,OAAa,WAAR,yBAAQ,CAAb,W;QAAd,OAAc,cAAd,C;U
AAc,uB;UACV,IAAI,CAmCkE,4BAnCvD,iCAAK,KAAL,EAmCuD,EAnCtE,C;YACI,iBAAO,8BAAY,CAAZ,E
AAe,QAAQ,CAAR,IAAf,C;YAAP,mB;;;QAER,iBAAO,E;;;MAgCyD,qB;K;IAEpE,qC;MA3BK,Q;MAAsB,kBA
AtB,2D;MAAsB,kB;;QAXT,U;QAAA,SAAa,WAAR,eAAL,WAAK,CAAQ,CAAb,W;QAAd,OAAc,gBAAd,C;UA
Ac,yB;UACV,IAAI,CAwCsD,4BAxC3C,mCAAK,KAAL,EAwC2C,EAxC1D,C;YACI,iBAAO,gCAAY,CAAZ,E
AAe,QAAQ,CAAR,IAAf,C;YAAP,mB;;;QAER,iBAAO,E;;;MAqC6C,OA9BV,yB;K;IAgC9C,2B;MA9FI,iBAAiB,
C;MACjB,eAAe,mBAAS,CAAT,I;MACf,iBAAiB,K;MAEjB,OAAO,cAAc,QAArB,C;QACI,YAAgB,CAAC,UAA
L,GAAiB,UAAjB,GAAiC,Q;QAC7C,mCAAsB,iCAAK,KAAL,EAAtB,E;QAEA,IAAI,CAAC,UAAL,C;UACI,IA
AI,CAAC,KAAL,C;YACI,aAAa,I;;YAEb,0BAAc,CAAd,I;;UAEJ,IAAI,CAAC,KAAL,C;YACI,K;;YAEA,sBAAY
,CAAZ,I;;;MAgF+B,OA5EpC,8BAAY,UAAZ,EAAwB,WAAW,CAAX,IAAxB,C;K;yEA8EX,yB;MAAA,8B;MA
AA,qC;MAAA,4B;QAI2C,Q;QAAD,OAAuB,KAAtB,2DAAsB,CAAO,W;O;KAJxE,C;IAMA,gC;MAGoD,oB;;Q
A1E7B,gC;QAAA,gC;QAAL,mB;QAAA,kB;QAAA,kB;QAAd,0D;UACI,IAAI,wBAAW,iCAAK,KAAL,EAAX,E
AAJ,C;YACI,mBAAO,8BAAY,KAAZ,EAAmB,gBAAnB,C;YAAP,qB;;;QAER,mBAAO,E;;;MAsEyC,uB;K;mFA
EpD,yB;MAAA,8B;MAAA,+C;MAAA,4B;QAIgD,Q;QAAD,OAAuB,UAAtB,2DAAsB,CAAY,W;O;KAJlF,C;IA
MA,8B;MAGkD,kB;;QApEhC,Q;QAAA,OAAa,WAAR,yBAAQ,CAAb,W;QAAd,OAAc,cAAd,C;UAAc,uB;UAC
V,IAAI,wBAAW,iCAAK,KAAL,EAAX,EAAJ,C;YACI,iBAAO,8BAAY,CAAZ,EAAe,QAAQ,CAAR,IAAf,C;YA
AP,mB;;;QAER,iBAAO,E;;;MAgEuC,qB;K;+EAElD,yB;MAAA,8B;MAAA,2C;MAAA,4B;QAI8C,Q;QAAD,OA
AuB,QAAtB,2DAAsB,CAAU,W;O;KAJ9E,C;IAMA,8C;MAU8C,uB;QAAA,UAAgB,E;MAO5C,Q;MANd,IAAI,
SAAS,CAAb,C;QACI,MAAM,gCAAyB,oBAAiB,MAAjB,wBAAzB,C;MACV,IAAI,UAAU,SAAK,OAAAnB,C;Q
ACI,OAAY,mBAAL,SAAK,EAAY,CAAZ,EAAe,SAAK,OAApB,C;MAEhB,SAAS,mBAAc,MAAd,C;MACK,gB
AAS,SAAK,OAAd,I;MAAd,aAAU,CAAV,iB;QACI,EAAG,gBAAO,OAAP,C;MACP,EAAG,gBAAO,SAAP,C;M
ACH,OAAO,E;K;IAGX,gD;MASwC,uB;QAAA,UAAgB,E;MACnD,Q;MAAD,OAAuB,SAAtB,6DAAsB,EAAS,
MAAT,EAAiB,OAAjB,CAA0B,W;K;IAErD,4C;MAU4C,uB;QAAA,UAAgB,E;MAQ1C,Q;MAPd,IAAI,SAAS,C
AAb,C;QACI,MAAM,gCAAyB,oBAAiB,MAAjB,wBAAzB,C;MACV,IAAI,UAAU,SAAK,OAAAnB,C;QACI,OA
AY,mBAAL,SAAK,EAAY,CAAZ,EAAe,SAAK,OAApB,C;MAEhB,SAAS,mBAAc,MAAd,C;MACT,EAAG,gBA
AO,SAAP,C;MACW,gBAAS,SAAK,OAAd,I;MAAd,aAAU,CAAV,iB;QACI,EAAG,gBAAO,OAAP,C;MACP,O
AAO,E;K;IAGX,8C;MASsC,uB;QAAA,UAAgB,E;MACjD,Q;MAAD,OAAuB,OAAtB,6DAAsB,EAAO,MAAP,E
AAe,OAAf,CAAwB,W;K;2FAEnD,qB;MAWI,OAAO,qBAAgB,SAAK,OAAL,KAAe,C;K;+EAG1C,qB;MAMoD,
4BAAU,C;K;sFAE9D,qB;MAMuD,0BAAS,C;K;mFAMhE,yB;MAAA,2C;MAAA,4B;QAMuD,QAAC,kB;O;KA
NxD,C;yFAQA,yB;MAAA,2C;MAAA,4B;QAWI,OAAO,qBAAqB,QAAL,SAAK,C;O;KAXhC,C;IAiB4D,+C;MA
AA,kC;MAAS,uB;MACjE,eAAoB,C;K;gDAEpB,Y;MAA2C,gB;MAAA,iE;MAAJ,4C;K;+CAEvC,Y;MAAyC,sB
AAQ,yB;K;;IARrD,+B;MAG4D,4C;K;+EAQ5D,qB;MAE8C,uCAAQ,E;K;+EAEtD,mC;MASI,OA5DgD,qBAAU,

CA4D1D,GAAe,cAAf,GAAmC,S;K;6EAEvC,yB;MAAA,2C;MAAA,0C;QASI,OAAI,kBAAJ,GAAe,cAAf,GAAm
C,S;O;KATvC,C;IAeI,mC;MAAQ,uBAAG,mBAAS,CAAT,IAAH,C;K;IAMR,qC;MAAQ,OAAA,SAAK,OAAL,G
AAc,CAAd,I;K;IAEZ,8C;MAIuB,Q;MAAA,0BAAS,CAAT,I;MAAnB,OAAgB,CAAT,8BACgB,gBAAZ,qBAAK,
KAAL,CAAY,CADhB,IAEoB,eAAhB,qBAAK,QAAQ,CAAR,IAAL,CAAgB,C;K;IAG/B,uC;MAGuD,ON3IyC,o
BM2I/B,KAAM,MN3IyB,EM2IlB,KAAM,aAAN,GAAqB,CAArB,IN3IkB,C;K;IM6IhG,yC;MAGqE,qCAAY,KA
AM,MAAlB,EAAyB,KAAM,aAAN,GAAqB,CAArB,IAAzB,C;K;uFAErE,iC;MAS2E,2BAAY,KAAZ,EAAmB,G
AAnB,C;K;mFAE3E,2C;MAO0D,wB;QAAA,WAAgB,gB;MAAkB,OAAA,8BAAY,UAAZ,EAAwB,QAAxB,CA
AkC,W;K;IAE9H,uC;MAG6D,OAAA,8BAAY,KAAM,MAAlB,EAAyB,KAAM,aAAN,GAAqB,CAArB,IAAzB,C
AAiD,W;K;IAE9G,sE;MAImD,qC;QAAA,wBAAgC,S;MAC/E,YAAY,sBAAQ,SAAR,C;MACZ,OAAW,UAAS,E
AApB,GAAwB,qBAAxB,GNjL4F,oBMiL/B,CNjL+B,EMiL5B,KNjL4B,C;K;IMoLhG,wE;MAIqD,qC;QAAA,wB
AAgC,S;MACjF,YAAY,sBAAQ,SAAR,C;MACZ,OAAW,UAAS,EAApB,GAAwB,qBAAxB,GN1L4F,oBM0L/B,
CN1L+B,EM0L5B,KN1L4B,C;K;IM6LhG,qE;MAIkD,qC;QAAA,wBAAgC,S;MAC9E,YAAY,sBAAQ,SAAR,C;
MACZ,OAAW,UAAS,EAApB,GAAwB,qBAAxB,GNnM4F,oBMmM/B,QAAQ,CAAR,INnM+B,EMmMpB,gBN
nMoB,C;K;IMsMhG,uE;MAIoD,qC;QAAA,wBAAgC,S;MAChF,YAAY,sBAAQ,SAAR,C;MACZ,OAAW,UAAS,
EAApB,GAAwB,qBAAxB,GN5M4F,oBM4M/B,QAAQ,SAAU,OAAlB,IN5M+B,EM4ML,gBN5MK,C;K;IM+Mh
G,0E;MAIuD,qC;QAAA,wBAAgC,S;MACnF,YAAY,0BAAY,SAAZ,C;MACZ,OAAW,UAAS,EAApB,GAAwB,
qBAAxB,GNrN4F,oBMqN/B,CNrN+B,EMqN5B,KNrN4B,C;K;IMwNhG,4E;MAIyD,qC;QAAA,wBAAgC,S;MA
CrF,YAAY,0BAAY,SAAZ,C;MACZ,OAAW,UAAS,EAApB,GAAwB,qBAAxB,GN9N4F,oBM8N/B,CN9N+B,E
M8N5B,KN9N4B,C;K;IMiOhG,yE;MAIsD,qC;QAAA,wBAAgC,S;MAClF,YAAY,0BAAY,SAAZ,C;MACZ,OA
AW,UAAS,EAApB,GAAwB,qBAAxB,GNvO4F,oBMuO/B,QAAQ,CAAR,INvO+B,EMuOpB,gBNvOoB,C;K;IM
0OhG,2E;MAIwD,qC;QAAA,wBAAgC,S;MACpF,YAAY,0BAAY,SAAZ,C;MACZ,OAAW,UAAS,EAApB,GAA
wB,qBAAxB,GNhP4F,oBMgP/B,QAAQ,SAAU,OAAlB,INhP+B,EMgPL,gBNhPK,C;K;IMmPhG,oE;MAOI,IAAI
,WAAW,UAAf,C;QACI,MAAM,8BAA0B,gBAAa,QAAb,oCAAkD,UAAlD,OAA1B,C;MACV,SAAS,sB;MACT,
EAAG,qBAAY,SAAZ,EAAkB,CAAlB,EAAqB,UAArB,C;MACH,EAAG,gBAAO,WAAP,C;MACH,EAAG,qBA
AY,SAAZ,EAAkB,QAAlB,EAA4B,gBAA5B,C;MACH,OAAO,E;K;yFAGX,yB;MAAA,8B;MAAA,qD;MAAA,+
D;QAOK,Q;QAAD,OAAuB,aAAtB,2DAAsB,EAAa,UAAb,EAAyB,QAAzB,EAAmC,WAAnC,CAAgD,W;O;KA
P3E,C;IASA,uD;MAOI,+BAAa,KAAM,MAAnB,EAA0B,KAAM,aAAN,GAAqB,CAArB,IAA1B,EAAkD,WAAl
D,C;K;yFAEJ,yB;MAAA,8B;MAAA,qD;MAAA,gD;QAOK,Q;QAAD,OAAuB,aAAtB,2DAAsB,EAAa,KAAb,EA
AoB,WAApB,CAAiC,W;O;KAP5D,C;IASA,sD;MASI,IAAI,WAAW,UAAf,C;QACI,MAAM,8BAA0B,gBAAa,Q
AAb,oCAAkD,UAAlD,OAA1B,C;MAEV,IAAI,aAAY,UAAhB,C;QACI,OAAY,mBAAL,SAAK,EAAY,CAAZ,E
AAe,gBAAf,C;MAEhB,SAAS,mBAAc,oBAAU,QAAV,GAAqB,UAArB,KAAd,C;MACT,EAAG,qBAAY,SAAZ,
EAAkB,CAAlB,EAAqB,UAArB,C;MACH,EAAG,qBAAY,SAAZ,EAAkB,QAAlB,EAA4B,gBAA5B,C;MACH,O
AAO,E;K;uFAGX,yB;MAAA,8B;MAAA,mD;MAAA,kD;QASK,Q;QAAD,OAAuB,YAAtB,2DAAsB,EAAAY,UA
AZ,EAAwB,QAAxB,CAAkC,W;O;KAT7D,C;IAWA,yC;MAKqE,8BAAY,KAAM,MAAlB,EAAyB,KAAM,aAA
N,GAAqB,CAArB,IAAzB,C;K;uFAErE,yB;MAAA,8B;MAAA,mD;MAAA,mC;QAOK,Q;QAAD,OAAuB,YAAt
B,2DAAsB,EAAAY,KAAAZ,CAAmB,W;O;KAP9C,C;IASA,yC;MAKI,IAAI,wBAAW,MAAX,CAAAJ,C;QACI,OAA
O,8BAAY,MAAO,OAAnB,EAA2B,gBAA3B,C;;MAEX,OAAO,8BAAY,CAAZ,EAAe,gBAAf,C;K;IAGX,2C;MA
KI,IAAI,wBAAW,MAAX,CAAAJ,C;QACI,ONlWyE,oBMkWxD,MAAO,ONlWiD,C;;MMoW7E,OAAO,S;K;IAG
X,yC;MAKI,IAAI,sBAAS,MAAT,CAAAJ,C;QACI,OAAO,8BAAY,CAAZ,EAAe,mBAAS,MAAO,OAAhB,IAAf,C
;;MAEX,OAAO,8BAAY,CAAZ,EAAe,gBAAf,C;K;IAGX,2C;MAKI,IAAI,sBAAS,MAAT,CAAAJ,C;QACI,ONrX
wF,oBMqXvE,CNrXuE,EMqXpE,mBAAS,MAAO,OAAhB,INrXoE,C;;MMuX5F,OAAO,S;K;IAGX,sD;MAMI,I
AAK,qBAAU,MAAO,OAAP,GAAgB,MAAO,OAAvB,IAAV,CAAD,IAA6C,wBAAW,MAAX,CAA7C,IAAmE,s
BAAS,MAAT,CAAvE,C;QACI,OAAO,8BAAY,MAAO,OAAnB,EAA2B,mBAAS,MAAO,OAAhB,IAA3B,C;;M
AEX,OAAO,8BAAY,CAAZ,EAAe,gBAAf,C;K;IAGX,wD;MAMI,IAAK,qBAAU,MAAO,OAAP,GAAgB,MAAO
,OAAvB,IAAV,CAAD,IAA6C,wBAAW,MAAX,CAA7C,IAAmE,sBAAS,MAAT,CAAvE,C;QACI,ON7YwF,oB
M6YvE,MAAO,ON7YgE,EM6YxD,mBAAS,MAAO,OAAhB,IN7YwD,C;;MM+Y5F,OAAO,S;K;IAGX,mD;MA
KmF,oCAAkB,SAAlB,EAA6B,SAA7B,C;K;IAEnF,mD;MAKuE,sCAAkB,SAAlB,EAA6B,SAA7B,C;K;IAEvE,iF
;MAIsE,qC;QAAA,wBAAgC,S;MAClG,YAAY,sBAAQ,SAAR,C;MACL,Q;MAAAA,IAAI,UAAS,EAAb,C;QAAAA

,OAAiB,qB;;QA5JvB,U;QA4JM,OA5JgB,aAAtB,+DAAsB,EA4JyC,CA5JzC,EA4J4C,KA5J5C,EA4JmD,WA5Jn
D,CAAgD,W;;MA4JvE,W;K;IAGJ,mF;MAIwE,qC;QAAA,wBAAgC,S;MACpG,YAAY,sBAAQ,SAAR,C;MACL,
Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAAiB,qB;;QArKvB,U;QAqKM,OArKgB,aAAtB,+DAAsB,EAqKyC,CA
rKzC,EAqK4C,KArK5C,EAqKmD,WArKnD,CAAgD,W;;MAqKvE,W;K;IAGJ,gF;MAIqE,qC;QAAA,wBAAgC,S
;MACjG,YAAY,sBAAQ,SAAR,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAAiB,qB;;QAA2B,iBAAa,Q
AAQ,CAAR,I;QAAb,eAAwB,gB;QA9K1E,U;QA8KM,OA9KgB,aAAtB,+DAAsB,EAAa,UAAb,EAAyB,QAAzB,
EA8K4D,WA9K5D,CAAgD,W;;MA8KvE,W;K;IAGJ,kF;MAIuE,qC;QAAA,wBAAgC,S;MACnG,YAAY,sBAAQ
,SAAR,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAAiB,qB;;QAA2B,iBAAa,QAAQ,SAAU,OAAlB,I;Q
AAb,eAAuC,gB;QAvLzF,U;QAuLM,OAvLgB,aAAtB,+DAAsB,EAAa,UAAb,EAAyB,QAAzB,EAuL2E,WAvL3
E,CAAgD,W;;MAuLvE,W;K;IAGJ,oF;MAI2E,qC;QAAA,wBAAgC,S;MACvG,YAAY,0BAAY,SAAZ,C;MACL,
Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAAiB,qB;;QAA2B,iBAAa,QAAQ,SAAU,OAAlB,I;QAAb,eAAuC,gB;Q
AhMzF,U;QAgMM,OAhMgB,aAAtB,+DAAsB,EAAa,UAAb,EAAyB,QAAzB,EAgM2E,WAhM3E,CAAgD,W;;
MAgMvE,W;K;IAGJ,sF;MAIyE,qC;QAAA,wBAAgC,S;MACrG,YAAY,0BAAY,SAAZ,C;MACL,Q;MAAA,IAA
I,UAAS,EAAb,C;QAAA,OAAAiB,qB;;QAA2B,iBAAa,QAAQ,CAAR,I;QAAb,eAAwB,gB;QAzM1E,U;QAyMM,O
AzMgB,aAAtB,+DAAsB,EAAa,UAAb,EAAyB,QAAzB,EAyM4D,WAzM5D,CAAgD,W;;MAyMvE,W;K;IAGJ,q
F;MAI0E,qC;QAAA,wBAAgC,S;MACtG,YAAY,0BAAY,SAAZ,C;MACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAA
A,OAAAiB,qB;;QAlNvB,U;QAkNM,OAlNgB,aAAtB,+DAAsB,EAkNyC,CAlNzC,EAkN4C,KAlN5C,EAkNmD,W
AlNnD,CAAgD,W;;MAkNvE,W;K;IAGJ,uF;MAI4E,qC;QAAA,wBAAgC,S;MACxG,YAAY,0BAAY,SAAZ,C;M
ACL,Q;MAAA,IAAI,UAAS,EAAb,C;QAAA,OAAAiB,qB;;QA3NvB,U;QA2NM,OA3NgB,aAAtB,+DAAsB,EA2N
yC,CA3NzC,EA2N4C,KA3N5C,EA2NmD,WA3NnD,CAAgD,W;;MA2NvE,W;K;+EAOJ,yC;MAQoF,OAAAA,KA
AM,iBAAQ,SAAR,EAAc,WAAd,C;K;+EAE1F,uC;MAOI,OAAAA,KAAM,iBAAQ,SAAR,EAAc,SAAd,C;K;yFAE
V,yC;MAMyF,OAAAA,KAAM,sBAAa,SAAb,EAAmB,WAAnB,C;K;+FAE/F,yB;MAAAA,oC;MAAAA,gC;MAAAA,u
C;QAeW,Q;QAAAA,IApe4C,mBAAS,CAoerD,C;uBAAkB,oBAAU,iCAAK,CAAL,EAAV,E;UAAAA,YNzhBoD,oB
MyhBrB,CNzhBqB,C;UMyhBtE,OLrjBwD,2BAAL,GAAAkB,K;;UKqjBrE,OAAyD,S;QAAhE,W;O;KAfJ,C;iGAkB
A,yB;MAAAA,oC;MAAAA,uC;QAeI,OAtfmD,mBAAS,CAsf5D,GAAyB,UAAU,iCAAK,CAAL,EAAV,CAAmB,W
AAnB,GN3iBoD,oBM2iBV,CN3iBU,CM2iB7E,GAA2E,S;O;KAf/E,C;+EAmBA,4B;MAIsE,OAAAA,KAAM,iBA
AQ,SAAR,C;K;IAE5E,0F;MAKI,IAAK,cAAc,CAAf,IAAsB,aAAa,CAAnC,IAA0C,cAAa,SAAK,OAAL,GAAc,M
AAd,IAAb,CAA1C,IAAiF,eAAc,KAAM,OAAN,GAAe,MAAf,IAAd,CAArF,C;QACI,OAAO,K;;MAGX,iBAAc,C
AAd,UAAsB,MAAtB,U;QACI,IAAI,CAA0B,SAAzB,qBAAK,aAAa,KAAb,IAAL,CAAyB,EAAO,iBAAM,cAAc,
KAAd,IAAN,CAAP,EAAmC,UAAnC,CAA9B,C;UACI,OAAO,K;;MAEf,OAAO,I;K;IAGX,mD;MAG+C,0B;QA
AA,aAAsB,K;MACjE,OAAA,SAAK,OAAL,GAAc,CAAd,IAA2B,SAAR,qBAAK,CAAL,CAAQ,EAAO,IAAP,E
AAa,UAAb,C;K;IAE/B,iD;MAG6C,0B;QAAA,aAAsB,K;MAC/D,OAAA,SAAK,OAAL,GAAc,CAAd,IAAmC,S
AAhB,qBAAK,2BAAL,CAAgB,EAAO,IAAP,EAAa,UAAb,C;K;IAEvC,qD;MAGyD,0B;QAAA,aAAsB,K;MAC3
E,IAAI,CAAC,UAAD,IAAe,6BAAf,IAAiC,0BAArC,C;QACI,OAAY,WAAL,SAAK,EAAW,MAAX,C;;QAEZ,O
AAO,6BAAkB,CAAlB,EAAqB,MAArB,EAA6B,CAA7B,EAAgC,MAAO,OAAvC,EAA+C,UAA/C,C;K;IAGf,iE;
MAG0E,0B;QAAA,aAAsB,K;MAC5F,IAAI,CAAC,UAAD,IAAe,6BAAf,IAAiC,0BAArC,C;QACI,OAAY,aAAL,
SAAK,EAAW,MAAX,EAAmB,UAAnB,C;;QAEZ,OAAO,6BAAkB,UAAlB,EAA8B,MAA9B,EAAsC,CAAtC,EA
AyC,MAAO,OAAhD,EAAwD,UAAxD,C;K;IAGf,mD;MAGuD,0B;QAAA,aAAsB,K;MACzE,IAAI,CAAC,UAA
D,IAAe,6BAAf,IAAiC,0BAArC,C;QACI,OAAY,SAAL,SAAK,EAAS,MAAT,C;;QAEZ,OAAO,6BAAkB,mBAA
S,MAAO,OAAhB,IAAlB,EAA0C,MAA1C,EAAkD,CAAlD,EAAqD,MAAO,OAA5D,EAAoE,UAApE,C;K;IAMf,
wD;MAQ8D,0B;QAAA,aAAsB,K;MAChF,qBAAfjnBO,MAAO,KeinBa,SAAK,OfjnBlB,EeinB0B,KAAM,OfjnBhC,
C;MemnBd,QAAQ,C;MACR,OAAO,IAAI,cAAJ,IAA8B,SAAR,qBAAK,CAAL,CAAQ,EAAO,iBAAM,CAAN,C
AAP,EAA8B,UAA9B,CAArC,C;QACI,a;;MAEJ,IAAS,mBAAL,SAAK,EAAmB,IAAI,CAAJ,IAAnB,CAAL,IAA
wC,mBAAN,KAAM,EAAmB,IAAI,CAAJ,IAAnB,CAA5C,C;QACI,a;;MAEJ,OAAO,8BAAY,CAAZ,EAAe,CAAf
,CAAkB,W;K;IAG7B,wD;MAQ8D,0B;QAAA,aAAsB,K;MAChF,iBAAiB,SAAK,O;MACtB,kBAAkB,KAAM,O;
MACxB,qBAAfxoBO,MAAO,KewoBa,UfxoBb,EewoByB,WfxoBzB,C;Me0oBd,QAAQ,C;MACR,OAAO,IAAI,cAAJ
,IAA+C,SAAzB,qBAAK,aAAa,CAAb,GAAiB,CAAjB,IAAL,CAAyB,EAAO,iBAAM,cAAc,CAAd,GAAkB,CAAl
B,IAAN,CAAP,EAAgD,UAAhD,CAAtD,C;QACI,a;;MAEJ,IAAS,mBAAL,SAAK,EAAmB,aAAa,CAAb,GAAiB,

CAAjB,IAAnB,CAAL,IAAqD,mBAAN,KAAM,EAAmB,cAAc,CAAd,GAAkB,CAAlB,IAAnB,CAAzD,C;QACI,a
;;MAEJ,OAAO,8BAAY,aAAa,CAAb,IAAZ,EAA4B,UAA5B,CAAwC,W;K;IAMnD,8D;MAQqD,0B;QAAA,aAA
kB,C;MAAG,0B;QAAA,aAAsB,K;MAMnE,UAAkB,M;MAL3C,IAAI,CAAC,UAAD,IAAe,KAAM,OAAN,KAAc
,CAA7B,IAAkC,6BAAtC,C;QACI,WAAiB,SAAN,KAAM,C;QACjB,ONjtBwF,kB2G3ME,oBrG45BrE,IqG55BqE
,C3G2MF,EMitB7D,UNjtB6D,C;;MMotBnE,uBAAX,UAAW,EAAc,CAAd,C;MAAkB,oC;kBAA3C,gD;QACI,kB
AAkB,qBAAI,KAAJ,C;QACR,c;;UlC4mXE,U;UAAhB,4BkC5mXQ,KlC4mXR,kB;YAAgB,cAAhB,UkC5mXQ,K
lC4mXR,S;YAAsB,IkC5mXC,SAAH,UlC4mXgB,oBkC5mXhB,CAAG,0BlC4mXD,C;cAAwB,aAAO,I;cAAP,e;;;
UAC9C,aAAO,K;;;QkC7mXH,e;UACI,OAAO,K;;MAEf,OAAO,E;K;IAGX,kE;MASyD,0B;QAAA,aAAkB,2B;M
AAW,0B;QAAA,aAAsB,K;MACxG,IAAI,CAAC,UAAD,IAAe,KAAM,OAAN,KAAc,CAA7B,IAAkC,6BAAtC,C
;QACI,WAAiB,SAAN,KAAM,C;QACjB,ONruB4F,sB2G3MM,oBrGg7BzE,IqGh7ByE,C3G2MN,EMquB7D,UNr
uB6D,C;;kBMyuBhG,iBAAyB,eAAX,UAAW,EAAa,2BAAb,CAAzB,WAAwD,CAAxD,U;QACI,kBAAkB,qBAA
I,KAAJ,C;QACR,c;;UlColXE,Q;UAAhB,wBkCplXQ,KlColXR,gB;YAAgB,cAAhB,UkCplXQ,KlColXR,O;YAAs
B,IkCplXC,SAAH,UlColXgB,oBkCplXhB,CAAG,0BlColXD,C;cAAwB,aAAO,I;cAAP,e;;;UAC9C,aAAO,K;;;Qk
CrlXH,e;UACI,OAAO,K;;MAGf,OAAO,E;K;IAIX,8E;MAA2G,oB;QAAA,OAAgB,K;MAOrG,UAKA,M;MAXlB
,cAAkB,CAAC,IAAL,GACV,aAAW,gBAAX,UAAW,EAAc,CAAd,CAAX,EAAsC,eAAT,QAAS,EAAa,gBAAb,
CAAtC,CADU,GAGV,SAAW,eAAX,UAAW,EAAa,2BAAb,CAAX,EAAmD,gBAAT,QAAS,EAAc,CAAd,CAAn
D,C;MAEJ,IAAI,iCAAkB,yBAAtB,C;QACkB,yB;QAAd,OAAc,cAAd,C;UAAc,uB;UACV,IAAU,cAAN,KAAM,
EAAc,CAAd,EAAiB,SAAjB,EAAuB,KAAvB,EAA8B,KAAM,OAApC,EAA4C,UAA5C,CAAV,C;YACI,OAAO,
K;;;QAGD,2B;QAAd,OAAc,gBAAd,C;UAAc,2B;UACV,IAAU,kBAAN,KAAM,EAAkB,CAAlB,EAAqB,SAArB,
EAA2B,OAA3B,EAAkC,KAAM,OAAxC,EAAgD,UAAhD,CAAV,C;YACI,OAAO,O;;;MAGnB,OAAO,E;K;IAG
X,qE;MAUsB,UAMA,M;MAflB,IAAI,CAAC,UAAD,IAAe,OAAQ,KAAR,KAAgB,CAAnC,C;QACI,aAAqB,UA
AR,OAAQ,C;QACrB,YAAgB,CAAC,IAAL,GAAW,sBAAQ,MAAR,EAAgB,UAAhB,CAAX,GAA4C,0BAAY,M
AAZ,EAAoB,UAApB,C;QACxD,OAAW,QAAQ,CAAZ,GAAe,IAAf,GAAyB,UAAS,MAAT,C;;MAGpC,cAAkB,
CAAC,IAAL,GAAW,aAAW,gBAAX,UAAW,EAAc,CAAd,CAAX,EAA6B,gBAA7B,CAAX,GAAoD,SAAW,eA
AX,UAAW,EAAa,2BAAb,CAAX,EAA0C,CAA1C,C;MAElE,IAAI,6BAAJ,C;QACkB,yB;oBAAd,OAAc,cAAd,C
;UAAc,yB;UACmB,sB;;Yb3sBrB,U;YAAA,Sa2sBa,Ob3sBb,W;YAAhB,OAAgB,gBAAhB,C;cAAgB,2B;cAAM,I
a2sBgC,cb3sBlB,Oa2sBkB,EAAc,CAAd,sBb3sBlB,Oa2sBmD,OAAjC,ab3sBhC,C;gBAAwB,qBAAO,O;gBAAP,u
B;;;YAC9C,qBAAO,I;;;Ua0sBC,uC;UACA,IAAI,sBAAJ,C;YACI,OAAO,YAAS,cAAT,C;;;QAGD,2B;oBAAd,O
AAc,gBAAd,C;UAAc,2B;UACmB,wB;;YbjtBrB,U;YAAA,SaitBa,ObjtBb,W;YAAhB,OAAgB,gBAAhB,C;cAAg
B,6B;cAAM,IaitBgC,kBbjtBlB,SaitBkB,EAAkB,CAAlB,sBbjtBlB,SaitBuD,OAArC,abjtBhC,C;gBAAwB,uBAAO,
S;gBAAP,uB;;;YAC9C,uBAAO,I;;;UagtBC,2C;UACA,IAAI,wBAAJ,C;YACI,OAAO,YAAS,gBAAT,C;;;MAInB,
OAAO,I;K;IAGX,iE;MAY+D,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MACtG,4BAAU,OAAV,EAAm
B,UAAnB,EAA+B,UAA/B,EAAkD,KAAlD,C;K;IAEJ,mE;MAYmE,0B;QAAA,aAAkB,2B;MAAW,0B;QAAA,a
AAsB,K;MAClH,4BAAU,OAAV,EAAmB,UAAnB,EAA+B,UAA/B,EAAkD,IAAlD,C;K;IAEJ,kE;MAWgE,0B;Q
AAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MACvG,gB;MAAA,8CAAU,OAAV,EAAmB,UAAnB,EAA+B,UA
A/B,EAAkD,KAAlD,mDAAmE,E;K;IAEvE,sE;MAYoE,0B;QAAA,aAAkB,2B;MAAW,0B;QAAA,aAAsB,K;MA
CnH,gB;MAAA,8CAAU,OAAV,EAAmB,UAAnB,EAA+B,UAA/B,EAAkD,IAAlD,mDAAkE,E;K;IAKtE,6D;MA
M4C,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MACnF,OAAW,cAAc,gCAAzB,GACI,sBAAW,mBAAY
,IAAZ,CAAX,EAA8B,UAA9B,EAA0C,UAA1C,CADJ,GNz2B4F,kB2G3ME,oBrGujC5E,IqGvjC4E,C3G2MF,E
M42BpE,UN52BoE,C;K;IM+2BhG,+D;MAQgD,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MACvF,OAA
W,cAAc,gCAAzB,GACI,sBAAQ,MAAR,EAAgB,UAAhB,EAA4B,gBAA5B,EAAoC,UAApC,CADJ,GNx3B4F,k
BM23B1E,MN33B0E,EM23BlE,UN33BkE,C;K;IM83BhG,iE;MAQgD,0B;QAAA,aAAkB,2B;MAAW,0B;QAAA,
aAAsB,K;MAC/F,OAAW,cAAc,gCAAzB,GACI,0BAAe,mBAAY,IAAZ,CAAf,EAAkC,UAAlC,EAA8C,UAA9C,C,
CADJ,GNp4BgG,sB2G3MM,oBrGklChF,IqGllCgF,C3G2MN,EMu4BpE,UNv4BoE,C;K;IM04BpG,mE;MAQoD,0
B;QAAA,aAAkB,2B;MAAW,0B;QAAA,aAAsB,K;MACnG,OAAW,cAAc,gCAAzB,GACI,sBAAQ,MAAR,EAAAg
B,UAAhB,EAA4B,CAA5B,EAA+B,UAA/B,EAAkD,IAAlD,CADJ,GNn5BgG,sBMs5B1E,MNt5B0E,EMs5BlE,U
Nt5BkE,C;K;IMy5BpG,mD;MAM+D,0B;QAAA,aAAsB,K;MACjF,OAAI,yBAAJ,GACI,sBAAQ,KAAR,UAA4B,
UAA5B,KAA2C,CAD/C,GAGI,sBAAQ,KAAR,EAAe,CAAf,EAAkB,gBAAlB,EAA0B,UAA1B,KAAyC,C;K;IAIj

D,kD;MAMsD,0B;QAAA,aAAsB,K;MACxE,6BAAQ,IAAR,UAA2B,UAA3B,KAA0C,C;K;kFAE9C,4B;MAI0E,
OAAA,KAAM,yBAAgB,SAAhB,C;K;IAM3C,yE;MACjC,oB;MACA,8B;MACA,oB;MACA,kC;K;IAG8C,sF;MA
AA,gE;MAC1C,iBAAqB,E;MACrB,yBAAwC,WAAX,yCAAW,EAAS,CAAT,EAAY,oCAAM,OAAlB,C;MACxC
,uBAA2B,sB;MAC3B,gBAA0B,I;MAC1B,eAAmB,C;K;0EAEnB,Y;MACI,IAAI,uBAAkB,CAAtB,C;QACI,iBAA
Y,C;QACZ,gBAAW,I;;QAEX,IAAI,4CAAQ,CAAR,IAAa,uDAAa,yCAA1B,IAAmC,uBAAkB,yCAAM,OAA/D,C
;UACI,gBAAW,qCAAyB,iBAAN,yCAAM,CAAzB,C;UACX,uBAAkB,E;;UAElB,YAAkB,iDAAN,yCAAM,EAA
a,oBAAb,C;UAClB,IAAI,SAAS,IAAb,C;YACI,gBAAW,qCAAyB,iBAAN,yCAAM,CAAzB,C;YACX,uBAAkB,E
;;YAElB,IAAK,QAAiB,KAAjB,aAAL,EAAY,SAAU,KAAV,a;YACZ,gBAAW,gCAAwB,KAAxB,C;YACX,yBA
AoB,QAAQ,MAAR,I;YACpB,uBAAkB,0BAAwB,WAAU,CAAd,GAAiB,CAAjB,GAAwB,CAA5C,K;;;QAG1B,i
BAAY,C;;K;oEAIpB,Y;MAKiB,Q;MAJb,IAAI,mBAAa,EAAjB,C;QACI,iB;MACJ,IAAI,mBAAa,CAAjB,C;QACI
,MAAM,6B;MACV,aAAa,mE;MAEb,gBAAW,I;MACX,iBAAY,E;MACZ,OAAO,M;K;uEAGX,Y;MACI,IAAI,m
BAAa,EAAjB,C;QACI,iB;MACJ,OAAO,mBAAa,C;K;;iDA9C5B,Y;MAA8C,+D;K;;IAgEU,0E;MAAA,0C;QhB1
mCjD,SgB2mCH,sBAAW,kBAAX,EAAuB,YAAvB,EAAkD,kBAAlD,C;QAAA,OAAwE,KAAK,CAAT,GAAY,I
AAZ,GAAsB,OAAM,CAAN,C;O;K;IAdlG,iF;MAUkE,0B;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MAAO
,qB;QAAA,QAAa,C;MAC7H,wBAAwB,KAAxB,C;MAEA,OAAO,4BAAwB,SAAxB,EAA8B,UAA9B,EAA0C,K
AA1C,EAAiD,gDAAjD,C;K;IAwBiD,gF;MAAA,0C;QAAkB,Q;QAAA,oCAAU,sBAAV,EAA0B,YAA1B,EAAq
D,kBAArD,EAAwE,KAAxE,aAAsF,GAAG,UAAH,EAAe,WAAO,OAAtB,CAAtF,O;O;K;IAlB9E,mF;MAc0E,0B
;QAAA,aAAkB,C;MAAG,0B;QAAA,aAAsB,K;MAAO,qB;QAAA,QAAa,C;MACrI,wBAAwB,KAAxB,C;MACA
,qBAAgC,OAAX,UAAW,C;MAEhC,OAAO,4BAAwB,SAAxB,EAA8B,UAA9B,EAA0C,KAA1C,EAAiD,sDAAj
D,C;K;IAIX,wC;MnBltCI,IAAI,EmBmtCI,SAAS,CnBntCb,CAAJ,C;QACI,cmBktCkB,8C;QnBjtClB,MAAM,gCA
AyB,OAAQ,WAAjC,C;;K;ImBkuCgE,sD;MAAA,qB;QAAE,yCAAU,EAAV,C;O;K;IAZhF,mE;MAWmE,0B;QA
AA,aAAsB,K;MAAO,qB;QAAA,QAAa,C;MACzG,OAAsE,OAAtE,+BAAkB,UAAlB,UAA2C,UAA3C,EAA+D,
KAA/D,CAAsE,EAAI,iCAAJ,C;K;IAE1E,yD;MAWyD,0B;QAAA,aAAsB,K;MAAO,qB;QAAA,QAAa,C;MAC/F,
IAAI,UAAW,OAAX,KAAmB,CAAvB,C;QACI,gBAAgB,WAAW,CAAX,C;QAChB,IAAI,EAAC,SAh/BuC,YAA
U,CAg/BlD,CAAJ,C;UACI,OAAO,mBAAM,SAAN,EAAiB,UAAjB,EAA6B,KAA7B,C;;;MAI2E,kBAAb,cAAtE,
+BAAkB,UAAlB,UAA2C,UAA3C,EAA+D,KAA/D,CAAsE,C;MbgPtE,kBAAM,iBAAa,qCAAwB,EAAxB,CAAb
,C;MAuEA,Q;MAAA,6B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT,WAAY,WaxTgF,uBbwTlE,IaxTkE,CbwThF,C;
;MaxThB,ObyTO,W;K;Ia9SmE,wD;MAAA,qB;QAAE,yCAAU,EAAV,C;O;K;IARhF,qE;MAOiE,0B;QAAA,aAA
sB,K;MAAO,qB;QAAA,QAAa,C;MACvG,OAAsE,OAAtE,6BAAkB,UAAlB,UAA2C,UAA3C,EAA+D,KAA/D,C
AAsE,EAAI,mCAAJ,C;K;IAE1E,2D;MAOuD,0B;QAAA,aAAsB,K;MAAO,qB;QAAA,QAAa,C;MAC7F,IAAI,U
AAW,OAAX,KAAmB,CAAvB,C;QACI,OAAO,mBAAoB,oBAAd,WAAW,CAAX,CAAc,CAApB,EAAgC,UAAh
C,EAA4C,KAA5C,C;;MAG+E,kBAAb,cAAtE,6BAAkB,UAAlB,UAA2C,UAA3C,EAA+D,KAA/D,CAAsE,C;Mb
uNtE,kBAAM,iBAAa,qCAAwB,EAAxB,CAAb,C;MAuEA,Q;MAAA,6B;MAAb,OAAa,cAAb,C;QAAa,sB;QACT
,WAAY,Wa/RgF,uBb+RlE,Ia/RkE,Cb+RhF,C;;Ma/RhB,ObgSO,W;K;Ia7RX,0D;MASI,wBAAwB,KAAxB,C;MA
EA,oBAAoB,C;MACpB,gBAAgB,sBAAQ,SAAR,EAAmB,aAAnB,EAAkC,UAAlC,C;MAChB,IAAI,cAAa,EAAb
,IAAmB,UAAS,CAAhC,C;QACI,OAAO,OAAO,SAAK,WAAZ,C;;MAGX,gBAAgB,QAAQ,C;MACxB,aAAa,iB
AAsB,SAAJ,GAAqB,eAAN,KAAM,EAAa,EAAb,CAArB,GAA2C,EAA7D,C;;QAET,MAAO,WA36B6E,8BA26
B/D,aA36B+D,EA26BhD,SA36BgD,CAAkC,WA26B/G,C;QACP,gBAAgB,YAAY,SAAU,OAAtB,I;QAEhB,IAA
I,aAAa,MAAO,KAAP,MAAe,QAAQ,CAAR,IAAf,CAAjB,C;UAA2C,K;QAC3C,YAAY,sBAAQ,SAAR,EAAmB,
aAAnB,EAAkC,UAAlC,C;;MACP,sBAAa,EAAb,C;MAET,MAAO,WAl7BiF,8BAk7BnE,aAl7BmE,EAk7BpD,gB
Al7BoD,CAAkC,WAk7BnH,C;MACP,OAAO,M;K;2EAGX,mC;MAOmD,qB;QAAA,QAAa,C;MAAmB,OAAA,
KAAM,eAAM,SAAN,EAAY,KAAZ,C;K;+FAEzF,mC;MAU6D,qB;QAAA,QAAa,C;MAAuB,OAAA,KAAM,yB
AAgB,SAAhB,EAAsB,KAAtB,C;K;IAEvG,iC;MAK2D,mCAAgB,MAAhB,EAAwB,IAAxB,EAA8B,IAA9B,E;K;
IAE3D,0B;MAKgD,OAAe,UAAf,uBAAe,C;K;IAqB/D,uD;MAQsB,Q;MAPlB,IAAI,iCAAkB,yBAAtB,C;QACI,O
AAY,SAAL,SAAK,EAAAO,KAAP,EAA2B,IAA3B,C;;MAGhB,IAAI,cAAS,KAAb,C;QAAAoB,OAAO,I;MAC3B,I
AAI,qBAAgB,aAAhB,IAAAiC,SAAK,OAAL,KAAAe,KAAAM,OAA1D,C;QAAAkE,OAAO,K;MAEvD,uB;MAAlB,aA
AU,CAAAV,gB;QACI,IAAI,CAAAS,SAAR,qBAAAK,CAAAL,CAAAQ,EAAAO,iBAAAM,CAAAN,CAAAP,EAAA8B,IAA9B,C
AAb,C;UACI,OAAO,K;;;MAIf,OAAO,I;K;IAGX,6C;MAQsB,Q;MAPlB,IAAI,iCAAkB,yBAAtB,C;QACI,OAAO,

kBAAQ,KAAR,C;;MAGX,IAAI,cAAS,KAAb,C;QAAoB,OAAO,I;MAC3B,IAAI,qBAAgB,aAAhB,IAAiC,SAAK,
OAAL,KAAe,KAAM,OAA1D,C;QAAkE,OAAO,K;MAEvD,uB;MAAlB,aAAU,CAAV,gB;QACI,IAAI,qBAAK,C
AAL,MAAW,iBAAM,CAAN,CAAf,C;UACI,OAAO,K;;;MAIf,OAAO,I;K;IAGX,oC;MAU+C,QAAM,SAAN,C;a
AC3C,M;UAD2C,OACjC,I;aACV,O;UAF2C,OAEhC,K;;UACH,MAAM,gCAAyB,mDAAgD,SAAzE,C;;K;IAGlB
,0C;MAUsD,QAAM,SAAN,C;aAClD,M;UADkD,OACxC,I;aACV,O;UAFkD,OAEvC,K;;UAFuC,OAG1C,I;;K;Im
Lr8CZ,sB;MAAA,0B;MAII,aAC+B,e;MAC/B,cACgC,e;MAChC,WAC6B,e;MAC7B,YAC8B,e;MAC9B,eACiC,e;
MACjC,YAC8B,gB;MAC9B,aAC+B,gB;MAC/B,YAC8B,gB;MAC9B,aAC+B,gB;MAC/B,eACiC,gB;MACjC,iB
ACmC,gB;MACnC,qBAEuC,gB;MACvC,sBAEwC,gB;MACxC,kBACoC,gB;MACpC,cACgC,gB;MAChC,iBAC
mC,gB;MACnC,iBACmC,gB;MACnC,iBACmC,gB;MACnC,YAC8B,gB;MAC9B,aAC+B,iB;MAC/B,aAC+B,iB;
MAC/B,uBACyC,iB;MACzC,wBAC0C,iB;MAC1C,sBACwC,iB;MACxC,uBACyC,iB;MACzC,wBAC0C,iB;MA
C1C,sBACwC,iB;MACxC,cACgC,iB;MAChC,oBACsC,iB;MACtC,cACgC,iB;MAChC,gBACkC,iB;MAClC,aAC
+B,iB;MAC/B,mBACqC,iB;MACrC,YAC8B,iB;MAC9B,UAC4B,iB;MAC5B,mBACqC,iB;MACrC,gBACkC,iB;
MAClC,mBACqC,iB;MACrC,sBACwC,iB;MAExC,sBAGwC,gB;MAExC,uBAGyC,gB;K;;;IA7F7C,kC;MAAA,i
C;QAAA,gB;;MAAA,0B;K;;;;;;;;;2FCwE0C,Y;MAAQ,oCAAa,IAAb,C;K;IAiBpB,yC;MAAqB,kB;K;mIAC3C,Y;
MACmD,OAAA,UAAM,YAAN,aAAkB,CAAlB,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAl
B,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAlB,C;K;mIACnD,Y;MACmD,OAAA,UAAM,Y
AAN,aAAkB,CAAlB,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAlB,C;K;mIACnD,Y;MACm
D,OAAA,UAAM,YAAN,aAAkB,CAAlB,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAlB,C;K;
mIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,CAAlB,C;K;mIACnD,Y;MACmD,OAAA,UAAM,YAAN,a
AAkB,CAAlB,C;K;qIACnD,Y;MACmD,OAAA,UAAM,YAAN,aAAkB,EAAlB,C;K;gDAEnD,Y;MAMoC,OAAA
,UAAM,YAAY,iBAAQ,CAAR,EAAW,UAAM,YAAY,KAA7B,C;K;;;6EvElH9D,yB;MAAA,iD;MAAA,4B;QAI4
C,kBAAM,SAAN,C;O;KAJ5C,C;+EAMA,yB;MAAA,gD;MAAA,oC;QAI+D,kBAAM,SAAN,EAAY,MAAZ,C;O;
KAJ/D,C;+EAMA,yB;MAAA,oC;MAAA,qC;QAIqE,sBAAM,SAAN,EAAY,OAAZ,C;O;KAJrE,C;IrIY4B,4B;MA
mBxB,gC;MAnB6C,0B;MAW7B,UAEA,MAFA,EAGA,M;MALZ,IgIjC8D,IhIiC9D,C;QACI,IAAI,kBAAJ,C;UA
CQ,mB;UAAJ,IAAI,sEAAsB,SAAtB,EAAJ,C;YAAqC,MAAM,sBAAiB,YAAF,+CAAf,C;;UAEvC,qB;UAAJ,IA
AI,0EAAuB,UAAvB,EAAJ,C;YAAuC,MAAM,sBAAiB,YAAF,gDAAf,C;UACzC,qB;UAAJ,IAAI,kEAA+B,mBA
A/B,CAAJ,C;YAAwD,MAAM,sBAAiB,YAAF,mCAAf,C;;;K;mFAZlD,Y;MAAQ,kCAAa,CAAb,C;K;+FACU,Y;
MAAQ,OAAA,eAAS,QAAT,GAAqB,C;K;qCACvE,Y;MAA0B,QADwB,eAAS,QAAT,GAAqB,CAC7C,MAAqB,
C;K;sCAC/C,Y;MAA2B,QAFuB,eAAS,QAAT,GAAqB,CAE5C,MAAqB,C;K;yFACxB,Y;MAAQ,OAAI,kBAAJ,
mF;K;IAahC,8B;MAAA,kC;MACI,YAC4B,gB;MAE5B,gBACgC,iBAAiB,UAAjB,C;MAChC,4BAAsC,uC;K;mD
AEtC,yC;MAGI,2BAAoB,KAApB,EAA2B,UAA3B,EAAuC,UAAvC,C;K;iJAM8B,yB;MAAA,6C;MAAA,iD;MA
AA,4B;QAAQ,sD;O;KAAR,C;iJAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,sD;O;KAAR,C;iJAUE,yB;MA
AA,6C;MAAA,iD;MAAA,4B;QAAQ,sD;O;KAAR,C;mJAKF,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,uD;O;
KAAR,C;mJAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,uD;O;KAAR,C;mJAUE,yB;MAAA,6C;MAAA,iD;
MAAA,4B;QAAQ,uD;O;KAAR,C;mJAKH,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,uD;O;KAAR,C;mJAIC,
yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,uD;O;KAAR,C;mJAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAA
Q,uD;O;KAAR,C;yIAKR,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;yIAIC,yB;MAAA,6C;MA
AA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;yIAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;yI
AKH,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;yIAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;Q
AAQ,kD;O;KAAR,C;yIAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,kD;O;KAAR,C;qIAKL,yB;MAAA,6C;
MAAA,iD;MAAA,4B;QAAQ,gD;O;KAAR,C;qIAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,gD;O;KAAR,C
;qIAUE,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,gD;O;KAAR,C;mIAKJ,yB;MAAA,6C;MAAA,iD;MAAA,4
B;QAAQ,+C;O;KAAR,C;mIAIC,yB;MAAA,6C;MAAA,iD;MAAA,4B;QAAQ,+C;O;KAAR,C;mIAUE,yB;MAA
A,6C;MAAA,iD;MAAA,4B;QAAQ,+C;O;KAAR,C;uDAK9B,iB;MAK+C,OAAM,WAAN,KAAM,yC;K;uDAErD
,iB;MAKgD,OAAM,aAAN,KAAM,yC;K;uDAEtD,iB;MASkD,OAAM,aAAN,KAAM,yC;K;wDAGxD,iB;MAKg
D,OAAM,WAAN,KAAM,0C;K;wDAEtD,iB;MAKiD,OAAM,aAAN,KAAM,0C;K;wDAEvD,iB;MASmD,OAAM
,aAAN,KAAM,0C;K;wDAGzD,iB;MAKgD,OAAM,WAAN,KAAM,0C;K;wDAEtD,iB;MAKiD,OAAM,aAAN,K
AAM,0C;K;wDAEvD,iB;MASmD,OAAM,aAAN,KAAM,0C;K;mDAGzD,iB;MAK2C,OAAM,WAAN,KAAM,q

C;K;mDAEjD,iB;MAK4C,OAAM,aAAN,KAAM,qC;K;mDAElD,iB;MAS8C,OAAM,aAAN,KAAM,qC;K;mDAG
pD,iB;MAK2C,OAAM,WAAN,KAAM,qC;K;mDAEjD,iB;MAK4C,OAAM,aAAN,KAAM,qC;K;mDAElD,iB;MA
S8C,OAAM,aAAN,KAAM,qC;K;iDAGpD,iB;MAKyC,OAAM,WAAN,KAAM,mC;K;iDAE/C,iB;MAK0C,OAA
M,aAAN,KAAM,mC;K;iDAEhD,iB;MAS4C,OAAM,aAAN,KAAM,mC;K;gDAGlD,iB;MAKwC,OAAM,WAAN,
KAAM,kC;K;gDAE9C,iB;MAKyC,OAAM,aAAN,KAAM,kC;K;gDAE/C,iB;MAS2C,OAAM,aAAN,KAAM,kC;
K;iDAEjD,iB;;QAY4C,OACxC,cAAc,KAAd,EAAAiC,KAAjC,C;;QACF,+C;UACE,MAAM,6BAAyB,sCAAmC,K
AAnC,OAAzB,EAAsE,CAAtE,C;;UAHkC,O;;K;0DAM5C,iB;;QAeqD,OACjD,cAAc,KAAd,EAAAiC,IAAjC,C;;Q
ACF,+C;UACE,MAAM,6BAAyB,0CAAuC,KAAvC,OAAzB,EAA0E,CAA1E,C;;UAH2C,O;;K;uDAMrD,iB;;QA
WmD,OAC/C,cAAc,KAAd,EAAAiC,KAAjC,C;;QACF,+C;UAFiD,OAG/C,I;;UAH+C,O;;K;gEAMnD,iB;;QAO4D,
OACxD,cAAc,KAAd,EAAAiC,IAAjC,C;;QACF,+C;UAF0D,OAGxD,I;;UAHwD,O;;K;;;IA1YhE,0C;MAAA,yC;Q
AAA,wB;;MAAA,kC;K;oCAmZA,Y;MAC6C,kBAAY,YAAD,aAAX,EApaK,eAAS,QAAT,GAAqB,CAoa1B,C;K
;qCAE7C,iB;MAiBW,Q;MATH,IAAA,IAAK,aAAL,C;QACI,IAAI,KAAM,WAAN,IAAqB,IAAK,WAAL,KAAkB
,KAAM,WAAxB,gBAAoC,CAA7D,C;UACI,OAAO,I;;UAEP,MAAM,gCAAyB,2EAAzB,C;WAEd,IAAA,KAAM
,aAAN,C;QAAsB,OAAO,K;MAI7B,KAxb0C,eAAS,QAAT,GAAqB,CAwb/D,OAA0B,KAxbgB,WAAS,QAAT,G
AAqB,CAwb/D,E;QACI,aAAa,IAAK,QAAL,KAAa,KAAM,QAAnB,C;QAET,uB;UACI,iCAA0B,MAA1B,C;;UA
EA,kCAA2B,MAA3B,C;aAGZ,IAAA,IAAK,eAAL,C;QACI,mCAAqB,IAAK,QAA1B,EAAiC,KAAM,QAAvC,C;
;QAEA,mCAAqB,KAAM,QAA3B,EAAkC,IAAK,QAAvC,C;MAbR,W;K;gDAiBJ,kC;MAGW,Q;MAFP,kBAAkB
,cAAc,UAAd,C;MAClB,mBAAmB,eAAa,WAAb,C;MACZ,IAAI,8EAAsC,mBAAtC,CAAJ,C;QACH,yBAAyB,o
BAAa,cAAc,WAAd,CAAb,C;QACzB,uBAAgB,cAAc,YAAd,MAA8B,kBAA9B,CAAhB,C;;QAEA,wBAA8B,W
AAb,YAAa,yBAAsB,UAAtB,CAA9B,C;;MAJJ,W;K;sCAQJ,iB;MAMuD,wBAAS,KAAD,aAAR,C;K;uCAEvD,iB
;MAQe,UAUJ,M;MAXP,IAAI,iBAAJ,C;QAEQ,cAAS,CAAT,C;UAAc,MAAM,gCAAyB,mEAAzB,C;aACpB,YA
AQ,CAAR,C;UAAa,W;;UACL,OAAC,IAAD,a;QAHZ,W;;MAMJ,IAAI,UAAS,CAAb,C;QAAgB,OAAO,qC;MAE
vB,YAAY,Y;MACZ,aAAa,mCAAQ,KAAR,E;MACN,IAAI,kBAAJ,C;QACH,IAAI,yEAAJ,C;UAEI,yBAAgB,MA
AhB,C;;UAEA,IAAI,sCAAS,KAAT,IAAkB,KAAlB,CAAJ,C;YACI,mCAA0B,MAA1B,C;;YAEA,aAAa,cAAc,K
AAd,C;YACb,eAAe,eAAQ,cAAc,MAAd,CAAR,C;YACf,mBAAmB,oCAAS,KAAT,E;YACnB,kBAAkB,iBAAe,c
AAc,sCAAW,KAAX,EAAd,CAAf,C;YAClB,IAAI,4CAAe,KAAf,IAAwB,MAAxB,KAAkC,gBAAgB,YAAhB,gB
AAgC,CAAtE,C;cACI,0BAA6B,WAAZ,WAAY,EAAS,8BAAa,UAAb,CAAT,CAA7B,C;;cAEA,SAAI,YAAM,W
AAN,KAAM,CAAN,EAAmB,WAAN,KAAM,CAAnB,IAA0B,CAA9B,GAAiC,yCAAjC,GAA+C,qD;;;;;QAK3D,
IAAI,sCAAS,KAAT,IAAkB,KAAlB,CAAJ,C;UACI,0BAAwB,WAAP,MAAO,EAAS,8BAAa,UAAb,CAAT,CAA
xB,C;;UAEA,SAAI,YAAM,WAAN,KAAM,CAAN,EAAmB,WAAN,KAAM,CAAnB,IAA0B,CAA9B,GAAiC,yC
AAjC,GAA+C,qD;;;MAvBvD,a;K;uCA4BJ,iB;MASI,eAAqB,WAAN,KAAM,C;MACrB,IAAa,QAAT,KAAuB,K
AA3B,C;QACI,OAAO,mBAAM,QAAN,C;;MAGX,WAAW,kB;MACX,aAAa,sBAAS,IAAT,IAAiB,K;MAC9B,O
AAc,aAAP,MAAO,EAAW,IAAX,C;K;qCAGlB,iB;MAQe,Q;MADX,IAAI,UAAS,CAAb,C;QAEQ,sB;UAAgB,gD
;aAChB,sB;UAAgB,4D;;UACR,MAAM,gCAAyB,4DAAzB,C;QAHlB,W;;MAMJ,IAAI,kBAAJ,C;QACI,OAAO,g
BAAgB,qCAAQ,KAAR,EAAhB,C;;QAEP,IAAI,iBAAJ,C;UACI,OAAO,mBAAa,WAAN,KAAM,CAAb,C;QAEX
,aAAa,qCAAQ,KAAR,E;QAEb,IAAI,kEAAgC,mBAAhC,CAAJ,C;UACI,UAAU,cAAc,sBAAS,oCAAS,KAAT,E
AAT,CAAd,0BAA0C,KAA1C,E;UACV,OAAO,gBAAgB,cAAc,MAAd,MAAwB,GAAxB,CAAhB,C;;QAEX,OA
AO,iBAAiB,MAAjB,C;;K;qCAIf,iB;MAOI,eAAqB,WAAN,KAAM,C;MACrB,IAAa,QAAT,KAAuB,KAAvB,IA
AgC,aAAY,CAAhD,C;QACI,OAAO,iBAAI,QAAJ,C;;MAGX,WAAW,kB;MACX,aAAa,sBAAS,IAAT,IAAiB,K;
MAC9B,OAAc,aAAP,MAAO,EAAW,IAAX,C;K;oCAGlB,iB;MAEI,kBAAkB,SAAM,IAAK,cAAX,EAAwB,KA
AM,cAA9B,C;MAClB,OAAO,IAAK,kBAAS,WAAT,CAAL,GAA6B,KAAM,kBAAS,WAAT,C;K;oCAG9C,Y;M
ACmC,oCAAW,C;K;oCAE9C,Y;MACmC,oCAAW,C;K;oCAE9C,Y;MACmC,+BAAY,yCAAS,WAArB,KAAiC,
wBAAY,qDAAa,WAAzB,C;K;kCAEpE,Y;MACiC,QAAC,iB;K;yFAGC,Y;MAAQ,OAAI,iBAAJ,GAAmB,IAAD,
aAAlB,GAA6B,I;K;yCAExE,iB;MACI,kBAAkB,IAAK,WAAL,KAAkB,KAAM,WAAxB,C;MAClB,IAAI,yBAAc
,CAAd,IAAmB,CAAA,WAAY,QAAZ,GAAwB,CAAxB,MAA6B,CApD,C;QACI,OAAO,IAAK,WAAS,iBAAU
,KAAM,WAAhB,C;MAEzB,QAAQ,CArmBsC,eAAS,QAAT,GAAqB,CAqmB3D,KAAyB,KArmBa,WAAS,QAA
T,GAAqB,CAqmB3D,K;MACR,OAAW,iBAAJ,GAAkB,CAAC,CAAD,IAAlB,GAA0B,C;K;uHAMrC,kB;MAeI,
OAAO,OAAO,gBAAP,EAAoB,mBAApB,EAAoC,qBAApC,EAAsD,qBAAtD,EAAwE,yBAAxE,C;K;uHAGX,kB

;MAcI,OAAO,OAAO,iBAAP,EAAqB,qBAArB,EAAuC,qBAAvC,EAAyD,yBAAzD,C;K;uHAGX,kB;MAaI,OAA
O,OAAO,mBAAP,EAAuB,qBAAvB,EAAyC,yBAAzC,C;K;uHAGX,kB;MAYI,OAAO,OAAO,mBAAP,EAAuB,y
BAAvB,C;K;0FAKP,Y;MAAQ,OAAI,iBAAJ,GAAkB,CAAlB,GAA0B,6CAAe,EAAf,EAAmB,Q;K;4FAIrD,Y;M
AAQ,OAAI,iBAAJ,GAAkB,CAAlB,GAA0B,+CAAiB,EAAjB,EAAqB,Q;K;4FAIvD,Y;MAAQ,OAAI,iBAAJ,GA
AkB,CAAlB,GAA0B,+CAAiB,EAAjB,EAAqB,Q;K;gGAIvD,Y;MACI,sB;QADI,OACY,C;WAChB,wB;QAFI,OA
EY,cAAc,wCAAQ,IAAR,EAAd,CAA6B,Q;;QAFzC,OAGK,wCAAQ,UAAR,EAAuB,Q;K;0CAMxC,gB;MAQiB,
UAAN,M;MAAM,sB;MACT,iBAAA,yCAAS,WAAT,E;QAA4B,SAAP,wCAAO,kB;WAC5B,iBAAA,qDAAa,W
AAb,E;QAAgC,SAAP,wCAAO,kB;;QAG5B,6BAAoB,YAAM,WAA1B,EAAsC,kBAAtC,EAAmD,IAAnD,C;;MA
LR,a;K;wCAUJ,gB;MAUiB,UAAN,M;MAAM,sB;MACT,iBAAA,yCAAS,WAAT,E;;WACA,iBAAA,qDAAa,W
AAb,E;;;QACQ,+BAAoB,YAApB,EAA2B,kBAA3B,EAAwC,IAAxC,C;MAHZ,a;K;uCAOJ,gB;MAUI,OAAa,WA
Ab,oBAAO,IAAP,CAAa,4BAAyD,Q;K;kFAMhD,Y;MAAQ,6D;K;mFAMP,Y;MAAQ,8D;K;qFAMN,Y;MAAQ,g
E;K;qFAMR,Y;MAAQ,gE;K;0FAMH,Y;MAAQ,qE;K;0FAMR,Y;MAAQ,qE;K;yFAMT,Y;MAAQ,oE;K;uFASrC,
Y;MAAQ,2D;K;wFAQR,Y;MAAQ,4D;K;0FAQR,Y;MAAQ,8D;K;0FAQR,Y;MAAQ,8D;K;+FAQR,Y;MACI,OA
AW,uBAAgB,eAApB,GAAgC,YAAhC,GAA2C,4D;K;+FAatD,Y;MAAQ,mE;K;8FAYR,Y;MAEW,Q;MADP,YA
AY,Y;MAER,uB;QAAe,Y;WACf,8C;;WACA,+C;;;QACQ,qBAAc,KAAd,C;MAJZ,W;K;2CAUR,Y;MAUuC,8B;
K;4CAEvC,Y;MAUwC,+B;K;kCAExC,Y;MAuBwC,Q;MAAA,sB;MACpC,qB;QAD8B,OACxB,I;WACN,iBAAA
,yCAAS,WAAT,E;QAF8B,OAET,U;WACrB,iBAAA,qDAAa,WAAb,E;QAH8B,OAGL,W;;QAErB,iBAAiB,iB;Q
4H7iBF,gBAAhB,sB;Q5H+iBK,e;UAAgB,yBAAO,EAAP,C;QACF,YAAd,kB;QAvSD,WAAO,iB;QAAP,YAAoB
,oB;QAApB,cAAoC,sB;QAApC,cAAsD,sB;QAAtD,kBAAwE,0B;QA+S/D,0B;QAPJ,cAAc,iB;QACd,eAAe,UAA
S,C;QACxB,iBAAiB,YAAW,C;QAC5B,iBAAiB,YAAW,CAAX,IAAgB,gBAAe,C;QAChD,iBAAiB,C;QACjB,IA
AI,OAAJ,C;UACI,yBAAO,IAAP,CAAa,gBAAO,GAAP,C;UACb,+B;;QAEJ,IAAI,aAAa,YAAY,cAAc,UAA1B,C
AAb,CAAJ,C;UACI,IAAI,6DAAe,CAAnB,C;YAAsB,yBAAO,EAAP,C;UACtB,yBAAO,KAAP,CAAc,gBAAO,G
AAP,C;;QAElB,IAAI,eAAe,eAAe,YAAY,OAA3B,CAAf,CAAJ,C;UACI,IAAI,6DAAe,CAAnB,C;YAAsB,yBAA
O,EAAP,C;UACtB,yBAAO,OAAP,CAAgB,gBAAO,GAAP,C;;QAEpB,IAAI,UAAJ,C;UACI,IAAI,6DAAe,CAAn
B,C;YAAsB,yBAAO,EAAP,C;UAElB,gBAAW,CAAX,IAAgB,OAAhB,IAA2B,QAA3B,IAAuC,UAAvC,C;YACI
,mCAAiB,OAAjB,EAA0B,WAA1B,EAAuC,CAAvC,EAA0C,GAA1C,EAA2D,KAA3D,C;eACJ,mBAAe,OAAf,C
;YACI,mCAAiB,cAAc,OAAd,IAAjB,EAA0C,cAAc,OAAd,IAA1C,EAAmE,CAAnE,EAAsE,IAAtE,EAAwF,KA
AxF,C;eACJ,mBAAe,IAAf,C;YACI,mCAAiB,cAAc,IAAd,IAAjB,EAAsC,cAAc,IAAd,IAAtC,EAA2D,CAA3D,E
AA8D,IAA9D,EAAgF,KAAhF,C;;YAEA,yBAAO,WAAP,CAAoB,gBAAO,IAAP,C;;QAGhC,IAAI,cAAc,aAAa,C
AA/B,C;UAAkC,yBAAO,CAAP,EAAU,EAAV,CAAe,gBAAO,EAAP,C;QAvC/B,OQ52B3B,SoHoUqC,W;;K;4C
5HqlB5C,yE;MACI,yBAAO,KAAP,C;MACA,IAAI,eAAc,CAAlB,C;QACI,yBAAO,EAAP,C;QACA,iBAAuC,W
AAtB,UAAW,WAAW,EAAS,cAAT,EAAyB,EAAzB,C;QACR,sB;;UuB/0BzB,Q;UAAA,OAAQ,WAAR,evB+0Bc
,UuB/0Bd,CAAQ,CAAR,W;UAAd,OAAc,cAAd,C;YAAc,uB;YACV,IvB80BiD,UuB90BnC,YvB80BU,UuB90BV
,YAAK,KAAL,EvB80BmC,MAAM,EuB90BvD,C;cACI,qBAAO,K;cAAP,uB;;;UAGR,qBAAO,E;;;QvB00BC,oB
AAoB,qBAAuC,CAAvC,I;QAEhB,KAAC,SAAD,IAAc,gBAAgB,CAA9B,C;UAAmC,8BAAY,UAAZ,EAAwB,C
AAxB,EAA2B,aAA3B,C;;UAC3B,8BAAY,UAAZ,EAAwB,CAAxB,EAA2B,CAAC,CAAC,gBAAgB,CAAhB,IA
AD,IAAsB,CAAtB,IAAD,IAA4B,CAA5B,IAA3B,C;;MAGhB,yBAAO,IAAP,C;K;0CAGJ,0B;MAgBwC,wB;QAA
A,WAAgB,C;MKv+BxD,IAAI,ELw+BQ,YAAY,CKx+BpB,CAAJ,C;QACI,cLu+ByB,oD;QKt+BzB,MAAM,gCA
AyB,OAAQ,WAAjC,C;;MLu+BN,aAAa,sBAAS,IAAT,C;MACb,IAAW,WAAP,MAAO,CAAX,C;QAAyB,OAAO
,MAAO,W;MACvC,OAAO,sBAAsB,MAAtB,EAAuC,eAAT,QAAS,EAAa,EAAb,CAAvC,IAAgE,UAAL,IAAK,C
;K;qCAI3E,Y;M4H3nBuB,gBAAhB,sB;M5HyoBH,IAAI,iBAAJ,C;QAAkB,yBAAO,EAAP,C;MAClB,yBAAO,IA
AP,C;MAC4B,YAAd,kB;MAjXP,YAAO,kB;MAAP,cAAqB,sB;MAArB,cAAuC,sB;MAAvC,kBAAyD,0B;MAkX
5D,cACY,K;MACZ,IAAI,iBAAJ,C;QAEI,wB;;MAEJ,eAAe,oB;MACf,iBAAiB,YAAW,CAAX,IAAgB,gBAAe,C;
MAChD,iBAAiB,YAAW,CAAX,KAAiB,cAAc,QAA/B,C;MACjB,IAAI,QAAJ,C;QACI,yBAAO,OAAP,CAAc,g
BAAO,EAAP,C;;MAElB,IAAI,UAAJ,C;QACI,yBAAO,OAAP,CAAgB,gBAAO,EAAP,C;;MAEpB,IAAI,eAAe,C
AAC,QAAD,IAAa,CAAC,UAA7B,CAAJ,C;QACI,mCAAiB,OAAjB,EAA0B,WAA1B,EAAuC,CAAvC,EAA0C,
GAA1C,EAA2D,IAA3D,C;;MApBuB,OQ58B5B,SoHoUqC,W;K;;;;;kC5H5YhD,Y;MAAA,c;MAuBiD,2D;MAvBj
D,a;K;gCAAA,iB;MAAA,2IAuBiD,gDAvBjD,G;K;IAkjCA,qC;MAIW,Q;MAAA,IAAI,6DAAJ,C;QACH,uBAAg

B,4BAAiC,oBAAL,SAAK,CAAjC,EAA2C,IAA3C,yCAAhB,C;;QAES,oBAAT,8BAAS,EAAW,IAAX,C;MAHb,W;K;IAMJ,uC;MAII,kBAAkB,4BAA4B,SAA5B,0CAAiE,IAAjE,C;MAClB,IAAa,WAAD,aAAR,yDAAsB,WAAt B,CAAJ,C;QACI,OAAO,gBAAgB,4BAA4B,SAA5B,EAAkC,IAAlC,yCAAhB,C;;QAEP,aAAa,sBAAoB,SAApB, EAA0B,IAA1B,0C;QACb,OAAO,iBAAwB,WAAP,MAAO,yBAAsB,UAAtB,CAAxB,C;;K;IAIf,uC;MAaW,Q;M AHP,gBAAgB,oBAAoB,SAApB,EAA0B,IAA1B,yC;MK3jChB,IAAI,CL4jCI,CAAW,QAAV,SAAU,CK5jCnB,C; QACI,cL2jC0B,+B;QK1jC1B,MAAM,gCAAyB,OAAQ,WAAjC,C;;ML2jCV,YAAsB,YAAV,SAAU,C;MACf,IA AI,sEAAqB,SAArB,CAAJ,C;QACH,uBAAgB,KAAhB,C;;QAEA,aAAwE,YAA3D,oBAAoB,SAApB,EAA0B,IA A1B,0CAA2D,C;QACxE,kCAA2B,MAA3B,C;;MAJJ,W;K;IAgBuB,oC;MAAQ,oE;K;IAOP,sC;MAAQ,sE;K;IAW N,sC;MAAQ,sE;K;IAQV,qC;MAAQ,qE;K;IAOP,uC;MAAQ,uE;K;IAWN,uC;MAAQ,uE;K;IAQX,qC;MAAQ,qE; K;IAOP,uC;MAAQ,uE;K;IAWN,uC;MAAQ,uE;K;IAQhB,gC;MAAQ,gE;K;IAOP,kC;MAAQ,kE;K;IAWN,kC;M AAQ,kE;K;IAQX,gC;MAAQ,gE;K;IAOP,kC;MAAQ,kE;K;IAWN,kC;MAAQ,kE;K;IAQb,8B;MAAQ,8D;K;IAOP ,gC;MAAQ,gE;K;IAWN,gC;MAAQ,gE;K;IAQZ,6B;MAAQ,6D;K;IAOP,+B;MAAQ,+D;K;IAWN,+B;MAAQ,+D; K;yEAG/B,+B;MAIqE,8BAAW,SAAX,C;K;2EAErE,+B;MAUwE,8BAAW,SAAX,C;K;IAIxE,yC;MACI,aAAa,K AAM,O;MACnB,IAAI,WAAU,CAAd,C;QAAiB,MAAM,gCAAyB,qBAAzB,C;MACvB,YAAY,C;MACZ,aAAa,g CAAS,K;MACtB,qBAAqB,U;MACrB,QAAM,iBAAM,KAAN,CAAN,C;aACI,E;aAAA,E;UAAY,qB;UAAZ,K;;M AEJ,cAAc,QAAQ,C;MACtB,iBAAiB,WAAiB,aAAN,KAAM,EAAW,EAAX,C;MAE9B,cAAU,KAAV,C;QACI, MAAM,gCAAyB,eAAzB,C;WACV,qBAAM,KAAN,MAAgB,EAAhB,C;QACI,IAAI,mCAAW,MAAf,C;UAAuB, MAAM,+B;QAC7B,sBAAsB,K;QACtB,sBAAsB,K;QACtB,eAA8B,I;QAC9B,OAAO,QAAQ,MAAf,C;UACI,IAA I,iBAAM,KAAN,MAAgB,EAApB,C;YACI,IAAI,mBAAmB,mCAAW,MAAlC,C;cAA0C,MAAM,+B;YAChD,kB AAkB,I;YAClB,Q;;UAEkB,iBAAe,K;UA+EjD,QAHgC,U;UAIhC,Y;YAAO,eAhFqB,KAgFjB,O;YAAJ,S;cAAc,S AAU,YAhFH,KAgFG,YAAK,CAAL,E;cAAV,OAhFqC,CAAM,kBAAK,EAAL,CAAN,qCAAkB,2C;;;;YAgFnC,a ;;UAhF7B,gBAAgB,KkBllCgE,WlB8pClF,UkB9pCkF,ElBmqCrF,CkBnqCqF,C;UlBmlChF,IAAI,SwBziCgC,YA AU,CxByiC9C,C;YAAyB,MAAM,+B;UAC/B,gBAAS,SAAU,OAAnB,I;UACqB,cAAU,K;UuB5sCpC,U;UAAA,I AAI,WAAS,CAAT,IAAc,WAAS,iBvB4sCP,KuB5sCO,CAA3B,C;YAAA,SvB4sCoB,KuB5sCkB,YAAI,OAAJ,C;; YvB4sCO,MAAM,gCAAyB,qCAAzB,C;;UAA9C,qB;UACA,qB;UACA,WAAW,sBAAsB,QAAtB,EAAgC,eAAh C,C;UACX,IAAI,YAAY,IAAZ,IAAoB,yBAAY,IAAZ,MAAxB,C;YAA0C,MAAM,gCAAyB,yCAAzB,C;UAChD, WAAW,I;UACX,eAAyB,WAAV,SAAU,EAAQ,EAAR,C;UACzB,IAAI,+CAAgC,WAAW,CAA/C,C;YACI,YAA Y,SkB5lCgE,WlB4lC5C,CkB5lC4C,ElB4lCzC,QkB5lCyC,C;YlB6lC5E,4BAA2C,aAAjC,0BAA0B,KAA1B,CAAi C,EAAW,IAAX,CAA3C,C;YACA,4BAAmD,aAAX,SAA9B,SkBjmCmD,WlBimC/B,QkBjmC+B,ClBimCrB,CA AW,EAAW,IAAX,CAAnD,C;;YAEA,4BAA+C,aAArC,0BAA0B,SAA1B,CAAqC,EAAW,IAAX,CAA/C,C;;;aAI Z,c;QACI,MAAM,+B;;QACV,IAAM,cAAN,KAAM,EAAc,KAAd,EAAqB,cAArB,EAAqC,CAArC,ESnzCH,MA AO,KTmzCmD,SAAS,KAAT,ISnzCnD,ETmzCmE,cAAe,OSnzClF,CTmzCJ,EAA4G,IAA5G,CAAN,C;UACI,SA AS,gCAAS,S;;UAIlB,iBAA8B,I;UAC9B,iBAAiB,K;UACjB,kBAAkB,CAAC,O;UACnB,IAAI,WAAW,iBAAM,K AAN,MAAgB,EAA3B,IAAwC,QAAN,KAAM,CAAN,KAAgB,EAAtD,C;YACI,cAAc,I;YACd,IAAI,oCAAW,uB AAX,EAAW,MAAX,CAAJ,C;cAAyB,MAAM,gCAAyB,eAAzB,C;;UAEnC,OAAO,QAAQ,MAAf,C;YACI,IAAI, cAAc,WAAlB,C;cA8CZ,UA7CwC,K;cA8CxC,Y;gBAAO,mBA9CiB,KA8Cb,O;gBAAJ,W;kBAAc,SA9C4B,UA8 ClB,YA9CP,KA8CO,YAAK,GAAL,EA9CkB,MAAM,E;;;;gBA8Cd,iB;;cA9CzB,QA+CT,G;;YA7CK,aAAa,I;YA CS,mBAAe,K;YA0CjD,UAHgC,Y;YAIhC,Y;cAAO,mBA3CqB,KA2CjB,O;cAAJ,W;gBAAc,WAAU,YA3CH,KA 2CG,YAAK,GAAL,E;gBAAV,SA3CqC,CAAM,kBAAK,EAAL,CAAN,uCAAkB,oBAAM,E;;;;cA2CzC,iB;;YA3C 7B,kBAAgB,KkBvnCgE,WlB8pClF,YkB9pCkF,ElBmqCrF,GkBnqCqF,C;YlBwnChF,IAAI,WwB9kCgC,YAAU,C xB8kC9C,C;cAAyB,MAAM,+B;YAC/B,gBAAS,WAAU,OAAnB,I;YACqB,mBAAe,K;YAuChD,UAHgC,Y;YAI hC,Y;cAAO,mBAxCoB,KAwChB,O;cAAJ,W;gBAAc,WAAU,YAxCJ,KAwCI,YAAK,GAAL,E;gBAAV,SAxCoC ,CAAM,kBAAK,GAAL,CAAN,mC;;;;cAwChB,iB;;YAxC7B,eAAe,KkB1nCiE,WlB8pClF,YkB9pCkF,ElBmqCrF, GkBnqCqF,C;YlB2nChF,gBAAS,QAAS,OAAlB,I;YACA,aAAW,wBAAwB,QAAxB,C;YACX,IAAI,cAAY,IAAZ ,IAAoB,2BAAY,MAAZ,MAAxB,C;cAA0C,MAAM,gCAAyB,yCAAzB,C;YAChD,aAAW,M;YACX,iBAAyB,W AAV,WAAU,EAAQ,EAAR,C;YACzB,IAAI,aAAW,CAAf,C;cACI,cAAY,WkBjoCgE,WlBioC5C,CkBjoC4C,ElBi oCzC,UkBjoCyC,C;clBkoC5E,4BAAyB,aAAT,OAAN,OAAM,CAAS,EAAW,MAAX,CAAzB,C;cACA,4BAAmD ,aAAX,SAA9B,WkBtoCmD,WlBsoC/B,UkBtoC+B,ClBsoCrB,CAAW,EAAW,MAAX,CAAnD,C;cACA,IAAI,QA

AQ,MAAZ,C;gBAAoB,MAAM,gCAAyB,mCAAzB,C;;cAE1B,4BAA6B,aAAT,OAAV,WAAU,CAAS,EAAW,M
AAX,CAA7B,C;;;;;MAKhB,OAAW,UAAJ,GAAiB,MAAD,aAAhB,GAA6B,M;K;IAIxC,0C;MACI,aAAa,KAAM,
O;MACnB,iBAAiB,C;MACjB,IAAI,SAAS,CAAT,IAAc,YAAY,IAAZ,mBAAM,CAAN,EAAlB,C;QAAoC,+B;;M
AChC,YAAC,SAAS,UAAT,IAAD,IAAwB,E;MAAxB,S;QAA4D,gBAA7B,yBAAkB,iBAAN,KAAM,CAAlB,C;Q
AA6B,c;;UW8ShD,U;UADhB,IAAI,wCAAsB,mBAA1B,C;YAAqC,aAAO,I;YAAP,e;;UACrB,6B;UAAhB,OAAg
B,gBAAhB,C;YAAgB,2B;YAAM,IAAI,CX9S4C,CAAa,kBAAK,EAAL,CAAb,oCW8SjC,OX9SiC,EW8ShD,C;c
AAyB,aAAO,K;cAAP,e;;;UAC/C,aAAO,I;;;QX/SyD,iB;;MAAhE,S;QAEI,OAAW,iBAAM,CAAN,MAAY,EAAh
B,sD;;MAGX,OAAiB,WAAN,KAAM,EAAW,GAAX,CAAV,GAAyC,OAAR,QAAN,KAAM,EAAK,CAAL,CAA
Q,CAAzC,GAA6D,OAAN,KAAM,C;K;IAKxE,0D;MAII,QAHgC,U;MAIhC,OAAO,IAAI,gBAAJ,IAJqC,SAIvB,
CAAU,iCAAK,CAAL,EAAV,CAArB,C;QAAyC,a;;MAJzC,OkB9pC4F,oBlB8pClF,UkB9pCkF,ElBmqCrF,CkBnq
CqF,C;K;IlBgqChG,qD;MACI,QAAQ,U;MACR,OAAO,IAAI,gBAAJ,IAAc,UAAU,iCAAK,CAAL,EAAV,CAAr
B,C;QAAyC,a;;MACzC,OAAO,C;K;;;;;IAmBX,8B;MAA+C,qCAAQ,OAAR,E;K;IAC/C,+B;MAAgD,2CAAS,OA
AT,E;K;IAEhD,sC;MAAiD,oBAAS,sBAAgB,CAAhB,CAAT,C;K;IACjD,wC;MAAmD,oBAAU,uBAAiB,CAAjB,
CAAD,yBAAuB,CAAvB,EAAT,C;K;IACnD,oD;MAAoE,oBAAU,sBAAgB,CAAhB,CAAD,yBAAsB,iBAAtB,E
AAT,C;K;IACpE,0C;MACI,IAAI,sEAAqB,SAArB,CAAJ,C;QAAA,OACI,gBAAgB,KAAhB,C;;QADJ,OAGI,iBA
AiB,cAAc,KAAd,CAAjB,C;;K;IAGR,4C;MACI,IAAI,kEAAgC,mBAAhC,CAAJ,C;QAAA,OACI,gBAAgB,cAAc,
MAAd,CAAhB,C;;QADJ,OAGI,iBAAwB,WAAP,MAAO,yBAAsB,UAAtB,CAAxB,C;;K;I6Mt4CR,8B;MAEgD,
QAAM,SAAN,M;aAC5C,a;UAD4C,OAChB,I;aAC5B,c;UAF4C,OAEf,I;aAC7B,c;UAH4C,OAGf,I;aAC7B,S;UAJ
4C,OAIpB,G;aACxB,S;UAL4C,OAKpB,G;aACxB,O;UAN4C,OAMtB,G;aACtB,M;UAP4C,OAOvB,G;;UxMuEw
B,MAAM,6BAA8B,CwMtEnE,mBAAgB,SxMsEmD,YAA9B,C;;K;IwMnEvD,4C;MACwE,QAAM,SAAN,C;aAC
pE,I;UADoE,6C;aAEpE,I;UAFoE,8C;aAGpE,I;UAHoE,8C;aAIpE,G;UAJoE,yC;aAKpE,G;UALoE,yC;aAMpE,G;
UANoE,uC;aAOpE,G;UAPoE,sC;;UAQ5D,MAAM,gCAAyB,uCAAoC,SAA7D,C;;K;IAGlB,yD;MAGQ,KAAC,e
AAD,C;QAEQ,IADE,OACF,Q;UAHZ,sC;;UAIoB,MAAM,gCAAyB,4EAAqD,OAArD,CAAzB,C;;QAIlB,QAAM,
OAAN,C;eACI,E;YATZ,uC;eAUY,E;YAVZ,yC;eAWY,E;YAXZ,yC;;YAYoB,MAAM,gCAAyB,yDAAkC,OAAl
C,CAAzB,C;;;K;IC5F9B,4B;K;;;MC8FI,kC;;;;;IAlEA,gC;MAAA,oC;K;6CAUI,Y;MAAwC,OAAA,iCAAoB,U;K;
8CAC5D,Y;MAAkC,OAAA,iCAAoB,W;K;IAiBrB,qD;MAAqB,8B;K;8DAClD,Y;MAAsC,OAAA,iCAAoB,qBA
AY,IAAZ,C;K;+DAC1D,oB;MAAuD,OAAA,iCAAoB,uBAAc,IAAd,EAAoB,QAApB,C;K;gEAC3E,oB;MAAwD,
OAAA,iCAAoB,uBAAc,IAAd,EAAqB,QAAD,aAApB,C;K;gEAC5E,Y;MAAuC,QAAC,iBAAa,a;K;mEACrD,Y;
MAA0C,OAAA,iBAAa,a;K;iEAEvD,iB;MACI,IAAI,yDAAJ,C;QACI,MAAM,gCAAyB,sFAAmF,IAAnF,aAA6F,
KAAtH,C;MACV,OAAO,IAAK,eAAM,KAAN,C;K;iEAGhB,iB;MAW4D,OAAA,iCAAoB,2BAAkB,IAAlB,EAA
wB,KAAxB,C;K;qEAEhF,iB;MAQI,OAAC,mBAAO,KAAP,CAAc,iBAAU,gCAAS,KAAnB,C;K;;;;;4DAjDvB,Y;
MAAA,OAesD,gEAftD,M;K;4DAAA,Y;MAAA,c;MAesD,gE;MAftD,a;K;0DAAA,iB;MAAA,2IAesD,0DAftD,G;
K;;;IAbJ,4C;MAAA,2C;QAAA,0B;;MAAA,oC;K;IAkEA,gC;MAAA,oC;K;;;IAAA,4C;MAAA,2C;QAAA,0B;;M
AAA,oC;K;;;;qCA2BA,oB;MAWyD,4BAAiB,IAAjB,EAAuB,QAAvB,C;K;sCAEzD,oB;MAW+D,wBAAM,QAA
D,aAAL,C;K;sCAG/D,Y;MAMqC,QAAC,iBAAa,a;K;yCAEnD,Y;MAMwC,OAAA,iBAAa,a;K;;;;gDAUrD,oB;M
AAkF,wBAAM,QAAD,aAAL,C;K;mDAmBlF,iB;MAYI,OAAC,mBAAO,KAAP,CAAc,iBAAU,gCAAS,KAAnB,
C;K;;IAYO,4C;MAAC,gB;MAAoB,4B;K;4CAC/C,Y;MAAsC,OAAA,SAAK,aAAL,cAAoB,eAApB,C;K;6CAEtC
,oB;MAAkD,4BAAiB,SAAjB,EAAuB,4BAAa,QAAb,CAAvB,C;K;;ICxMV,sC;MAAC,gB;K;IAOf,4E;MAAC,4B;
MAA6B,8B;MAAgD,sB;K;+DACpG,Y;MAAsC,OAAI,aAAO,aAAX,GAA0B,aAAD,aAAzB,GAAsE,aAA/B,iBA
AW,OAAX,UAAoB,gBAApB,CAA+B,EAAW,iBAAW,KAAtB,CAAhC,cAA8D,aAA9D,C;K;gEAC5E,oB;MAA4
D,+CAAa,gBAAb,EAAwB,iBAAxB,EAAoC,0BAAS,QAAT,CAApC,C;K;kEAC5D,iB;MAQa,Q;MAPT,IAAI,8D
AA0B,QAAA,IAAK,aAAL,EAAmB,KAAM,aAAzB,CAA9B,C;QACI,MAAM,gCAAyB,sFAAmF,IAAnF,aAA6F,
KAAtH,C;MAMV,IAAI,EAAK,OAAL,IAAK,SAAL,wBAAe,KAAM,SAArB,aAA+B,IAAK,SAAO,aAA/C,C;QA
A6D,OAAO,gCAAS,K;MAC7E,iBAAiB,IAAK,SAAL,cAAc,KAAM,SAApB,C;MACjB,oBAAuD,aAAlC,IAAK,
YAAL,UAAiB,KAAM,YAAvB,CAAkC,EAAW,iBAAW,KAAtB,C;MAEvD,OAAO,CAAI,6CAAkB,UAAD,aAA
jB,QAAJ,IAAkC,gCAAS,KAA3C,GAAqD,0BAAgB,UAAhB,C;K;2DAGhE,iB;MACqE,Q;MAAjE,oEAAyB,OA
AA,IAAK,aAAL,EAAmB,KAAM,aAAzB,CAAzB,KAAgE,CAAC,0BAAO,KAAP,CAAD,wBAAkB,gCAAS,KA
A3B,QAAhE,C;K;8EAEJ,Y;MACI,IAAI,aAAO,aAAX,C;QAAyB,OAAO,a;MAChC,WAAAW,iBAAW,K;MACtB,I

AAI,kEAAJ,C;QACI,OAAiB,aAAV,gBAAU,EAAW,IAAX,CAAV,aAA6B,aAA7B,C;;MAEX,YAAY,mEAAmD,
IAAnD,C;MACZ,sBAAsB,qBAAY,KAAZ,C;MACtB,mBAAmB,wBAAY,KAAZ,C;MAEL,YAAP,a;MhN+oBJ,o
BAAO,oB;MAAP,wBAAuB,0B;MgN9oBtB,mBAAmB,oBAAoB,OAApB,I;MACnB,qBAAqB,oBAAoB,OAApB,
I;MAFzB,OAKkB,gCAAb,ahNAiC,WgNAD,chNAC,2BgNAjC,CAAD,ahN0CoC,agNzC3B,wCAAkB,YAAlB,Eh
NyC2B,4BgN1CpC,chN6D+B,agN3DvB,ahN2DuB,uBgN7D/B,C;K;6DAOR,Y;MAA+B,OAAA,gCAAoB,W;K;6
DAEnD,Y;MAAkC,yBAAe,gBAAf,cAA0C,UAAhB,iBAAW,KAAK,CAA1C,WAA0D,aAA1D,WAAqE,gCAArE,
WAA6F,iBAA7F,M;K;;+CAGtC,Y;MAA6C,+CAAa,WAAb,EAAqB,IAArB,EAA2B,gCAAS,KAApC,C;K;;IAW
H,wC;MAAC,gB;K;IAQf,gF;MAAC,4B;MAA+B,8B;MAAkD,sB;K;mEAC1G,Y;MAAsC,OAAgC,aAA/B,iBAA
W,OAAX,GAAoB,gBAAW,EAAW,iBAAW,KAAtB,CAAhC,cAA8D,aAA9D,C;K;oEACtC,oB;MAA4D,mDAAe,
gBAAf,EAA0B,iBAA1B,EAAsC,0BAAS,QAAT,CAAtC,C;K;sEAE5D,iB;MAIa,Q;MAHT,IAAI,kEAA4B,QAAA
,IAAK,aAAL,EAAmB,KAAM,aAAzB,CAAhC,C;QACI,MAAM,gCAAyB,sFAAmF,IAAnF,aAA6F,KAAtH,C;M
AEV,IAAI,EAAK,OAAL,IAAK,SAAL,wBAAe,KAAM,SAArB,aAA+B,IAAK,SAAO,aAA/C,C,C;QAA6D,OAAO,
gCAAS,K;MAC7E,iBAAiB,IAAK,SAAL,cAAc,KAAM,SAApB,C;MACjB,oBAAuD,aAAlC,IAAK,YAAL,GAAi
B,KAAM,YAAW,EAAW,iBAAW,KAAtB,C;MACvD,OAAO,CAAI,6CAAkB,UAAD,aAAjB,QAAJ,IAAkC,gCA
AS,KAA3C,GAAqD,0BAAgB,UAAhB,C;K;+DAGhE,iB;MAC8E,Q;MAA1E,OAAO,iEAA2B,OAAA,IAAK,aAA
L,EAAmB,KAAM,aAAzB,CAA3B,KAAkE,CAAC,0BAAO,KAAP,CAAD,wBAAkB,gCAAS,KAA3B,QAAlE,C;
K;iEAGX,Y;MACI,OAAkB,aAAV,gBAAU,EAAW,iBAAW,KAAtB,CAAV,aAAwC,aAAxC,CAAgD,W;K;iEAG
5D,Y;MAAkC,2BAAiB,gBAAjB,GAA4C,UAAhB,iBAAW,KAAK,CAA5C,WAA4D,aAA5D,UAAqE,iBAArE,M;
K;;iDAGtC,Y;MAA6C,mDAAe,WAAf,EAAuB,IAAvB,EAA6B,gCAAS,KAAtC,C;K;;IAGjD,0B;MAgB8B,yE;M
AC1B,mB;K;oCAEA,Y;MAA4B,qB;K;iDAE5B,oB;MAWc,Q;MADV,gBAAgB,QAAS,gBAAO,SAAP,C;MACf,I
AAI,gDAA+B,4CAAnC,C;QAEN,iBAAiB,mBAAU,SAAV,C;QACjB,IAAI,mBAAY,SAAZ,gBAAyB,CAAzB,IA
A8B,mBAAY,UAAZ,eAAyB,CAA3D,C;UAA8D,gBAAS,QAAT,C;QAC9D,iB;;QAEA,YAAY,QAAS,kBAAS,S
AAT,C;QAErB,mBAAiB,4BAAU,K;QAC3B,IAAI,sDAA+B,kDAAnC,C;UAAgE,gBAAS,QAAT,C;QACrD,8BA
AX,YAAW,C;;MAVf,qB;K;0CAcJ,oB;MACI,MAAM,6BAAsB,iDAA+C,cAA/C,cAA8D,UAAL,SAAK,CAA9D,
wBAA2F,QAA3F,MAAtB,C;K;;ICpKd,yC;MACI,iBAAiB,QAAS,mB;MAC1B,IA2DA,OA3DI,MA2DH,8BAAO,
CAAP,EAAD,kCA3DA,C;QACI,OAAO,wBAAwB,MAAxB,EAAgC,QAAhC,EAA0C,UAA1C,C;;MAEX,IAwDA
,OAxDI,UAwDH,8BAAO,CAAP,EAAD,kCAxDA,C;QACI,OAAO,sBAAsB,MAAtB,EAA8B,QAA9B,C;;MAGX,
aAAa,WAAS,UAAT,C;MACb,IAAM,WAAW,MAAX,CAAD,KAAyB,eAAe,MAAf,CAAzB,CAAD,cAAoD,CAA
xD,C;QACI,OAAW,oBAAS,CAAb,sD;;MAEX,OAAO,M;K;IAGX,+D;MACI,IAAI,QAAS,aAAT,IAA0B,WAAW
,UAAX,eAAwB,CAAtD,C;QAA0D,MAAM,gCAAyB,uCAAzB,C;MAChE,OAAO,M;K;IAGX,iD;MACI,WAAW,
qBAAW,CAAX,C;MACX,IAsCA,OAtCI,IAAK,mBAsCR,8BAAO,CAAP,EAAD,kCAtCA,C;QAEI,OAA8D,uBA
AtD,oBAAS,QAAS,yDAAoC,C;;QAE9D,OAAO,cAAc,cAAc,MAAd,EAAsB,IAAtB,CAAd,EAA2C,sBAAW,IAA
X,CAA3C,C;;K;IAIf,2C;MACI,IA6BA,OA7BI,QA6BH,8BAAO,CAAP,EAAD,kCA7BA,C;QACI,OAAkB,aAAT,
QAAS,kCAAX,a;;MAEX,OAAO,qBAAqB,OAArB,EAA8B,QAA9B,C;K;IAGX,qD;MACI,IAsBA,OAtBI,SAsBH
,8BAAO,CAAP,EAAD,kCAtBA,C;QACI,IAAI,kBAAa,SAAb,CAAJ,C;UAA4B,OAAO,gCAAS,K;QAC5C,OAA
mB,aAAV,SAAU,kCAAZ,a;;MAEX,IAkBA,OAlBI,SAkBH,8BAAO,CAAP,EAAD,kClBA,C;QACI,OAAiB,aA
AV,SAAU,kC;;MAErB,OAAO,qBAAqB,SAArB,EAAgC,SAAhC,C;K;IAGX,kD;MACI,aAAa,kBAAW,QAAX,C;
MACb,IAAK,WAAW,QAAX,CAAD,KAA2B,WAAW,QAAX,CAAqB,MAAhD,eAAwD,CAA5D,C;QACI,eAAe,
iCAAW,OAAX,YAA6B,iCAAW,OAAX,EAA7B,C;QACf,eAAe,oCAAW,OAAX,YAA6B,oCAAW,OAAX,EAA7
B,C;QACf,OjN8C4C,aiN9CrC,QjN8CqC,4BiN9CrC,ajNQoC,aiNRZ,QjNQY,2BiNRpC,C;;MAEX,OjNM+C,aiNN
xC,MjNMwC,2B;K;IiNHnD,gC;MAEI,cAAC,uCAAO,CAAP,EAAD,kC;K;qFChEJ,yB;MAAA,yC;MAAA,wB;Q
A4CI,WAAW,8B;QAjC6B,KAkCxC,E;QAlCA,OAmCO,IAAK,a;O;KA9ChB,C;uFAeA,4B;MAYI,WAAW,mB;M
ACX,O;MACA,OAAO,IAAK,a;K;uFAGhB,4B;MAYI,WAAW,mB;MACX,O;MACA,OAAO,IAAK,a;K;IAYe,qC
;MAAC,kB;MAAc,wB;K;;sCAR9C,Y;MAQgC,iB;K;sCARhC,Y;MAQ8C,oB;K;wCAR9C,2B;MAAA,sBAQgC,q
CARhC,EAQ8C,8CAR9C,C;K;oCAAA,Y;MAAA,OAQgC,iDARhC,IAQ8C,8CAR9C,O;K;oCAAA,Y;MAAA,c;
MAQgC,sD;MAAc,yD;MAR9C,a;K;kCAAA,iB;MAAA,4IAQgC,sCARhC,IAQ8C,4CAR9C,I;K;iGAUA,yB;MAA
A,yC;MAkCA,8C;MAlCA,wB;QA+CI,WAAW,8B;QACX,aAnC8C,KAmCjC,E;QAnCb,OAoCO,oBAAW,MAAX
,EAAmB,IAAK,aAAxB,C;O;KAjDX,C;mGAgBA,yB;MAAA,8C;MAAA,mC;QAaI,WAAW,mB;QACX,aAAa,O;

QACb,OAAO,oBAAW,MAAX,EAAmB,IAAK,aAAxB,C;O;KAfX,C;mGAkBA,yB;MAAA,8C;MAAA,mC;QAaI,
WAAW,mB;QACX,aAAa,O;QACb,OAAO,oBAAW,MAAX,EAAmB,IAAK,aAAxB,C;O;KAfX,C;IpK/CA,2E;M
ASI,sC;MAAA,4C;K;IATJ,mGAWY,Y;MAAQ,2B;KAXpB,E;IAAA,4DAaQ,kB;MACI,wBAAW,MAAX,C;K;IA
dZ,wF;IqKewC,sC;MACpC,0B;K;;IAGJ,kC;MAUI,OAA2C,CAA3C,2BAA6B,uBAA7B,EAAoC,KAApC,CAA2C
,e;K;IAE/C,8B;K;kDAuBI,4B;MASI,MAAM,qCAA8B,8CAA9B,C;K;;;IAW4B,8C;MAGtC,6B;MAEmD,UAMX,
M;MAPxC,kBACmD,mE;MAEnD,eAC0B,K;MAE1B,cACwC,kE;MAExC,gBACmC,gB;K;iGAG/B,Y;MAAQ,0C
;K;0DAEZ,kB;MACI,cAAY,I;MACZ,gBAAc,M;K;IAGsE,iG;MAAA,uB;QAExE,Q;QAAZ,qCAAY,8D;QACZ,sC
AAa,a;QAFb,OAGA,yB;O;K;2DAJJ,+B;MAAkD,OAAsC,wDAAtC,c;K;IAOyE,uH;MAAA,uB;QAExG,Q;QAAf,i
BAAe,8F;QACf,eAAK,2B;QAA6B,mC;Q3M/FtB,gBAAT,Q;Q2MoG0D,kB;QAJzD,sBAAsB,SAAK,W;QAC3B,I
AAI,eAAa,eAAjB,C;UAEI,iC;UACA,mBAAY,oCAAwB,eAAxB,EAAyC,kEAAzC,C;;UAGZ,mBAAY,kE;;QAEh
B,oBAAa,e;QAZjB,OAcA,yB;O;K;6DAfJ,0C;MAAqF,OAAsC,qEAAtC,c;K;IAqBzB,mI;MAAAA,qB;QACxD,yCA
AgB,uB;QAGhB,qCAAY,Y;QACZ,uCAAc,E;QAClB,W;O;K;iEATA,iC;MAGwB,wCAAa,mCAAb,EAAoC,kFA
ApC,C;K;mDAQxB,Y;MAMuB,UADC,MACD,EAIH,MAJG,EAaK,M;MAjBxB,OAAO,IAAP,C;QAEI,aAAa,IA
AK,S;QACF,SAAL,IAAK,O;QAAL,mB;UACyB,gBAArB,0D;UtKtBhB,U;UADP,yB;UsKuBe,OtKtBR,sF;;QsKq
BC,WAAW,M;QAGX,IAAI,mDAAoB,MAApB,QAAJ,C;;YAIiB,SAAT,epKtJV,CoKsJuD,IpKtJvD,EoKsJ6D,Yp
KtJ7D,EoKsJoE,IpKtJpE,EAA8C,KAA9C,C;;YoKuJQ,gC;cACE,IrKvJhB,oBDgDQ,WAAO,csKuG0B,CtKvG1B,
CAAP,CChDR,C;cqKwJgB,Q;;cALI,O;;UAAR,c;UAQA,IAAI,MAAM,yBAAV,C;YACI,IrKrKhB,oBDgDQ,WsK
qHoB,0EtKrHpB,CChDR,C;;;UqKwKY,gBAAc,gB;UACd,IAAK,oBAAW,MAAX,C;;;K;;0ECxMrB,4B;MAyLI,Q
ApLK,SAoLG,GApLoB,KAoLpB,I;MACR,IAAI,CArLC,SAqLD,GArLwB,KAqLxB,IAAiB,CAAjB,IAAsB,eArL
E,KAqLF,MArLrB,SAqLL,C;QAA6C,a;;MArL7C,OAsLO,C;K;kEApLX,yB;MAAA,0B;MAAA,mC;QAgMI,QAv
LK,SAuLG,GAvLe,KAuLf,I;QAvLR,OAAgC,OAwLzB,KAxLgB,KAwLX,GAAW,CAAC,CAAC,IAxLF,KAwLC
,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAlD,KAxLyB,C;O;KATpC,C;4EAWA,4B;MA
uKI,QAlKK,SAkKG,GAlKoB,KAkKpB,I;MACR,IAAI,CAnKC,SAmKD,GAnKwB,KAmKxB,IAAiB,CAAjB,IA
AsB,eAnKE,KAmKF,MAnKrB,SAmKL,C;QAA6C,a;;MAnK7C,OAoKO,C;K;kEAlKX,yB;MAAA,4B;MAAA,m
C;QA8KI,QArKK,SAqKG,GArKe,KAqKf,I;QArKR,OAAgC,QAsKzB,KAtKgB,KAsKX,GAAW,CAAC,CAAC,I
AtKF,KAsKC,KAAmB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAlD,KAtKyB,C;O;KATpC,C;4E
AWA,4B;MAqJI,QAhJK,SAgJG,GAhJc,KAgJd,I;MACR,IAAI,CAjJC,SAiJD,GAjJkB,KAiJlB,IAAiB,CAAjB,IAA
sB,eAjJJ,KAiJI,MAjJrB,SAiJL,C;QAA6C,a;;MAjJ7C,OAkJO,C;K;kEAhJX,4B;MA4JI,QAnJK,SAmJG,GAnJS,K
AmJT,I;MAnJR,OAoJO,KApJU,KAoJL,GAAW,CAAC,CAAC,IApJR,KAoJO,KAAmB,KAAK,CAAC,CAAD,IA
AL,CAAnB,CAAD,KAAkC,EAAlD,K;K;4EAlJX,yB;MA6NA,0B;MA7NA,mC;QAKkB,kBAAT,oBAAL,SAAK,
C;QA6NL,QAAQ,gBA7Ne,KA6Nf,C;QACR,IAAI,gBA9NmB,KA8NnB,eAAiB,CAAjB,IAAsB,mBA9NH,KA8N
G,GAAa,WAAb,CAA1B,C;UAA6C,W;;QA9N7C,OA+NO,C;O;KApOX,C;kEAOA,4B;MAyOI,QAhOK,oBAAL,
SAAK,CAgOG,QAhOU,KAgOV,C;MAhOR,OAiOO,MAjOW,KAiON,KAAa,MAjOP,KAiOO,CAAD,KAAmB,K
AAM,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAlC,CAAX,CAAL,C;K;4EA/NX,4B;MAiHI,QA5GK,SA4GG,G
A5GoB,KA4GpB,I;MACR,IAAI,CA7GC,SA6GD,GA7GwB,KA6GxB,IAAiB,CAAjB,IAAsB,eA7GE,KA6GF,MA
7GrB,SA6GL,C;QAA6C,a;;MA7G7C,OA8GO,C;K;kEA5GX,yB;MAAA,0B;MAAA,mC;QAwHI,QA/GK,SA+GG
,GA/Ge,KA+Gf,I;QA/GR,OAAgC,OAgHzB,KAhHgB,KAgHX,GAAW,CAAC,CAAC,IAhHF,KAgHC,KAAmB,K
AAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAlD,KAhHyB,C;O;KATpC,C;4EAWA,4B;MA+FI,QA1F
K,SA0FG,GA1FoB,KA0FpB,I;MACR,IAAI,CA3FC,SA2FD,GA3FwB,KA2FxB,IAAiB,CAAjB,IAAsB,eA3FE,K
A2FF,MA3FrB,SA2FL,C;QAA6C,a;;MA3F7C,OA4FO,C;K;kEA1FX,yB;MAAA,4B;MAAA,mC;QAsGI,QA7FK,
SA6FG,GA7Fe,KA6Ff,I;QA7FR,OAAgC,QA8FzB,KA9FgB,KA8FX,GAAW,CAAC,CAAC,IA9FF,KA8FC,KAA
mB,KAAK,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAlD,KA9FyB,C;O;KATpC,C;4EAWA,4B;MA6EI,Q
AxEK,SAwEG,GAxEc,KAwEd,I;MACR,IAAI,CAzEC,SAyED,GAzEkB,KAyElB,IAAiB,CAAjB,IAAsB,eAzEJ,K
AyEI,MAzErB,SAyEL,C;QAA6C,a;;MAzE7C,OA0EO,C;K;kEAxEX,4B;MAoFI,QA3EK,SA2EG,GA3ES,KA2ET
,I;MA3ER,OA4EO,KA5EU,KA4EL,GAAW,CAAC,CAAC,IA5ER,KA4EO,KAAmB,KAAK,CAAC,CAAD,IAAL,
CAAnB,CAAD,KAAkC,EAAlD,K;K;4EA1EX,yB;MAqJA,0B;MArJA,mC;QAKkB,kBAAT,oBAAL,SAAK,C;QA
qJL,QAAQ,gBArJe,KAqJf,C;QACR,IAAI,gBAtJmB,KAsJnB,eAAiB,CAAjB,IAAsB,mBAtJH,KAsJG,GAAa,WA
Ab,CAA1B,C;UAA6C,W;;QAtJ7C,OAuJO,C;O;KA5JX,C;kEAOA,4B;MAiKI,QAxJK,oBAAL,SAAK,CAwJG,Q

AxJU,KAwJV,C;MAxJR,OAyJO,MAzJW,KAyJN,KAAa,MAzJP,KAyJO,CAAD,KAAmB,KAAM,CAAD,aAAL,
CAAnB,CAAD,YAAkC,EAAlC,CAAX,CAAL,C;K;2EAvJX,4B;MAyCI,QApCA,SAoCQ,GApCY,KAoCZ,I;MA
CR,IAAI,CArCJ,SAqCI,GArCgB,KAqChB,IAAiB,CAAjB,IAAsB,eArCN,KAqCM,MArC1B,SAqCA,C;QAA6C,a
;;MArC7C,OAsCO,C;K;iEApCX,yB;MAAA,0B;MAAA,mC;QAgDI,QAvCA,SAuCQ,GAvCO,KAuCP,I;QAvCR,
OAAwB,OAwCjB,KAxCQ,KAwCH,GAAW,CAAC,CAAC,IAxCV,KAwCS,KAAmB,KAAK,CAAC,CAAD,IAA
L,CAAnB,CAAD,KAAkC,EAAlD,KAxCiB,C;O;KAT5B,C;4EAWA,4B;MAuBI,QAlBA,SAkBQ,GAlBY,KAkBZ,
I;MACR,IAAI,CAnBJ,SAmBI,GAnBgB,KAmBhB,IAAiB,CAAjB,IAAsB,eAnBN,KAmBM,MAnB1B,SAmBA,C;
QAA6C,a;;MAnB7C,OAoBO,C;K;mEAlBX,yB;MAAA,4B;MAAA,mC;QA8BI,QArBA,SAqBQ,GArBO,KAqBP,I
;QArBR,OAAwB,QAsBjB,KAtBQ,KAsBH,GAAW,CAAC,CAAC,IAtBV,KAsBS,KAAmB,KAAK,CAAC,CAAD,
IAAL,CAAnB,CAAD,KAAkC,EAAlD,KAtBiB,C;O;KAT5B,C;4EAWA,4B;MAKI,QAAQ,YAAO,KAAP,I;MAC
R,IAAI,aAAS,KAAT,IAAiB,CAAjB,IAAsB,eAAI,KAAJ,MAAa,SAAvC,C;QAA6C,a;;MAC7C,OAAO,C;K;mEA
GX,4B;MASI,QAAQ,YAAO,KAAP,I;MACR,OAAO,KAAK,QAAW,CAAC,CAAC,IAAM,KAAP,KAAmB,KAA
K,CAAC,CAAD,IAAL,CAAnB,CAAD,KAAkC,EAAlD,K;K;4EAGX,yB;MAwEA,0B;MAxEA,mC;QAKkB,kBA
AT,oBAAL,SAAK,C;QAwEL,QAAQ,gBAxEe,KAwEf,C;QACR,IAAI,gBAzEmB,KAyEnB,eAAiB,CAAjB,IAAs
B,mBAzEH,KAyEG,GAAa,WAAb,CAA1B,C;UAA6C,W;;QAzE7C,OA0EO,C;O;KA/EX,C;kEAOA,4B;MAoFI,Q
A3EK,oBAAL,SAAK,CA2EG,QA3EU,KA2EV,C;MA3ER,OA4EO,MA5EW,KA4EN,KAAa,MA5EP,KA4EO,CA
AD,KAAmB,KAAM,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAlC,CAAX,CAAL,C;K;6EA1EX,yB;MAsDA,0B;
MAtDA,mC;QAKS,cAAe,oBAAN,KAAM,C;QAsDpB,QAtDA,SAsDQ,KAAO,OAAP,C;QACR,IAvDA,SAuDI,K
AAS,OAAT,eAAiB,CAAjB,IAAsB,mBAAI,OAAJ,GAvD1B,SAuD0B,CAA1B,C;UAA6C,W;;QAvD7C,OAwDO,
C;O;KA7DX,C;mEAOA,yB;MAAA,0B;MAAA,mC;QASS,cAAU,oBAAN,KAAM,C;QAyDf,QAzDA,SAyDQ,Q
AAO,OAAP,C;QAzDR,OAAyB,OA0DlB,MAAK,YAAa,MAAM,OAAN,CAAD,KAAmB,KAAM,CAAD,aAAL,
CAAnB,CAAD,YAAkC,EAAlC,CAAX,CAAL,CA1DkB,S;O;KAT7B,C;6EAWA,yB;MAoCA,0B;MApCA,mC;Q
AKS,cAAe,oBAAN,KAAM,C;QAoCpB,QApCA,SAoCQ,KAAO,OAAP,C;QACR,IArCA,SAqCI,KAAS,OAAT,e
AAiB,CAAjB,IAAsB,mBAAI,OAAJ,GArC1B,SAqC0B,CAA1B,C;UAA6C,W;;QArC7C,OAsCO,C;O;KA3CX,C;
mEAOA,yB;MAAA,4B;MAAA,mC;QASS,cAAU,oBAAN,KAAM,C;QAuCf,QAvCA,SAuCQ,QAAO,OAAP,C;Q
AvCR,OAAyB,QAwClB,MAAK,YAAa,MAAM,OAAN,CAAD,KAAmB,KAAM,CAAD,aAAL,CAAnB,CAAD,Y
AAkC,EAAlC,CAAX,CAAL,CAxCkB,S;O;KAT7B,C;6EAWA,yB;MAkBA,0B;MAlBA,mC;QAKS,cAAe,oBAA
N,KAAM,C;QAkBpB,QAlBA,SAkBQ,KAAO,OAAP,C;QACR,IAnBA,SAmBI,KAAS,OAAT,eAAiB,CAAjB,IAA
sB,mBAAI,OAAJ,GAnB1B,SAmB0B,CAA1B,C;UAA6C,W;;QAnB7C,OAoBO,C;O;KAzBX,C;mEAOA,4B;MAS
S,cAAU,oBAAN,KAAM,C;MAqBf,QArBA,SAqBQ,QAAO,OAAP,C;MArBR,OAsBO,MAAK,YAAa,MAAM,O
AAN,CAAD,KAAmB,KAAM,CAAD,aAAL,CAAnB,CAAD,YAAkC,EAAlC,CAAX,CAAL,CAtBkB,Q;K;6EAE7
B,yB;MAAA,0B;MAAA,mC;QAKI,QAAQ,cAAO,KAAP,C;QACR,IAAI,cAAS,KAAT,eAAiB,CAAjB,IAAsB,m
BAAI,KAAJ,GAAa,SAAb,CAA1B,C;UAA6C,W;;QAC7C,OAAO,C;O;KAPX,C;mEAUA,4B;MASI,QAAQ,iBAA
O,KAAP,C;MACR,OAAO,MAAK,UAAa,MAAM,KAAN,CAAD,KAAmB,KAAM,CAAD,aAAL,CAAnB,CAAD,
YAAkC,EAAlC,CAAX,CAAL,C;K;kEAGX,yB;M5GmqB2C,0B;M4GnqB3C,mC;QAWI,QAAQ,YAAO,K;QACJ,
iBAAS,G;QAAT,S;UAAsB,O5GupB0C,W4GvpB1C,C5GupB0C,C4GvpB1C,K5GupB0C,W4GvpBhC,K5GupBg
C,C;;Q4GvpB3E,OAAO,OAAgD,IAAI,KAApD,GAA+D,C;O;KAZ1E,C;mEAeA,yB;M5G0H6C,0B;M4G1H7C,m
C;QAqCI,QA1BK,SA0BG,GA1BY,K;QA2BT,iBAAK,G;QAAL,S;UAAY,O5GoF0B,W4GpF1B,C5GoF0B,C4Gp
F1B,K5GoF0B,W4G/G7B,K5G+G6B,C;;Q4G/GjD,OA2BO,OAAsC,IA3BzB,KA2Bb,GAAqD,C;O;KAtChE,C;mE
AaA,yB;M5G6G6C,0B;M4G7G7C,mC;QAwBI,QAbA,SAaQ,GAbO,K;QAcJ,iBAAK,G;QAAL,S;UAAY,O5GoF0
B,W4GpF1B,C5GoF0B,C4GpF1B,K5GoF0B,W4GlGlC,K5GkGkC,C;;Q4GlGjD,OAcO,OAAsC,IAd9B,KAcR,GA
AqD,C;O;KAzBhE,C;mEAaA,yB;M5GgG6C,0B;M4GhG7C,mC;QAWI,QAAQ,YAAO,K;QACJ,iBAAK,G;QAAL
,S;UAAY,O5GoF0B,W4GpF1B,C5GoF0B,C4GpF1B,K5GoF0B,W4GpFhB,K5GoFgB,C;;Q4GpFjD,OAAO,OAAs
C,IAAI,KAA1C,GAAqD,C;O;KAZhE,C;4ECtVA,yB;MAAA,8B;MAAA,4B;QAOyC,Q;QAAA,gFAAoB,C;O;KA
P7D,C;ICM0B,4C;MA+CtB,qC;MA/CuB,kB;MAAgB,kB;MAAgB,kB;MAMvD,iBAAsB,iBAAU,UAAV,EAAiB,
UAAjB,EAAwB,UAAxB,C;K;0CAEtB,+B;MjNWA,IAAI,EiNViB,CAAT,sBAAY,GAAZ,KAA4C,CAAT,sBAAY
,GAA/C,MAA+E,CAAT,sBAAY,GAAlF,CjNUR,CAAJ,C;QACI,ciNVI,2E;QjNWJ,MAAM,gCAAyB,OAAQ,WA
AjC,C;;MiNTN,OAAO,CAAA,KAAM,IAAI,EAAV,KAAgB,KAAM,IAAI,CAA1B,IAA+B,KAA/B,I;K;uCAGX,

Y;MAGkC,OAAE,UAAF,oBAAS,UAAT,SAAgB,U;K;qCAElD,iB;MAEwB,gB;MADpB,IAAI,SAAS,KAAb,C;Q
AAoB,OAAO,I;MACP,iE;MAAD,mB;QAA6B,OAAO,K;;MAAvD,mBAAmB,M;MACnB,OAAO,IAAK,UAAL,K
AAgB,YAAa,U;K;uCAGxC,Y;MAA+B,qB;K;8CAE/B,iB;MAAoD,wBAAU,KAAM,UAAhB,I;K;gDAEpD,wB;M
AKI,OAAA,IAAK,MAAL,GAAa,KAAb,KAAuB,IAAK,MAAL,KAAc,KAAd,IACf,IAAK,MAAL,IAAc,KADtB,
C;K;gDAGJ,+B;MAKI,OAAA,IAAK,MAAL,GAAa,KAAb,KAAuB,IAAK,MAAL,KAAc,KAAd,KACd,IAAK,M
AAL,GAAa,KAAb,KAAsB,IAAK,MAAL,KAAc,KAAd,IACf,IAAK,MAAL,IAAc,KADrB,CADc,CAAvB,C;K;IA
IJ,mC;MAAA,uC;MACI,2BAIuC,G;MAEvC,eAIoC,uCAA0B,M;K;;;IAXlE,+C;MAAA,8C;QAAA,6B;;MAAA,uC
;K;;IA9CA,iD;MAAA,uD;MAG6C,0BAAK,KAAL,EAAY,KAAZ,EAAmB,CAAnB,C;MAH7C,Y;K;IA6DJ,qC;M
AAA,yC;K;8CAEI,Y;MAC2B,yBAAc,CAAd,EAAiB,CAAjB,EAAoB,EAApB,C;K;;;IAH/B,iD;MAAA,gD;QAAA
,+B;;MAAA,yC;K;4FCxDI,yB;MAAA,2D;MAAA,4B;QAAQ,MAAM,6BAAoB,6BAApB,C;O;KAAd,C;;;;ICSJ,u
B;MAG2C,+BAAoB,KAApB,C;K;4EAE3C,wC;MAO4F,sB;K;IAE5F,6C;MAAA,e;MAAA,iB;MAAA,uB;K;IAA
A,2C;MAAA,8C;O;MAKI,wF;MAKA,sF;MAMA,wE;K;;IAXA,yD;MAAA,iC;MAAA,iD;K;;IAKA,wD;MAAA,i
C;MAAA,gD;K;;IAMA,iD;MAAA,iC;MAAA,yC;K;;IAhBJ,uC;MAAA,iJ;K;;IAAA,4C;MAAA,a;aAAA,c;UAAA,
sD;aAAA,a;UAAA,qD;aAAA,M;UAAA,8C;;UAAA,gE;;K;;IAyBA,+B;MAAA,mC;K;;;IAAA,2C;MAAA,0C;QA
AA,yB;;MAAA,mC;K;IAGoC,qC;MAChC,qBAAsC,W;MACtC,gBAA2B,iC;K;uFAGvB,Y;MAMW,Q;MALP,IA
AI,kBAAW,iCAAf,C;QACI,gBAAS,mC;QACT,qBAAc,I;;MAGlB,OAAO,gF;K;6CAGf,Y;MAAwC,yBAAW,iC;
K;wCAEnD,Y;MAAkC,OAAI,oBAAJ,GAA2B,SAAN,UAAM,CAA3B,GAA2C,iC;K;8CAE7E,Y;MAAkC,+BAA
oB,UAApB,C;K;;IAGG,oC;MAAC,4B;K;wEAAA,Y;MAAA,2B;K;kDAEtC,Y;MAAwC,W;K;6CAExC,Y;MAAk
C,OAAM,SAAN,UAAM,C;K;;oFC2C5C,yB;MAAA,gD;MAAA,4B;QAM6C,OAAmB,aAAlB,YAAY,GAAM,C;
O;KANhE,C;oGAQA,yB;MhH7FA,4B;MgH6FA,4B;QAMqD,OhH7FM,YgH6FL,YAAY,GhH7FP,CgH6FN,GAA
6C,EAA7C,I;O;KANrD,C;sGAQA,yB;MAAA,kE;MAAA,4B;QAMsD,OAAmB,sBAAlB,YAAW,GAAO,C;O;KA
NzE,C;8FAQA,yB;MAAA,0D;MAAA,0B;MAAA,4B;QAOmD,OAAuC,OAApB,kBAAlB,YAAY,GAAM,CAAo
B,C;O;KAP1F,C;4FASA,yB;MAAA,wD;MAAA,0B;MAAA,4B;QAOkD,OAA2B,OAAnB,iBAAR,SAAQ,CAAm
B,C;O;KAP7E,C;IAUA,2C;MAaI,OAA+E,OAA9E,SAAQ,KAAI,WAAa,CAAjB,CAAR,GAAkD,CAAlB,YAAY,
GAAM,MAAK,CAAL,IAAU,WAAa,CAAvB,CAA4B,C;K;IAEnF,4C;MAaI,OAA+E,OAA9E,SAAQ,IAAI,CAAJ,
IAAS,WAAa,CAAtB,CAAR,GAAwD,CAAlB,YAAY,GAAM,OAAK,WAAa,CAAlB,CAAsB,C;K;oFAEnF,yB;M
AAA,gD;MAAA,4B;QAM8C,OAAqB,aAApB,YAAY,KAAQ,C;O;KANnE,C;oGAQA,yB;MhHtKA,4B;MgHsKA
,4B;QAOI,OhHvKuD,YgHuKtD,YAAY,KhHvK0C,CgHuKvD,GAA+C,EAA/C,I;O;KAPJ,C;sGASA,yB;MAAA,k
E;MAAA,4B;QAMuD,OAAqB,sBAApB,YAAW,KAAS,C;O;KAN5E,C;8FAQA,yB;MAAA,0D;MAAA,4B;MAA
A,4B;QAOqD,OAAyC,QAApB,kBAApB,YAAY,KAAQ,CAAoB,C;O;KAP9F,C;4FASA,yB;MAAA,wD;MAAA,
4B;MAAA,4B;QAOoD,OAA2B,QAAnB,iBAAR,SAAQ,CAAmB,C;O;KAP/E,C;IAUA,2C;MAaI,OAAoF,QAAnF
,SAAQ,KAAI,WAAa,EAAjB,CAAR,GAAqD,CAApB,YAAY,KAAQ,MAAK,EAAL,IAAW,WAAa,EAAxB,CAA
8B,C;K;IAExF,4C;MAaI,OAAoF,QAAnF,SAAQ,IAAI,EAAJ,IAAU,WAAa,EAAvB,CAAR,GAA4D,CAApB,YA
AY,KAAQ,OAAK,WAAa,EAAlB,CAAuB,C;K;0EpNlRxF,yB;MAaA,kF;MAbA,wB;QAuBI,IAAI,CAbI,KAaR,C;
UACI,cAda,qB;UAeb,MAAM,8BAAyB,OAAQ,WAAjC,C;;O;KAzBd,C;0EAaA,yB;MAAA,kF;MAAA,qC;QAUI,
IAAI,CAAC,KAAL,C;UACI,cAAc,a;UACd,MAAM,8BAAyB,OAAQ,WAAjC,C;;O;KAZd,C;sFAgBA,yB;MAW
A,kF;MAXA,wB;QAQW,yB;QAeP,IAfsB,KAelB,QAAJ,C;UACI,cAhB2B,0B;UAiB3B,MAAM,8BAAyB,OAAQ,
WAAjC,C;;UAEN,wBAnBkB,K;;QAAtB,4B;O;KARJ,C;wFAWA,yB;MAAA,kF;MAAA,qC;QAYI,IAAI,aAAJ,C;
UACI,cAAc,a;UACd,MAAM,8BAAyB,OAAQ,WAAjC,C;;UAEN,OAAO,K;;O;KAhBf,C;oEAoBA,yB;MAaA,4E;
MAbA,wB;QAuBI,IAAI,CAbE,KAaN,C;UACI,cAdW,e;UAeX,MAAM,2BAAsB,OAAQ,WAA9B,C;;O;KAzBd,C
;sEAaA,yB;MAAA,4E;MAAA,qC;QAUI,IAAI,CAAC,KAAL,C;UACI,cAAc,a;UACd,MAAM,2BAAsB,OAAQ,W
AA9B,C;;O;KAZd,C;kFAgBA,yB;MAcA,4E;MAdA,wB;QAWW,uB;QAeP,IAfoB,KAehB,QAAJ,C;UACI,cAhBy
B,0B;UAiBzB,MAAM,2BAAsB,OAAQ,WAA9B,C;;UAEN,sBAnBgB,K;;QAApB,0B;O;KAXJ,C;oFAcA,yB;MA
AA,4E;MAAA,qC;QAYI,IAAI,aAAJ,C;UACI,cAAc,a;UACd,MAAM,2BAAsB,OAAQ,WAA9B,C;;UAEN,OAAO
,K;;O;KAhBf,C;oEAqBA,yB;MAAA,4E;MAAA,0B;QAMiD,MAAM,2BAAsB,OAAQ,WAA9B,C;O;KANvD,C;I
wCnHiC,uB;MA2D7B,8B;MA1DA,kB;K;mFAS8B,Y;MAAQ,iD;K;mFAMR,Y;MAAQ,gD;K;wFAItC,yB;MAAA,
gB;MAAA,8B;MAAA,mB;QAWgB,Q;QADR,mB;UADJ,OACiB,I;;UADjB,OAEY,2E;O;KAXhB,C;uCAcA,Y;M
AQQ,kBADE,UACF,kB;QADJ,OACkB,UAAM,U;;QADxB,OAEY,I;K;gCAGhB,Y;MAOQ,kBADE,UACF,kB;Q

ADJ,OACkB,UAAM,W;;QADxB,OAEY,sBAAU,UAAV,O;K;IAKhB,4B;MAAA,gC;K;wHAKI,yB;MAAA,iC;M
AAA,wB;QAOI,uBAAO,KAAP,C;O;KAPJ,C;wHASA,yB;MAAA,kD;MAAA,iC;MAAA,4B;QAOI,uBAAO,cAA
c,SAAd,CAAP,C;O;KAPJ,C;;;IAdJ,wC;MAAA,uC;QAAA,sB;;MAAA,gC;K;IAwBsB,mC;MAClB,0B;K;sCAGA,i
B;MAA4C,+CAAoB,uBAAa,KAAM,UAAnB,C;K;wCAChE,Y;MAA+B,OAAU,SAAV,cAAU,C;K;wCACzC,Y;
MAAkC,oBAAU,cAAV,M;K;;;;;;gCA/F1C,Y;MAAA,c;MAOI,sD;MAPJ,a;K;8BAAA,iB;MAAA,2IAOI,sCAPJ,G;
K;IAmGA,kC;MAOI,OAAO,mBAAQ,SAAR,C;K;IAEX,mC;MAQI,IAAI,8CAAJ,C;QAA6B,MAAM,eAAM,U;K;
gFAG7C,yB;MAAA,4B;MAAA,qB;MAxCQ,kD;MAwCR,wB;QAOW,Q;;UACI,OAlDH,WAkDW,OAlDX,C;;UA
mDN,gC;YACS,OA3CH,WAAO,cA2CI,CA3CJ,CAAP,C;;YAwCD,O;;QAAP,W;O;KAPJ,C;kFAcA,yB;MAAA,4
B;MAAA,qB;MAtDQ,kD;MAsDR,mC;QAOW,Q;;UACI,OAhEH,WAgEW,gBAhEX,C;;UAiEN,gC;YACS,OAzD
H,WAAO,cAyDI,CAzDJ,CAAP,C;;YAsDD,O;;QAAP,W;O;KAPJ,C;8EAgBA,yB;MAAA,oD;MAAA,gB;MAAA,
8B;MAAA,4B;QAUW,Q;QADP,yB;QACA,OAAO,gF;O;KAVX,C;+EAaA,yB;MAAA,gB;MAAA,8B;MAAA,uC;
QAegB,UADL,M;QAAM,gBAAgB,2B;QACzB,sB;UAAQ,yF;;UACA,mBAAU,SAAV,C;QAFZ,a;O;KAdJ,C;kFA
oBA,yB;MAAA,gB;MAAA,8B;MAAA,0C;QAUW,Q;QADP,IAAI,mBAAJ,C;UAAe,OAAO,Y;QACtB,OAAO,gF;
O;KAVX,C;qEAaA,yB;MAAA,gB;MAAA,8B;MAAA,kD;QAiB0B,UADf,M;QAAM,gBAAgB,2B;QACzB,sB;U
AAQ,mBAAU,gFAAV,C;;UACA,mBAAU,SAAV,C;QAFZ,a;O;KAhBJ,C;mEAwBA,yB;MAAA,4B;MAAA,gB;
MAAA,8B;MAAA,uC;YAe8C,I;YADnC,M;QACH,wB;UAAa,gB;UAAO,SA7JhB,WA6JwB,UAAU,gFAAV,CA7
JxB,C;;UA8JI,oBAAO,eAAP,C;QAFZ,a;O;KAdJ,C;gFAoBA,yB;MAAA,gB;MAAA,8B;MAAA,iC;MA1GA,qB;
MAtDQ,kD;MAgKR,uC;QAWW,Q;QACH,wB;UA/GG,U;;YA+GkC,U;YA9G9B,SAhEH,gBA8KuB,UAAU,sFA
AV,CA9KvB,C;;YAiEN,gC;cACS,SAzDH,gBAAO,cAyDI,CAzDJ,CAAP,C;;cAsDD,O;;UA+GU,a;;UACL,uBAA
O,eAAP,C;QAFZ,W;O;KAXJ,C;wEAiBA,yB;MAAA,4B;MAAA,uC;QAcW,Q;QAAM,gBAAgB,2B;QACzB,sB;U
AAQ,gB;;UACO,OAnMX,WAmMmB,UAAU,SAAV,CAnMnB,C;;QAiMR,W;O;KAdJ,C;wFAoBA,yB;MA/IA,4
B;MAAA,qB;MAtDQ,kD;MAqMR,uC;QAWW,Q;QAAM,gBAAgB,2B;QACzB,sB;UAAQ,gB;;UApJL,U;;YACI,
SAhEH,WAoNkB,oBApNlB,C;;YAiEN,gC;cACS,SAzDH,WAAO,cAyDI,CAzDJ,CAAP,C;;cAsDD,O;;UAqJK,a;;
QAFZ,W;O;KAXJ,C;4EAmBA,6B;MAUI,Q;MAAA,iD;QAAyB,Y;;MACzB,OAAO,S;K;4EAGX,yB;MAAA,gB;
MAAA,8B;MAAA,oC;QAU0B,Q;QAAtB,IAAI,mBAAJ,C;UAAe,OAAO,gFAAP,C;;QACf,OAAO,S;O;KAXX,C;
IrCtTgC,sC;MAAC,uB;QAAA,UAAkB,kC;mBAA4C,O;;K;;0DAE/F,yB;MAAA,2D;MAAA,mB;QAKoC,MAAM,
8B;O;KAL1C,C;oEAOA,yB;MAAA,2D;MAAA,yB;QAMkD,MAAM,6BAAoB,sCAAmC,MAAvD,C;O;KANxD,
C;gEAUA,iB;MAUI,OAAO,O;K;kEAGX,4B;MAUI,OAAO,gB;K;oEAGX,2B;MAUI,OAAgB,MAAT,QAAS,C;K
;oEAGpB,4B;MAUI,gB;MACA,OAAO,S;K;kEAGX,4B;MAWI,MAAM,SAAN,C;MACA,OAAO,S;K;kEAGX,4B
;MAUI,OAAO,MAAM,SAAN,C;K;sEAGX,gC;MAWI,OAAW,UAAU,SAAV,CAAJ,GAAAqB,SAArB,GAA+B,I;K
;8EAG1C,gC;MAWI,OAAW,CAAC,UAAU,SAAV,CAAL,GAAAsB,SAAtB,GAAgC,I;K;wEAG3C,yB;MAWI,iBA
Ac,CAAd,UAAsB,KAAtB,U;QACI,OAAO,KAAP,C;;K;wEkNjJR,iB;MAGkF,Y;K;ICa9C,6B;MAChC,kB;MACA
,oB;K;8BAGA,Y;MAGyC,aAAG,UAAH,UAAW,WAAX,M;K;;gCAvB7C,Y;MAgBI,iB;K;gCAhBJ,Y;MAiBI,kB;
K;kCAjBJ,yB;MAAA,gBAgBI,qCAhBJ,EAiBI,wCAjBJ,C;K;8BAAA,Y;MAAA,c;MAgBI,sD;MACA,uD;MAjBJ,
a;K;4BAAA,iB;MAAA,4IAgBI,sCAhBJ,IAiBI,wCAjBJ,I;K;IA0BA,6B;MAMoD,gBAAK,SAAL,EAAW,IAAX,C;
K;IAEpD,8B;MAI8C,iBAAO,eAAP,EAAc,gBAAd,E;K;IAiBD,sC;MACzC,kB;MACA,oB;MACA,kB;K;gCAGA,
Y;MAGyC,aAAG,UAAH,UAAW,WAAX,UAAoB,UAApB,M;K;;kCAxB7C,Y;MAgBI,iB;K;kCAhBJ,Y;MAiBI,k
B;K;kCAjBJ,Y;MAkBI,iB;K;oCAlBJ,gC;MAAA,kBAgBI,qCAhBJ,EAiBI,wCAjBJ,EAkBI,qCAlBJ,C;K;gCAAA,
Y;MAAA,c;MAgBI,sD;MACA,uD;MACA,sD;MAlBJ,a;K;8BAAA,iB;MAAA,4IAgBI,sCAhBJ,IAiBI,wCAjBJ,IA
kBI,sCAlBJ,I;K;IA2BA,8B;MAImD,iBAAO,eAAP,EAAc,gBAAd,EAAsB,eAAtB,E;K;InOlE1B,qB;MAErB,6B;M
AFkG,gB;K;IAElG,2B;MAAA,+B;MACI,iBAGoC,UAAM,CAAN,C;MAEpC,iBAGoC,UAAM,MAAN,C;MAEpC
,kBAGmC,C;MAEnC,iBAGkC,C;K;;;IAnBtC,uC;MAAA,sC;QAAA,qB;;MAAA,+B;K;kGAsBA,iB;MAOmE,OA
Aa,0BAqP1C,SAAL,GAAiB,GArP8B,EAAU,KAqPpD,KAAL,GAAiB,GArP8B,C;K;sGAEhF,iB;MAM2D,OAAa,
0BA6OlC,SAAL,GAAiB,GA7OsB,EAAU,KE8O5C,KAAL,GAAiB,KF9OsB,C;K;sGAExE,yB;MAoQA,6B;MCrQ
A,8C;MDCA,wB;QAMyD,OCAS,YAAiB,CDuQhD,cAAU,SAAL,GAAiB,GAAtB,CCvQgD,MAAjB,EDAe,KCA
c,KAA7B,C;O;KDNlE,C;sGAQA,yB;MAsQA,WAS6D,wB;MAT7D,+B;MkBvQA,gD;MlBCA,wB;QAM0D,OkB
AS,aAAkB,ClByQhD,eAAW,oBAAL,SAAK,CAAL,UAAN,CkBzQgD,MAAlB,ElBAgB,KkBAc,KAA9B,C;O;Kl
BNnE,C;4FAQA,yB;MAoPA,6B;MApPA,wB;QAEsD,OCMD,cAAU,CDqP5B,cAAU,SAAL,GAAiB,GAAtB,CCr

P4B,MAAK,GAAW,CDqP5C,cA3PsC,KA2P5B,KAAL,GAAiB,GAAtB,CCrP4C,MAAX,IAAf,C;O;KDRrD,C;4F
AGA,yB;MAiPA,6B;MAjPA,wB;QAEuD,OCGF,cAAU,CDqP5B,cAAU,SAAL,GAAiB,GAAtB,CCrP4B,MAAK,
GAAW,CCsP5C,cFzPuC,KEyP7B,KAAL,GAAiB,KAAtB,CDtP4C,MAAX,IAAf,C;O;KDLrD,C;4FAGA,yB;MA8
OA,6B;MA9OA,wB;QAEqD,OCAA,cAAU,CDqP5B,cAAU,SAAL,GAAiB,GAAtB,CCrP4B,MAAK,GDAI,KCA
O,KAAX,IAAf,C;O;KDFrD,C;4FAGA,yB;MAqPA,WAS6D,wB;MAT7D,+B;MArPA,wB;QAEuD,OkBAA,eAA
W,ClB4P7B,eAAW,oBAAL,SAAK,CAAL,UAAN,CkB5P6B,MAAK,KlBAI,KkBAO,KAAX,CAAhB,C;O;KlBFv
D,C;8FAIA,yB;MAuOA,6B;MAvOA,wB;QAEuD,OCMD,cAAU,CDwO7B,cAAU,SAAL,GAAiB,GAAtB,CCxO6
B,MAAK,GAAY,CDwO9C,cA9OwC,KA8O9B,KAAL,GAAiB,GAAtB,CCxO8C,MAAZ,IAAf,C;O;KDRtD,C;8F
AGA,yB;MAoOA,6B;MApOA,wB;QAEwD,OCGF,cAAU,CDwO7B,cAAU,SAAL,GAAiB,GAAtB,CCxO6B,MA
AK,GAAY,CCyO9C,cF5OyC,KE4O/B,KAAL,GAAiB,KAAtB,CDzO8C,MAAZ,IAAf,C;O;KDLtD,C;8FAGA,yB;
MAiOA,6B;MAjOA,wB;QAEsD,OCAA,cAAU,CDwO7B,cAAU,SAAL,GAAiB,GAAtB,CCxO6B,MAAK,GDAK,
KCAO,KAAZ,IAAf,C;O;KDFtD,C;8FAGA,yB;MAwOA,WAS6D,wB;MAT7D,+B;MAxOA,wB;QAEwD,OkBAA
,eAAW,ClB+O9B,eAAW,oBAAL,SAAK,CAAL,UAAN,CkB/O8B,MAAK,UlBAK,KkBAO,KAAZ,CAAhB,C;O;
KlBFxD,C;8FAIA,yB;MA0NA,6B;MA1NA,wB;QAEuD,OCMD,cAAe,YAAL,CD2N7B,cAAU,SAAL,GAAiB,G
AAtB,CC3N6B,MAAK,EAAY,CD2N9C,cAjOwC,KAiO9B,KAAL,GAAiB,GAAtB,CC3N8C,MAAZ,CAAf,C;O;
KDRtD,C;8FAGA,yB;MAuNA,6B;MAvNA,wB;QAEwD,OCGF,cAAe,YAAL,CD2N7B,cAAU,SAAL,GAAiB,GA
AtB,CC3N6B,MAAK,EAAY,CC4N9C,cF/NyC,KE+N/B,KAAL,GAAiB,KAAtB,CD5N8C,MAAZ,CAAf,C;O;KD
LtD,C;8FAGA,yB;MAoNA,6B;MApNA,wB;QAEsD,OCAA,cAAe,YAAL,CD2N7B,cAAU,SAAL,GAAiB,GAAtB
,CC3N6B,MAAK,EDAK,KCAO,KAAZ,CAAf,C;O;KDFtD,C;8FAGA,yB;MA2NA,WAS6D,wB;MAT7D,+B;MA
3NA,wB;QAEwD,OkBAA,eAAW,ClBkO9B,eAAW,oBAAL,SAAK,CAAL,UAAN,CkBlO8B,MAAK,UlBAK,Kk
BAO,KAAZ,CAAhB,C;O;KlBFxD,C;0FAIA,yB;MA6MA,6B;MCvMA,4C;MDNA,wB;QAEqD,OCMD,WD8MjB
,cAAU,SAAL,GAAiB,GAAtB,CC9MiB,ED8MjB,cApNoC,KAoN1B,KAAL,GAAiB,GAAtB,CC9MiB,C;O;KDRp
D,C;0FAGA,yB;MA0MA,6B;MCvMA,4C;MDHA,wB;QAEsD,OCGF,WD8MjB,cAAU,SAAL,GAAiB,GAAtB,C
C9MiB,EC+MjB,cFlNqC,KEkN3B,KAAL,GAAiB,KAAtB,CD/MiB,C;O;KDLpD,C;0FAGA,yB;MAuMA,6B;MC
vMA,4C;MDAA,wB;QAEoD,OCAA,WD8MjB,cAAU,SAAL,GAAiB,GAAtB,CC9MiB,EDAkB,KCAlB,C;O;KDF
pD,C;0FAGA,yB;MA8MA,WAS6D,wB;MAT7D,+B;MkB9MA,8C;MlBAA,wB;QAEsD,OkBAA,YlBqNjB,eAA
W,oBAAL,SAAK,CAAL,UAAN,CkBrNiB,ElBAmB,KkBAnB,C;O;KlBFtD,C;0FAIA,yB;MAgMA,6B;MClLA,kD
;MDdA,wB;QAMqD,OCcD,cDqLjB,cAAU,SAAL,GAAiB,GAAtB,CCrLiB,EDqLjB,cAnMoC,KAmM1B,KAAL,
GAAiB,GAAtB,CCrLiB,C;O;KDpBpD,C;0FAOA,yB;MAyLA,6B;MClLA,kD;MDPA,wB;QAMsD,OCOF,cDqLjB
,cAAU,SAAL,GAAiB,GAAtB,CCrLiB,ECsLjB,cF7LqC,KE6L3B,KAAL,GAAiB,KAAtB,CDtLiB,C;O;KDbpD,C;
0FAOA,yB;MAkLA,6B;MClLA,kD;MDAA,wB;QAMoD,OCAA,cDqLjB,cAAU,SAAL,GAAiB,GAAtB,CCrLiB,
EDAkB,KCAlB,C;O;KDNpD,C;0FAOA,yB;MAqLA,WAS6D,wB;MAT7D,+B;MkBrLA,oD;MlBAA,wB;QAMsD
,OkBAA,elBwLjB,eAAW,oBAAL,SAAK,CAAL,UAAN,CkBxLiB,ElBAmB,KkBAnB,C;O;KlBNtD,C;oGAQA,y
B;MAmKA,6B;MCvMA,4C;MDoCA,wB;QAMiD,OCxCG,WD8MjB,cAAU,SAAL,GAAiB,GAAtB,CC9MiB,ED
8MjB,cAtKqC,KAsK3B,KAAL,GAAiB,GAAtB,CC9MiB,C;O;KDkCpD,C;oGAOA,yB;MA4JA,6B;MCvMA,4C;
MD2CA,wB;QAMkD,OC/CE,WD8MjB,cAAU,SAAL,GAAiB,GAAtB,CC9MiB,EC+MjB,cFhKsC,KEgK5B,KAA
L,GAAiB,KAAtB,CD/MiB,C;O;KDyCpD,C;oGAOA,yB;MAqJA,6B;MCvMA,4C;MDkDA,wB;QAMgD,OCtDI,
WD8MjB,cAAU,SAAL,GAAiB,GAAtB,CC9MiB,EDsDmB,KCtDnB,C;O;KDgDpD,C;oGAOA,yB;MAwJA,WAS
6D,wB;MAT7D,+B;MkB9MA,8C;MlBsDA,wB;QAMkD,OkB1DI,YlBqNjB,eAAW,oBAAL,SAAK,CAAL,UAAN
,CkBrNiB,ElB0DoB,KkB1DpB,C;O;KlBoDtD,C;0FAQA,yB;MAsIA,6B;MClLA,kD;MDiPJ,0B;MAAAA,+B;MAr
MI,wB;QAQ6C,OAwMR,eAAW,OCtPI,cDqLjB,cAAU,SAAL,GAAiB,GAAtB,CCrLiB,EDqLjB,cAvI4B,KAuIlB,
KAAL,GAAiB,GAAtB,CCrLiB,CA4Lf,KD0DW,CAAX,C;O;KAhNrC,C;0FASA,yB;MA6HA,6B;MClLA,kD;MC
kPJ,4B;MAAAA,iC;MF7LI,wB;QAQ+C,OEgMR,gBAAY,QDvPC,cDqLjB,cAAU,SAAL,GAAiB,GAAtB,CCrLiB,E
CsLjB,cF/H8B,KE+HpB,KAAL,GAAiB,KAAtB,CDtLiB,CAsMb,KCiDY,CAAZ,C;O;KFxMvC,C;0FASA,yB;MA
oHA,6B;MClLA,kD;MD8DA,wB;QAQ2C,OChES,cDqLjB,cAAU,SAAL,GAAiB,GAAtB,CCrLiB,EDgES,KChET
,C;O;KDwDpD,C;0FASA,yB;MAqHA,WAS6D,wB;MAT7D,+B;MkBrLA,oD;MlBgEA,wB;QAQ6C,OkBlES,elB
wLjB,eAAW,oBAAL,SAAK,CAAL,UAAN,CkBxLiB,ElBkEU,KkBlEV,C;O;KlB0DtD,C;0EAUA,yB;MAAAA,0B;
MAAAA,+B;MAAAA,mB;QAAM0C,sBAAAW,OAAAL,SAAAK,KAAAX,C;O;KAN1C,C;0EAQA,yB;MAAAA,0B;MAAAA,+

B;MAAA,mB;QAM0C,sBAAW,OAAL,SAAK,KAAX,C;O;KAN1C,C;kGAQA,yB;MAAA,8C;MAiFA,6B;MAjF
A,wB;QAE8D,0BAwF3B,cAAU,SAAL,GAAiB,GAAtB,CAxF2B,EAwF3B,cAxFoD,KAwF1C,KAAL,GAAiB,GA
AtB,CAxF2B,C;O;KAF9D,C;wGAIA,yB;MAAA,yC;MA6EA,6B;MA7EA,wB;QAQiE,aA8E9B,cAAU,SAAL,GA
AiB,GAAtB,CA9E8B,EA8E9B,cA9EkD,KA8ExC,KAAL,GAAiB,GAAtB,CA9E8B,C;O;KARjE,C;0FAUA,yB;M
AAA,+B;M8LlPJ,0B;M9LkPI,wB;QAEmD,sB8LjPgC,O9LiP1B,IAAK,K8LjPX,G9LiPoB,KAAM,K8LjPM,C9LiP
hC,C;O;KAFnD,C;wFAGA,yB;MAAA,+B;M8LhPJ,0B;M9LgPI,wB;QAEkD,sB8L/O+B,O9L+OzB,IAAK,K8L/O
X,G9L+OmB,KAAM,K8L/OM,C9L+O/B,C;O;KAFlD,C;0FAGA,yB;MAAA,+B;M8L9OJ,0B;M9L8OI,wB;QAE
mD,sB8L7OgC,O9L6O1B,IAAK,K8L7OX,G9L6OoB,KAAM,K8L7OM,C9L6OhC,C;O;KAFnD,C;0EAGA,yB;M
AAA,+B;M8L5OJ,0B;M9L4OI,mB;QAEiC,sB8L3OqB,OAAP,C9L2OR,S8L3Oe,C9L2OrB,C;O;KAFjC,C;gFAIA,
Y;MASmC,gB;K;kFACnC,yB;M8LpPJ,4B;M9LoPI,mB;QASqC,O8L1PiD,Q9L0P5C,S8L1PY,G9L0PE,G8L1P8B,
C;O;K9LiPtF,C;8EAUA,Y;MASiC,OAAK,SAAL,GAAiB,G;K;gFAClD,yB;MAAA,WASqD,wB;MATrD,mB;QA
SmC,OAAK,oBAAL,SAAK,CAAL,U;O;KATnC,C;kFAWA,Y;MAEqC,W;K;oFACrC,yB;MAAA,iC;M8LtRJ,4B;
M9LsRI,mB;QASuC,uB8L5R+C,Q9L4RnC,S8L5RG,G9L4RW,G8L5RqB,C9L4R/C,C;O;KATvC,C;gFAUA,yB;
MAAA,6B;MAAA,mB;QASmC,qBAAU,SAAL,GAAiB,GAAtB,C;O;KATnC,C;kFAUA,yB;MAAA,WAS6D,wB;
MAT7D,+B;MAAA,mB;QASqC,sBAAW,oBAAL,SAAK,CAAL,UAAN,C;O;KATrC,C;kFAWA,Y;MAMqC,OAp
DC,SAAL,GAAiB,G;K;oFAqDlD,Y;MAMuC,OA3DD,SAAL,GAAiB,G;K;+BA6DlD,Y;MAAyC,OAAQ,CA7DX,
SAAL,GAAiB,GA6DD,Y;K;;;;;+BApVrD,Y;MAAA,c;MAGsG,qD;MAHtG,a;K;6BAAA,iB;MAAA,2IAGsG,oCA
HtG,G;K;wEAwVA,yB;MAAA,+B;MAAA,4B;QAU0C,sBAAM,SAAN,C;O;KAV1C,C;0EAWA,yB;MAAA,0B;
MAAA,+B;MAAA,4B;QAW2C,sBAAW,OAAL,SAAK,CAAX,C;O;KAX3C,C;0EAYA,yB;MAAA,0B;MAAA,+
B;MAAA,4B;QAWyC,sBAAW,OAAL,SAAK,CAAX,C;O;KAXzC,C;0EAYA,yB;MAAA,0B;MAAA,+B;MAAA,
4B;QAW0C,sBAAW,OAAL,SAAK,SAAX,C;O;KAX1C,C;IiCxXA,6B;MACqB,sB;K;uCAKjB,iB;MAM6C,OjCm
VP,UiCnVO,aAAQ,KAAR,CjCmVP,C;K;uCiCjVtC,wB;MAOI,aAAQ,KAAR,IAAiB,KjC2Oc,K;K;kFiCvOL,Y;M
AAQ,OAAA,YAAQ,O;K;oCAE9C,Y;MAC8E,+BAAS,YAAT,C;K;IAExD,oC;MAAC,oB;MACnB,eAAoB,C;K;4
CACpB,Y;MAAyB,sBAAQ,YAAM,O;K;yCACvC,Y;MAAoD,Q;MAA9B,IAAI,eAAQ,YAAM,OAAlB,C;QAAA,
OjC8TY,UiC9TY,aAAM,mBAAN,EAAM,2BAAN,OjC8TZ,C;;QiC9T0C,MAAM,2BAAuB,YAAM,WAA7B,C;K;
;0CAGtF,mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,K;MAExC,OAAe,WAAR,YAAQ,E
AAS,OjCsNO,KiCtNhB,C;K;+CAGnB,oB;MACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;;QdioDvB,U;QADhB,IAAI
,wCAAsB,mBAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,6B;QAAhB,OAAgB,gBAAhB,C;UAAgB,2B;UcjoD6B,
2BdioDR,OcjoDQ,Q;UAAA,W;YAAuB,oBAAR,YAAQ,EdioD/B,OnB/6CF,KiClNiC,C;;UdioD9C,IAAI,OAAJ,C;
YAAyB,aAAO,K;YAAP,e;;;QAC/C,aAAO,I;;;McloDH,iB;K;mCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,KA
AqB,C;K;;IA9CvD,sC;MAAA,oD;MACgC,uBAAK,cAAU,IAAV,CAAL,C;MADhC,Y;K;;;;oCAPJ,Y;MAAA,OA
KqB,qDALrB,M;K;oCAAA,Y;MAAA,c;MAKqB,wD;MALrB,a;K;kCAAA,iB;MAAA,2IAKqB,0CALrB,G;K;gFA
wDA,yB;MAAA,yC;MAWsC,yC;QAAA,wB;UAAW,OAAA,aAAK,KAAL,CjCiMV,K;S;O;MiC5MvC,6B;QAWI,
OAAO,oBAAW,+BAAU,IAAV,GAAgB,uBAAhB,CAAX,C;O;KAXX,C;kFAcA,oB;MAGqE,e;K;IhCrE7C,oB;M
AEpB,4B;MAFiG,gB;K;IAEjG,0B;MAAA,8B;MACI,iBAGmC,SAAK,CAAL,C;MAEnC,iBAGmC,SAAK,EAAL,
C;MAEnC,kBAGmC,C;MAEnC,iBAGkC,E;K;;;IAnBtC,sC;MAAA,qC;QAAA,oB;;MAAA,8B;K;oGAsBA,yB;M
DqRA,6B;MCrQA,8C;MAhBA,wB;QAM0D,OAiBQ,YAAY,IAAK,KAAjB,EAA6B,CDuQ5D,cCxRsC,KDwR5B,
KAAL,GAAiB,GAAtB,CCvQ4D,MAA7B,C;O;KAvBlE,C;oGAQA,yB;MC8QA,6B;MDtQA,8C;MARA,wB;QAM
2D,OASO,YAAY,IAAK,KAAjB,EAA6B,CCwQ5D,cDjRuC,KCiR7B,KAAL,GAAiB,KAAtB,CDxQ4D,MAA7B,
C;O;KAflE,C;gGAQA,yB;MAAA,8C;MAAA,wB;QAOkE,mBAAY,IAAK,KAAjB,EAAuB,KAAM,KAA7B,C;O;
KAPlE,C;oGASA,yB;MA0RA,kBAS6D,sB;MAT7D,+B;MiB3RA,gD;MjBCA,wB;QAM0D,OiBAS,aAAkB,CjB6
RhD,eAAW,oBAAL,SAAK,CAAL,iBAAN,CiB7RgD,MAAlB,EjBAgB,KiBAc,KAA9B,C;O;KjBNnE,C;0FAQA,y
B;MDoPA,6B;MCpPA,wB;QAEsD,OAMD,cAAK,IAAK,KAAK,GAAW,CDqP5C,cC3P6B,KD2PnB,KAAL,GAA
iB,GAAtB,CCrP4C,MAAX,IAAf,C;O;KARrD,C;0FAGA,yB;MCkPA,6B;MDlPA,wB;QAEuD,OAGF,cAAK,IAA
K,KAAK,GAAW,CCsP5C,cDzP8B,KCyPpB,KAAL,GAAiB,KAAtB,CDtP4C,MAAX,IAAf,C;O;KALrD,C;0FAG
A,yB;MAAA,6B;MAAA,wB;QAEqD,qBAAK,IAAK,KAAK,GAAK,KAAM,KAAX,IAAf,C;O;KAFrD,C;0FAGA,
yB;MAyQA,kBAS6D,sB;MAT7D,+B;MAzQA,wB;QAEuD,OiBAA,eAAW,CjBgR7B,eAAW,oBAAL,SAAK,CA
AL,iBAAN,CiBhR6B,MAAK,KjBAI,KiBAO,KAAX,CAAhB,C;O;KjBFvD,C;4FAIA,yB;MDuOA,6B;MCvOA,w

B;QAEuD,OAMD,cAAK,IAAK,KAAK,GAAY,CDwO9C,cC9O+B,KD8OrB,KAAL,GAAiB,GAAtB,CCxO8C,M
AAZ,IAAf,C;O;KARtD,C;4FAGA,yB;MCqOA,6B;MDrOA,wB;QAEwD,OAGF,cAAK,IAAK,KAAK,GAAY,CCy
O9C,cD5OgC,KC4OtB,KAAL,GAAiB,KAAtB,CDzO8C,MAAZ,IAAf,C;O;KALtD,C;4FAGA,yB;MAAA,6B;MA
AA,wB;QAEsD,qBAAK,IAAK,KAAK,GAAM,KAAM,KAAZ,IAAf,C;O;KAFtD,C;4FAGA,yB;MA4PA,kBAS6D
,sB;MAT7D,+B;MA5PA,wB;QAEwD,OiBAA,eAAW,CjBmQ9B,eAAW,oBAAL,SAAK,CAAL,iBAAN,CiBnQ8B
,MAAK,UjBAK,KiBAO,KAAZ,CAAhB,C;O;KjBFxD,C;4FAIA,yB;MD0NA,6B;MC1NA,wB;QAEuD,OAMD,cA
Ae,YAAV,IAAK,KAAK,EAAY,CD2N9C,cCjO+B,KDiOrB,KAAL,GAAiB,GAAtB,CC3N8C,MAAZ,CAAf,C;O;
KARtD,C;4FAGA,yB;MCwNA,6B;MDxNA,wB;QAEwD,OAGF,cAAe,YAAV,IAAK,KAAK,EAAY,CC4N9C,cD
/NgC,KC+NtB,KAAL,GAAiB,KAAtB,CD5N8C,MAAZ,CAAf,C;O;KALtD,C;4FAGA,yB;MAAA,6B;MAAA,wB;
QAEsD,qBAAe,YAAV,IAAK,KAAK,EAAM,KAAM,KAAZ,CAAf,C;O;KAFtD,C;4FAGA,yB;MA+OA,kBAS6D
,sB;MAT7D,+B;MA/OA,wB;QAEwD,OiBAA,eAAW,CjBsP9B,eAAW,oBAAL,SAAK,CAAL,iBAAN,CiBtP8B,
MAAK,UjBAK,KiBAO,KAAZ,CAAhB,C;O;KjBFxD,C;wFAIA,yB;MD6MA,6B;MCvMA,4C;MANA,wB;QAEq
D,OAMD,WAAW,IAAX,ED8MjB,cCpN2B,KDoNjB,KAAL,GAAiB,GAAtB,CC9MiB,C;O;KARpD,C;wFAGA,y
B;MC2MA,6B;MDxMA,4C;MAHA,wB;QAEsD,OAGF,WAAW,IAAX,EC+MjB,cDlN4B,KCkNlB,KAAL,GAAi
B,KAAtB,CD/MiB,C;O;KALpD,C;wFAGA,yB;MAAA,4C;MAAA,wB;QAEoD,kBAAW,IAAX,EAAiB,KAAjB,C
;O;KAFpD,C;wFAGA,yB;MAkOA,kBAS6D,sB;MAT7D,+B;MiBlOA,8C;MjBAA,wB;QAEsD,OiBAA,YjByOjB,
eAAW,oBAAL,SAAK,CAAL,iBAAN,CiBzOiB,EjBAmB,KiBAnB,C;O;KjBFtD,C;wFAIA,yB;MDgMA,6B;MClL
A,kD;MAdA,wB;QAMqD,OAcD,cAAc,IAAd,EDqLjB,cCnM2B,KDmMjB,KAAL,GAAiB,GAAtB,CCrLiB,C;O;K
ApBpD,C;wFAOA,yB;MC0LA,6B;MDnLA,kD;MAPA,wB;QAMsD,OAOF,cAAc,IAAd,ECsLjB,cD7L4B,KC6Ll
B,KAAL,GAAiB,KAAtB,CDtLiB,C;O;KAbpD,C;wFAOA,yB;MAAA,kD;MAAA,wB;QAMoD,qBAAc,IAAd,EA
AoB,KAApB,C;O;KANpD,C;wFAOA,yB;MAyMA,kBAS6D,sB;MAT7D,+B;MiBzMA,oD;MjBAA,wB;QAMsD,
OiBAA,ejB4MjB,eAAW,oBAAL,SAAK,CAAL,iBAAN,CiB5MiB,EjBAmB,KiBAnB,C;O;KjBNtD,C;kGAQA,yB;
MDmKA,6B;MCvMA,4C;MAoCA,wB;QAMiD,OAxCG,WAAW,IAAX,ED8MjB,cCtK4B,KDsKlB,KAAL,GAAi
B,GAAtB,CC9MiB,C;O;KAkCpD,C;kGAOA,yB;MC6JA,6B;MDxMA,4C;MA2CA,wB;QAMkD,OA/CE,WAAW,
IAAX,EC+MjB,cDhK6B,KCgKnB,KAAL,GAAiB,KAAtB,CD/MiB,C;O;KAyCpD,C;kGAOA,yB;MAlDA,4C;MA
kDA,wB;QAMgD,OAtDI,WAAW,IAAX,EAsDA,KAtDA,C;O;KAgDpD,C;kGAOA,yB;MA4KA,kBAS6D,sB;MA
T7D,+B;MiBlOA,8C;MjBsDA,wB;QAMkD,OiB1DI,YjByOjB,eAAW,oBAAL,SAAK,CAAL,iBAAN,CiBzOiB,Ej
B0DoB,KiB1DpB,C;O;KjBoDtD,C;wFAQA,yB;MDsIA,6B;MClLA,kD;MDiPJ,0B;MAAA,+B;MCrMI,wB;QAQ6
C,ODwMR,eAAW,OCtPI,cAAc,IAAd,EDqLjB,cCvImB,KDuIT,KAAL,GAAiB,GAAtB,CCrLiB,CA4Lf,KD0DW,
CAAX,C;O;KChNrC,C;wFASA,yB;MC8HA,6B;MDnLA,kD;MCkPJ,4B;MAAA,iC;MD7LI,wB;QAQ+C,OCgMR,
gBAAY,QDvPC,cAAc,IAAd,ECsLjB,cD/HqB,KC+HX,KAAL,GAAiB,KAAtB,CDtLiB,CAsMb,KCiDY,CAAZ,C;
O;KDxMvC,C;wFASA,yB;MA9DA,kD;MA8DA,wB;QAQ2C,OAhES,cAAc,IAAd,EAgEL,KAhEK,C;O;KAwDp
D,C;wFASA,yB;MAyIA,kBAS6D,sB;MAT7D,+B;MiBzMA,oD;MjBgEA,wB;QAQ6C,OiBlES,ejB4MjB,eAAW,o
BAAL,SAAK,CAAL,iBAAN,CiB5MiB,EjBkEU,KiBlEV,C;O;KjB0DtD,C;wEAUA,yB;MAAA,6B;MAAA,mB;Q
AMyC,qBAAK,SAAK,QAAV,C;O;KANzC,C;wEAQA,yB;MAAA,6B;MAAA,mB;QAMyC,qBAAK,SAAK,QAA
V,C;O;KANzC,C;gGAQA,yB;MAAA,8C;MAAA,wB;QAE6D,0BAAU,IAAV,EAAgB,KAAhB,C;O;KAF7D,C;sG
AIA,yB;MAAA,yC;MAAA,wB;QAQgE,mBAAW,KAAX,C;O;KARhE,C;wFAUA,yB;MAAA,6B;MAAA,2B;QA
OmD,qBAAK,aAAS,QAAd,C;O;KAPnD,C;wFASA,yB;MAAA,6B;MAAA,2B;QAOmD,qBAAK,cAAU,QAAf,C;
O;KAPnD,C;wFASA,yB;MAAA,6B;MAAA,wB;QAEiD,qBAAK,IAAK,KAAL,GAAc,KAAM,KAAzB,C;O;KAFj
D,C;sFAGA,yB;MAAA,6B;MAAA,wB;QAEgD,qBAAK,IAAK,KAAL,GAAa,KAAM,KAAxB,C;O;KAFhD,C;wF
AGA,yB;MAAA,6B;MAAA,wB;QAEiD,qBAAK,IAAK,KAAL,GAAc,KAAM,KAAzB,C;O;KAFjD,C;wEAGA,y
B;MAAA,6B;MAAA,mB;QAEgC,qBAAU,CAAL,SAAL,C;O;KAFhC,C;8EAIA,yB;MAAA,0B;MAAA,mB;QAU
mC,OAAK,OAAL,SAAK,C;O;KAVxC,C;gFAWA,yB;MAAA,4B;MAAA,mB;QAUqC,OAAK,QAAL,SAAK,C;O
;KAV1C,C;4EAWA,Y;MASiC,gB;K;8EACjC,yB;MAAA,kBASqD,sB;MATrD,mB;QASmC,OAAK,oBAAL,SAA
K,CAAL,iB;O;KATnC,C;gFAWA,yB;MDwDJ,0B;MAAA,+B;MCxDI,mB;QASqC,OD0DA,eAAW,OC1DX,SD0
DW,CAAX,C;O;KCnErC,C;kFAUA,yB;MC+CJ,4B;MAAA,iC;MD/CI,mB;QASuC,OCiDA,gBAAY,QDjDZ,SCiD
Y,CAAZ,C;O;KD1DvC,C;8EAUA,Y;MAEmC,W;K;gFACnC,yB;MAAA,kBAS6D,sB;MAT7D,+B;MAAA,mB;Q
ASqC,sBAAW,oBAAL,SAAK,CAAL,iBAAN,C;O;KATrC,C;gFAWA,yB;MASA,gD;MATA,mB;QAQqC,OAOE,

aAAa,SAAb,C;O;KAfvC,C;kFASA,yB;MAAA,gD;MAAA,mB;QAMuC,oBAAa,SAAb,C;O;KANvC,C;8BAQA,Y
;MAAyC,OArDD,oBAAL,SAAK,CAAL,iBAqDe,W;K;;;;;8BA1WtD,Y;MAAA,c;MAGqG,qD;MAHrG,a;K;4BAA
A,iB;MAAA,2IAGqG,oCAHrG,G;K;sEA8WA,yB;MAAA,6B;MAAA,4B;QAWwC,qBAAU,SAAV,C;O;KAXxC,
C;wEAYA,yB;MAAA,6B;MAAA,4B;QAWyC,qBAAU,SAAV,C;O;KAXzC,C;wEAYA,yB;MAAA,6B;MAAA,4
B;QAUuC,qBAAK,SAAL,C;O;KAVvC,C;wEAWA,yB;MAAA,6B;MAAA,4B;QAWwC,qBAAK,SAAK,QAAV,C
;O;KAXxC,C;uEAaA,yB;MAAA,gD;MAAA,4B;QASyC,oBAAkB,SAAlB,C;O;KATzC,C;wEAUA,yB;MAAA,gD
;MAAA,4B;QAS0C,oBAAa,SAAb,C;O;KAT1C,C;IiCraA,4B;MACqB,sB;K;sCAKjB,iB;MAM4C,OjCiYT,SiCjYS
,aAAQ,KAAR,CjCiYT,C;K;sCiC/XnC,wB;MAOI,aAAQ,KAAR,IAAiB,KjCmRY,K;K;iFiC/QH,Y;MAAQ,OAAA
,YAAQ,O;K;mCAE9C,Y;MAC6E,8BAAS,YAAT,C;K;IAEvD,mC;MAAC,oB;MACnB,eAAoB,C;K;2CACpB,Y;
MAAyB,sBAAQ,YAAM,O;K;wCACvC,Y;MAAoD,Q;MAA9B,IAAI,eAAQ,YAAM,OAAlB,C;QAAA,OjC4WS,S
iC5We,aAAM,mBAAN,EAAM,2BAAN,OjC4Wf,C;;QiC5W4C,MAAM,2BAAuB,YAAM,WAA7B,C;K;;yCAGrF,
mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,OAAJ,C;QAAgC,OAAO,K;MAEvC,OAAe,WAAR,YAAQ,EAAS,OjC
8PK,KiC9Pd,C;K;8CAGnB,oB;MACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;;QfioDvB,U;QADhB,IAAI,wCAAsB,
mBAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,6B;QAAhB,OAAgB,gBAAhB,C;UAAgB,2B;UejoD6B,2BfioDR,
OejoDQ,O;UAAA,W;YAAsB,oBAAR,YAAQ,EfioD9B,OlBv4CJ,KiC1PkC,C;;UfioD7C,IAAI,OAAJ,C;YAAyB,a
AAO,K;YAAP,e;;;QAC/C,aAAO,I;;;MeloDH,iB;K;kCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,KAAqB,C;K;;
IA9CvD,qC;MAAA,mD;MACgC,sBAAK,eAAS,IAAT,CAAL,C;MADhC,Y;K;;;;mCAPJ,Y;MAAA,OAKqB,oDA
LrB,M;K;mCAAA,Y;MAAA,c;MAKqB,wD;MALrB,a;K;iCAAA,iB;MAAA,2IAKqB,0CALrB,G;K;8EAwDA,yB;
MAAA,uC;MAWoC,wC;QAAA,wB;UAAW,OAAA,aAAK,KAAL,CjCyOV,K;S;O;MiCpPrC,6B;QAWI,OAAO,m
BAAU,gCAAS,IAAT,GAAe,sBAAf,CAAV,C;O;KAXX,C;gFAcA,oB;MAGkE,e;K;IkMjE5C,wC;MA8BlB,iC;MA
9BsD,2BAAgB,KAAhB,EAAuB,YAAvB,EAAqC,CAArC,C;K;kFAC7B,Y;MAAQ,iB;K;yFACD,Y;MAAQ,gB;K;
yFAKR,Y;MACxB,Q;MAAJ,IAAI,yCAAQ,4BAAK,UAAb,QAAJ,C;QvNmHyC,MAAM,6BuNnHb,6EvNmH2C,
WAA9B,C;;MuNlH/C,OnOoDiD,SmOpD1C,SnOoDoD,KAAK,GAAW,CmOpD7D,WnOoD6D,MAAX,IAAf,C;K;
2CmOjDrD,iB;MAA8C,WnO+BoB,YmO/BpB,UnO+BqC,KAAjB,EmO/BX,KnO+BwC,KAA7B,CmO/BpB,K;M
AAA,S;QAAkB,OnO+BE,YmO/BF,KnO+BmB,KAAjB,EmO/BO,SnO+BsB,KAA7B,CmO/BF,K;;MAAlB,W;K;k
CAE9C,Y;MAKkC,OnOwBgC,YmOxBhC,UnOwBiD,KAAjB,EmOxBxB,SnOwBqD,KAA7B,CmOxBhC,I;K;iCA
ElC,iB;MAEY,UAAwB,M;MADhC,2CAAuB,kBAAa,KAAM,UAAnB,KACf,2CAAS,KAAM,MAAf,cAAwB,6C
AAQ,KAAM,KAAd,QAAxB,CADe,CAAvB,C;K;mCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,MAAK,U
nO0QA,KmO1QL,QAAqB,SnO0QhB,KmO1QL,I;K;mCAE5B,Y;MAAkC,OAAE,UAAF,qBAAU,S;K;IAE5C,+B;
MAAA,mC;MACI,aAC8B,cAAU,4BAAK,UAAf,EAA0B,4BAAK,UAA/B,C;K;;;IAFlC,2C;MAAA,0C;QAAA,yB
;;MAAA,mC;K;;IAYJ,oD;MA4CI,uC;MAtCI,IAAI,SAAQ,CAAZ,C;QAAuB,MAAa,gCAAyB,wBAAzB,C;MACp
C,IAAI,SAAQ,WAAZ,C;QAA2B,MAAa,gCAAyB,wEAAzB,C;MAG5C,aAGyB,K;MAEzB,YAGwB,4BAA0B,K
AA1B,EAAiC,YAAjC,EAA+C,IAA/C,C;MAExB,YAGuB,I;K;yCAEvB,Y;MAAgD,mCAAwB,UAAxB,EAA+B,S
AA/B,EAAqC,SAArC,C;K;wCAEhD,Y;MAMqC,OAAI,YAAO,CAAX,GnOhC6B,YmOgCf,UnOhCgC,KAAjB,E
mOgCP,SnOhCoC,KAA7B,CmOgCf,IAAd,GnOhC6B,YmOgCG,UnOhCc,KAAjB,EmOgCW,SnOhCkB,KAA7B,
CmOgCG,I;K;uCAErE,iB;MAEY,UAAwB,M;MADhC,iDAA6B,kBAAa,KAAM,UAAnB,KACrB,2CAAS,KAAM
,MAAf,cAAwB,6CAAQ,KAAM,KAAd,QAAxB,KAA8C,cAAQ,KAAM,KADvC,CAA7B,C;K;yCAGJ,Y;MACI,O
AAI,cAAJ,GAAe,EAAf,GAAwB,OAAM,MAAK,UnOkNN,KmOlNC,QAAqB,SnOkNtB,KmOlNC,IAAN,SAAgD
,SAAhD,I;K;yCAE5B,Y;MAAkC,OAAI,YAAO,CAAX,GAAgB,UAAF,qBAAU,SAAV,cAAqB,SAAnC,GAAgD,
UAAF,2BAAgB,SAAhB,eAA4B,CAAC,SAAD,IAA5B,C;K;IAEhF,qC;MAAA,yC;K;kEACI,sC;MAQ2F,2BAAg
B,UAAhB,EAA4B,QAA5B,EAAsC,IAAtC,C;K;;;IAT/F,iD;MAAA,gD;QAAA,+B;;MAAA,yC;K;;IAmBiC,oD;M
ACjC,sBAA2B,I;MAC3B,iBAAmC,OAAO,CAA1C,GnOhEkE,YmOgErB,KnOhEsC,KAAjB,EmOgEZ,InOhEyC,
KAA7B,CmOgErB,KAA7C,GnOhEkE,YmOgEF,KnOhEmB,KAAjB,EmOgEO,InOhEsB,KAA7B,CmOgEF,K;M
AChE,cnO6RmC,SmO7RhB,InO6RgB,C;MmO5RnC,cAAuB,cAAJ,GAAa,KAAb,GAAwB,mB;K;gDAE3C,Y;MA
AkC,qB;K;6CAElC,Y;MACI,YAAY,W;MACZ,IAAI,6BAAS,mBAAT,QAAJ,C;QACI,IAAI,CAAC,cAAL,C;UA
Ac,MAAa,6B;QAC3B,iBAAU,K;;QAEV,cnO1D6C,SmO0D7C,WnO1DuD,KAAK,GmO0DpD,WnO1D+D,KAA
X,IAAf,C;;MmO4DjD,OAAO,K;K;;IlN7HU,qB;MAErB,6B;MAFkG,gB;K;IAElG,2B;MAAA,+B;MACI,iBAGoC,
a;MAEpC,iBAGoC,c;MAEpC,kBAGmC,C;MAEnC,iBAGkC,E;K;;;IAnBtC,uC;MAAA,sC;QAAA,qB;;MAAA,+B

;K;sGAsBA,yB;MlB+RA,WAS6D,wB;MAT7D,+B;MkBvQA,gD;MAxBA,wB;QAM0D,OAyBS,aAAa,IAAK,KA
AlB,EAA8B,ClByQ5D,eAAW,oBkBlSyB,KlBkS9B,KAAK,CAAL,UAAN,CkBzQ4D,MAA9B,C;O;KA/BnE,C;sG
AQA,yB;MhBwRA,aAS6D,0B;MAT7D,+B;MgBxQA,gD;MAhBA,wB;QAM2D,OAiBQ,aAAa,IAAK,KAAlB,EA
A8B,ChB0Q5D,eAAW,oBgB3R0B,KhB2R/B,KAAK,CAAL,YAAN,CgB1Q4D,MAA9B,C;O;KAvBnE,C;sGAQA
,yB;MjBmSA,kBAS6D,sB;MAT7D,+B;MiB3RA,gD;MARA,wB;QAMyD,OASU,aAAa,IAAK,KAAlB,EAA8B,Cj
B6R5D,eAAW,oBiBtSwB,KjBsS7B,KAAK,CAAL,iBAAN,CiB7R4D,MAA9B,C;O;KAfnE,C;kGAQA,yB;MAAA
,gD;MAAA,wB;QAOmE,oBAAa,IAAK,KAAlB,EAAwB,KAAM,KAA9B,C;O;KAPnE,C;4FASA,yB;MlB8PA,W
AS6D,wB;MAT7D,+B;MkB9PA,wB;QAEuD,OASA,eAAM,IAAK,KAAK,KAAW,ClB4P7C,eAAW,oBkBrQiB,K
lBqQtB,KAAK,CAAL,UAAN,CkB5P6C,MAAX,CAAhB,C;O;KAXvD,C;4FAGA,yB;MhB4PA,aAS6D,0B;MAT7
D,+B;MgB5PA,wB;QAEwD,OAMD,eAAM,IAAK,KAAK,KAAW,ChB6P7C,eAAW,oBgBnQkB,KhBmQvB,KA
AK,CAAL,YAAN,CgB7P6C,MAAX,CAAhB,C;O;KARvD,C;4FAGA,yB;MjB4QA,kBAS6D,sB;MAT7D,+B;MiB
5QA,wB;QAEsD,OAGC,eAAM,IAAK,KAAK,KAAW,CjBgR7C,eAAW,oBiBnRgB,KjBmRrB,KAAK,CAAL,iBA
AN,CiBhR6C,MAAX,CAAhB,C;O;KALvD,C;4FAGA,yB;MAAA,+B;MAAA,wB;QAEuD,sBAAM,IAAK,KAAK
,KAAK,KAAM,KAAX,CAAhB,C;O;KAFvD,C;8FAIA,yB;MlBiPA,WAS6D,wB;MAT7D,+B;MkBjPA,wB;QAEw
D,OASA,eAAM,IAAK,KAAK,UAAY,ClB+O/C,eAAW,oBkBxPmB,KlBwPxB,KAAK,CAAL,UAAN,CkB/O+C,
MAAZ,CAAhB,C;O;KAXxD,C;8FAGA,yB;MhB+OA,aAS6D,0B;MAT7D,+B;MgB/OA,wB;QAEyD,OAMD,eAA
M,IAAK,KAAK,UAAY,ChBgP/C,eAAW,oBgBtPoB,KhBsPzB,KAAK,CAAL,YAAN,CgBhP+C,MAAZ,CAAhB,
C;O;KARxD,C;8FAGA,yB;MjB+PA,kBAS6D,sB;MAT7D,+B;MiB/PA,wB;QAEuD,OAGC,eAAM,IAAK,KAAK,
UAAY,CjBmQ/C,eAAW,oBiBtQkB,KjBsQvB,KAAK,CAAL,iBAAN,CiBnQ+C,MAAZ,CAAhB,C;O;KALxD,C;8
FAGA,yB;MAAA,+B;MAAA,wB;QAEwD,sBAAM,IAAK,KAAK,UAAM,KAAM,KAAZ,CAAhB,C;O;KAFxD,C
;8FAIA,yB;MlBoOA,WAS6D,wB;MAT7D,+B;MkBpOA,wB;QAEwD,OASA,eAAM,IAAK,KAAK,UAAY,ClBk
O/C,eAAW,oBkB3OmB,KlB2OxB,KAAK,CAAL,UAAN,CkBlO+C,MAAZ,CAAhB,C;O;KAXxD,C;8FAGA,yB;
MhBkOA,aAS6D,0B;MAT7D,+B;MgBlOA,wB;QAEyD,OAMD,eAAM,IAAK,KAAK,UAAY,ChBmO/C,eAAW,o
BgBzOoB,KhByOzB,KAAK,CAAL,YAAN,CgBnO+C,MAAZ,CAAhB,C;O;KARxD,C;8FAGA,yB;MjBkPA,kBA
S6D,sB;MAT7D,+B;MiBlPA,wB;QAEuD,OAGC,eAAM,IAAK,KAAK,UAAY,CjBsP/C,eAAW,oBiBzPkB,KjByP
vB,KAAK,CAAL,iBAAN,CiBtP+C,MAAZ,CAAhB,C;O;KALxD,C;8FAGA,yB;MAAA,+B;MAAA,wB;QAEwD,s
BAAM,IAAK,KAAK,UAAM,KAAM,KAAZ,CAAhB,C;O;KAFxD,C;0FAIA,yB;MlBuNA,WAS6D,wB;MAT7D,
+B;MkB9MA,8C;MATA,wB;QAEsD,OASA,YAAY,IAAZ,ElBqNjB,eAAW,oBkB9Ne,KlB8NpB,KAAK,CAAL,
UAAN,CkBrNiB,C;O;KAXtD,C;0FAGA,yB;MhBqNA,aAS6D,0B;MAT7D,+B;MgB/MA,8C;MANA,wB;QAEuD,
OAMD,YAAY,IAAZ,EhBsNjB,eAAW,oBgB5NgB,KhB4NrB,KAAK,CAAL,YAAN,CgBtNiB,C;O;KARtD,C;0F
AGA,yB;MjBqOA,kBAS6D,sB;MAT7D,+B;MiBlOA,8C;MAHA,wB;QAEqD,OAGC,YAAY,IAAZ,EjByOjB,eA
AW,oBiB5Oc,KjB4OnB,KAAK,CAAL,iBAAN,CiBzOiB,C;O;KALtD,C;0FAGA,yB;MAAA,8C;MAAA,wB;QAE
sD,mBAAY,IAAZ,EAAkB,KAAlB,C;O;KAFtD,C;0FAIA,yB;MlB0MA,WAS6D,wB;MAT7D,+B;MkBrLA,oD;M
ArBA,wB;QAMsD,OAqBA,eAAe,IAAf,ElBwLjB,eAAW,oBkB7Me,KlB6MpB,KAAK,CAAL,UAAN,CkBxLiB,C
;O;KA3BtD,C;0FAOA,yB;MhBoMA,aAS6D,0B;MAT7D,+B;MgBtLA,oD;MAdA,wB;QAMuD,OAcD,eAAe,IAAf
,EhByLjB,eAAW,oBgBvMgB,KhBuMrB,KAAK,CAAL,YAAN,CgBzLiB,C;O;KApBtD,C;0FAOA,yB;MjBgNA,k
BAS6D,sB;MAT7D,+B;MiBzMA,oD;MAPA,wB;QAMqD,OAOC,eAAe,IAAf,EjB4MjB,eAAW,oBiBnNc,KjBmN
nB,KAAK,CAAL,iBAAN,CiB5MiB,C;O;KAbtD,C;0FAOA,yB;MAAA,oD;MAAA,wB;QAMsD,sBAAe,IAAf,EA
AqB,KAArB,C;O;KANtD,C;oGAQA,yB;MlB6KA,WAS6D,wB;MAT7D,+B;MkB9MA,8C;MAiCA,wB;QAMkD,
OArCI,YAAY,IAAZ,ElBqNjB,eAAW,oBkBhLgB,KlBgLrB,KAAK,CAAL,UAAN,CkBrNiB,C;O;KA+BtD,C;oG
AOA,yB;MhBuKA,aAS6D,0B;MAT7D,+B;MgB/MA,8C;MAwCA,wB;QAMmD,OA5CG,YAAY,IAAZ,EhBsNjB,
eAAW,oBgB1KiB,KhB0KtB,KAAK,CAAL,YAAN,CgBtNiB,C;O;KAsCtD,C;oGAOA,yB;MjBmLA,kBAS6D,sB;
MAT7D,+B;MiBlOA,8C;MA+CA,wB;QAMiD,OAnDK,YAAY,IAAZ,EjByOjB,eAAW,oBiBtLe,KjBsLpB,KAAK
,CAAL,iBAAN,CiBzOiB,C;O;KA6CtD,C;oGAOA,yB;MAtDA,8C;MAsDA,wB;QAMkD,OA1DI,YAAY,IAAZ,E
A0DA,KA1DA,C;O;KAoDtD,C;0FAQA,yB;MlBgJA,WAS6D,wB;MAT7D,+B;MkBrLA,oD;MlBsPJ,0B;MAAA,+
B;MkBjNI,wB;QAQ6C,OlBoNP,eAAW,OkB3PK,eAAe,IAAf,ElBwLjB,eAAW,oBkBjJM,KlBiJX,KAAK,CAAL,
UAAN,CkBxLiB,CAsLjB,KlBqEY,SAAX,C;O;KkB5NtC,C;0FASA,yB;MhBwIA,aAS6D,0B;MAT7D,+B;MgBtL
A,oD;MhBuPJ,4B;MAAA,iC;MgBzMI,wB;QAQ+C,OhB4MP,gBAAY,QgB5PE,eAAe,IAAf,EhByLjB,eAAW,oBg

BzIQ,KhByIb,KAAK,CAAL,YAAN,CgBzLiB,CAgMf,KhB4Da,SAAZ,C;O;KgBpNxC,C;0FASA,yB;MjBkJA,kB
AS6D,sB;MAT7D,+B;MiBzMA,oD;MjB4QJ,6B;MiBrNI,wB;QAQ2C,OjBwNP,ciBjRkB,eAAe,IAAf,EjB4MjB,eA
AW,oBiBnJI,KjBmJT,KAAK,CAAL,iBAAN,CiB5MiB,CA0MnB,KjBuEW,QAAV,C;O;KiBhOpC,C;0FASA,yB;
MAhEA,oD;MAgEA,wB;QAQ6C,OAlES,eAAe,IAAf,EAkEL,KAlEK,C;O;KA0DtD,C;0EAUA,yB;MAAA,+B;M
AAA,mB;QAM0C,sBAAM,SAAK,MAAX,C;O;KAN1C,C;0EAQA,yB;MAAA,+B;MAAA,mB;QAM0C,sBAAM,
SAAK,MAAX,C;O;KAN1C,C;kGAQA,yB;MAAA,gD;MAAA,wB;QAE+D,2BAAW,IAAX,EAAiB,KAAjB,C;O;
KAF/D,C;wGAIA,yB;MAAA,yC;MAAA,wB;QAQkE,mBAAW,KAAX,C;O;KARlE,C;0FAUA,yB;MAAA,+B;M
AAA,2B;QAOoD,sBAAM,oBAAS,QAAT,CAAN,C;O;KAPpD,C;0FASA,yB;MAAA,+B;MAAA,2B;QAOoD,sBA
AM,6BAAU,QAAV,CAAN,C;O;KAPpD,C;0FASA,yB;MAAA,+B;MAAA,wB;QAEmD,sBAAM,IAAK,KAAL,K
AAc,KAAM,KAApB,CAAN,C;O;KAFnD,C;wFAGA,yB;MAAA,+B;MAAA,wB;QAEkD,sBAAM,IAAK,KAAL,I
AAa,KAAM,KAAnB,CAAN,C;O;KAFlD,C;0FAGA,yB;MAAA,+B;MAAA,wB;QAEmD,sBAAM,IAAK,KAAL,
KAAc,KAAM,KAApB,CAAN,C;O;KAFnD,C;0EAGA,yB;MAAA,+B;MAAA,mB;QAEiC,sBAAM,SAAK,MAA
X,C;O;KAFjC,C;gFAIA,yB;MAAA,0B;MAAA,mB;QAUmC,OAAK,OAAL,SAAK,S;O;KAVxC,C;kFAWA,yB;M
AAA,4B;MAAA,mB;QAUqC,OAAK,QAAL,SAAK,S;O;KAV1C,C;8EAWA,Y;MAUiC,OAAA,SAAK,Q;K;gFA
CtC,Y;MASmC,gB;K;kFAEnC,yB;MlBmEJ,0B;MAAA,+B;MkBnEI,mB;QASqC,OlBqEC,eAAW,OkBrEZ,SlBqE
Y,SAAX,C;O;KkB9EtC,C;oFAUA,yB;MhB0DJ,4B;MAAA,iC;MgB1DI,mB;QASuC,OhB4DC,gBAAY,QgB5Db,
ShB4Da,SAAZ,C;O;KgBrExC,C;gFAUA,yB;MjBqEJ,6B;MiBrEI,mB;QASmC,OjBuEC,ciBvED,SjBuEW,QAAV,
C;O;KiBhFpC,C;kFAUA,Y;MAEqC,W;K;kFAErC,yB;MASA,kD;MATA,mB;QAQqC,OASE,cAAc,SAAd,C;O;K
AjBvC,C;oFASA,yB;MAAA,kD;MAAA,mB;QAQuC,qBAAc,SAAd,C;O;KARvC,C;+BAUA,Y;MAAyC,qBAAc,
SAAd,C;K;;;;;+BA7W7C,Y;MAAA,c;MAGsG,qD;MAHtG,a;K;6BAAA,iB;MAAA,2IAGsG,oCAHtG,G;K;wEAi
XA,yB;MAAA,+B;MAAA,4B;QAW0C,sBAAW,oBAAL,SAAK,CAAX,C;O;KAX1C,C;0EAYA,yB;MAAA,+B;
MAAA,4B;QAW2C,sBAAW,oBAAL,SAAK,CAAX,C;O;KAX3C,C;0EAYA,yB;MAAA,+B;MAAA,4B;QAWyC,
sBAAW,oBAAL,SAAK,CAAX,C;O;KAXzC,C;0EAYA,yB;MAAA,+B;MAAA,4B;QAU0C,sBAAM,SAAN,C;O;
KAV1C,C;yEAYA,yB;MAAA,kD;MAAA,4B;QAS2C,qBAAmB,SAAnB,C;O;KAT3C,C;0EAUA,yB;MAAA,kD;
MAAA,4B;QAS4C,qBAAc,SAAd,C;O;KAT5C,C;IiBxaA,6B;MACqB,sB;K;uCAKjB,iB;MAM6C,OjBgZP,UiBhZ
O,aAAQ,KAAR,CjBgZP,C;K;uCiB9YtC,wB;MAOI,aAAQ,KAAR,IAAiB,KjB8Rc,K;K;kFiB1RL,Y;MAAQ,OAA
A,YAAQ,O;K;oCAE9C,Y;MAC8E,+BAAS,YAAT,C;K;IAExD,oC;MAAC,oB;MACnB,eAAoB,C;K;4CACpB,Y;
MAAyB,sBAAQ,YAAM,O;K;yCACvC,Y;MAAoD,Q;MAA9B,IAAI,eAAQ,YAAM,OAAlB,C;QAAA,OjB2XY,U
iB3XY,aAAM,mBAAN,EAAM,2BAAN,OjB2XZ,C;;QiB3X0C,MAAM,2BAAuB,YAAM,WAA7B,C;K;;0CAGtF,
mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,QAAJ,C;QAAiC,OAAO,K;MAExC,OAAe,WAAR,YAAQ,EAAS,OjBy
QO,KiBzQhB,C;K;+CAGnB,oB;MACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;;QhBioDvB,U;QADhB,IAAI,wCAAs
B,mBAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,6B;QAAhB,OAAgB,gBAAhB,C;UAAgB,2B;UgBjoD6B,2BhBi
oDR,OgBjoDQ,Q;UAAA,W;YAAuB,oBAAR,YAAQ,EhBioD/B,OD53CF,KiBrQiC,C;;UhBioD9C,IAAI,OAAJ,C;
YAAyB,aAAO,K;YAAP,e;;;QAC/C,aAAO,I;;;MgBloDH,iB;K;mCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,OAAb,K
AAqB,C;K;;IA9CvD,sC;MAAA,oD;MACgC,uBAAK,iBAAU,IAAV,CAAL,C;MADhC,Y;K;;;;oCAPJ,Y;MAAA,
OAKqB,qDALrB,M;K;oCAAA,Y;MAAA,c;MAKqB,wD;MALrB,a;K;kCAAA,iB;MAAA,2IAKqB,0CALrB,G;K;
gFAwDA,yB;MAAA,yC;MAWsC,yC;QAAA,wB;UAAW,OAAA,aAAK,KAAL,CjBoPV,K;S;O;MiB/PvC,6B;QA
WI,OAAO,oBAAW,kBAAU,IAAV,EAAgB,uBAAhB,CAAX,C;O;KAXX,C;kFAcA,oB;MAGqE,e;K;IkMjE9C,2C
;MA8BnB,kC;MA9ByD,4BAAiB,KAAjB,EAAwB,YAAxB,K;K;qFAC/B,Y;MAAQ,iB;K;4FACD,Y;MAAQ,gB;K
;4FAKR,Y;MACzB,Q;MAAJ,IAAI,yCAAQ,6BAAM,UAAd,QAAJ,C;QxNmHyC,MAAM,6BwNnHZ,6ExNmH0C
,WAA9B,C;;MwNlH/C,OnNuDmD,UmNvD5C,SnNuDuD,KAAK,KAAW,CjBgR7C,UAAW,oBAAL,CoOvUzB,
WpOuUyB,MAAK,CAAL,iBAAN,CiBhR6C,MAAX,CAAhB,C;K;8CmNpDvD,iB;MAA+C,WnNuCoB,amNvCp
B,UnNuCsC,KAAlB,EmNvCX,KnNuCyC,KAA9B,CmNvCpB,K;MAAA,S;QAAkB,OnNuCE,amNvCF,KnNuCoB
,KAAlB,EmNvCO,SnNuCuB,KAA9B,CmNvCF,K;;MAAlB,W;K;qCAE/C,Y;MAKkC,OnNgCiC,amNhCjC,UnNg
CmD,KAAlB,EmNhCzB,SnNgCuD,KAA9B,CmNhCjC,I;K;oCAElC,iB;MAEY,UAAwB,M;MADhC,8CAAwB,k
BAAa,KAAM,UAAnB,KAChB,2CAAS,KAAM,MAAf,cAAwB,6CAAQ,KAAM,KAAd,QAAxB,CADgB,CAAxB
,C;K;sCAGJ,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,MnN2QK,CArCkB,UmNtOjB,UnNsO4B,KAAL,KAAo
B,CAVzB,UmN5NP,UnN4Na,yBmN5NH,EnN4NG,CAAN,CAUyB,MAApB,CAAN,CAqClB,MAAK,QmN3QV,

QnN2QK,CArCkB,UmNtOoB,SnNsOT,KAAL,KAAoB,CAVzB,UmN5N6B,SnN4NvB,yBmN5NgC,EnN4NhC,C
AAN,CAUyB,MAApB,CAAN,CAqClB,MAAK,QmN3QV,I;K;sCAE5B,Y;MAAkC,OAAE,UAAF,qBAAU,S;K;I
AE5C,gC;MAAA,oC;MACI,aAC+B,iBAAW,6BAAM,UAAjB,EAA4B,6BAAM,UAAlC,C;K;;;IAFnC,4C;MAAA,
2C;QAAA,0B;;MAAA,oC;K;;IAYJ,qD;MA4CI,wC;MAtCI,IAAI,gBAAJ,C;QAAwB,MAAa,gCAAyB,wBAAzB,C
;MACrC,IAAI,sCAAJ,C;QAA4B,MAAa,gCAAyB,yEAAzB,C;MAG7C,aAG0B,K;MAE1B,YAGyB,4BAA0B,KA
A1B,EAAiC,YAAjC,EAA+C,IAA/C,C;MAEzB,YAGwB,I;K;0CAExB,Y;MAAiD,oCAAyB,UAAzB,EAAgC,SAA
hC,EAAsC,SAAtC,C;K;yCAEjD,Y;MAMqC,OAAI,uBAAO,CAAX,GnNxB8B,amNwBhB,UnNxBkC,KAAlB,Em
NwBR,SnNxBsC,KAA9B,CmNwBhB,IAAd,GnNxB8B,amNwBE,UnNxBgB,KAAlB,EmNwBU,SnNxBoB,KAA9
B,CmNwBE,I;K;wCAErE,iB;MAEY,UAAwB,M;MADhC,kDAA8B,kBAAa,KAAM,UAAnB,KACtB,2CAAS,KA
AM,MAAf,cAAwB,6CAAQ,KAAM,KAAd,QAAxB,KAA8C,kBAAQ,KAAM,KAAd,CADxB,CAA9B,C;K;0CAG
J,Y;MACI,OAAI,cAAJ,GAAe,EAAf,GAAwB,OAAM,MnNmND,CArCkB,UmN9KX,UnN8KsB,KAAL,KAAoB,
CAVzB,UmNpKD,UnNoKO,yBmNpKG,EnNoKH,CAAN,CAUyB,MAApB,CAAN,CAqClB,MAAK,QmNnNJ,Qn
NmND,CArCkB,UmN9K0B,SnN8Kf,KAAL,KAAoB,CAVzB,UmNpKmC,SnNoK7B,yBmNpKsC,EnNoKtC,CAA
N,CAUyB,MAApB,CAAN,CAqClB,MAAK,QmNnNJ,IAAN,SAAqF,cAAU,6BAAU,EAAV,CAAV,CAAyB,QAA
9G,I;K;0CAE5B,Y;MAAkC,OAAI,uBAAO,CAAX,GAAgB,UAAF,qBAAU,SAAV,cAAqB,SAArB,WAAd,GAAg
D,UAAF,2BAAgB,SAAhB,cAA6B,SAAD,aAA5B,W;K;IAEhF,sC;MAAA,0C;K;mEACI,sC;MAQ+F,4BAAiB,U
AAjB,EAA6B,QAA7B,EAAuC,IAAvC,C;K;;;IATnG,kD;MAAA,iD;QAAA,gC;;MAAA,0C;K;;IAmBkC,qD;MAC
lC,sBAA2B,I;MAC3B,iBAAmC,kBAAO,CAA1C,GnNxDmE,amNwDtB,KnNxDwC,KAAlB,EmNwDb,InNxD2C,
KAA9B,CmNwDtB,KAA7C,GnNxDmE,amNwDH,KnNxDqB,KAAlB,EmNwDM,InNxDwB,KAA9B,CmNwDH,
K;MAChE,cnN4SsC,UmN5SnB,InN4SmB,C;MmN3StC,cAAuB,cAAJ,GAAa,KAAb,GAAwB,mB;K;iDAE3C,Y;
MAAkC,qB;K;8CAElC,Y;MACI,YAAY,W;MACZ,IAAI,6BAAS,mBAAT,QAAJ,C;QACI,IAAI,CAAC,cAAL,C;
UAAc,MAAa,6B;QAC3B,iBAAU,K;;QAEV,cnNvD+C,UmNuD/C,WnNvD0D,KAAK,KmNuDvD,WnNvDkE,KA
AX,CAAhB,C;;MmNyDnD,OAAO,K;K;;wECrIf,yB;MAAA,8C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,
CAAT,C;O;KAPX,C;wEAUA,yB;MAAA,8C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,
C;wEAUA,yB;MAAA,8C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,C;wEAUA,yB;MA
AA,8C;MAAA,uB;QAOI,OAAO,MAAM,CAAN,EAAS,CAAT,C;O;KAPX,C;oFC7BA,yB;MAAA,gD;MAAA,4B
;QAM6C,OAAQ,atOyShB,csOzSgB,C;O;KANrD,C;oGAQA,yB;MtHwCA,4B;MsHxCA,4B;QAMqD,OtHwCM,Y
hHyPtB,cgHzPsB,C;O;KsH9C3D,C;sGAQA,yB;MAAA,kE;MAAA,4B;QAMsD,OAAQ,sBtOyRzB,csOzRyB,C;O;
KAN9D,C;8FAQA,yB;MAAA,0D;MtOkXA,6B;MsOlXA,4B;QAOmD,OtOqXZ,csOrXoB,kBtOgRtB,csOhRsB,Ct
OqXpB,C;O;KsO5XvC,C;4FASA,yB;MAAA,wD;MtOyWA,6B;MsOzWA,4B;QAOkD,OtO4WX,csO5WmB,iBtO
uQrB,csOvQqB,CtO4WnB,C;O;KsOnXvC,C;gFASA,yB;MAAA,4C;MtOgWA,6B;MsOhWA,sC;QAayD,OtO6Vl
B,csO7V0B,WtOwP5B,csOxP4B,EAAW,QAAX,CtO6V1B,C;O;KsO1WvC,C;kFAgBA,yB;MAAA,8C;MtOgVA,6
B;MsOhVA,sC;QAa0D,OtO6UnB,csO7U2B,YtOwO7B,csOxO6B,EAAY,QAAZ,CtO6U3B,C;O;KsO1VvC,C;oFA
gBA,yB;MAAA,gD;MAAA,4B;QAM8C,OAAS,arN0OhB,cqN1OgB,C;O;KANvD,C;oGAQA,yB;MAAA,gE;MA
AA,4B;QAMsD,OAAS,qBrNkOxB,cqNlOwB,C;O;KAN/D,C;sGAQA,yB;MAAA,kE;MAAA,4B;QAMuD,OAAS,
sBrN0NzB,cqN1NyB,C;O;KANhE,C;8FAQA,yB;MAAA,0D;MrNuTA,+B;MqNvTA,4B;QAOqD,OrN0TX,eqN1T
oB,kBrNiNvB,cqNjNuB,CrN0TpB,C;O;KqNjU1C,C;4FASA,yB;MAAA,wD;MrN8SA,+B;MqN9SA,4B;QAOoD,
OrNiTV,eqNjTmB,iBrNwMtB,cqNxMsB,CrNiTnB,C;O;KqNxT1C,C;+EASA,yB;MAAA,4C;MrNqSA,+B;MqNrS
A,sC;QAa2D,OrNkSjB,eqNlS0B,WrNyL7B,cqNzL6B,EAAW,QAAX,CrNkS1B,C;O;KqN/S1C,C;iFAeA,yB;MtHg
EA,4C;M/FsNA,+B;MqNtRA,sC;QAa4D,OrNmRlB,e+FnNuB,W/F0G1B,c+F1G0B,EAAW,CsHhEK,QtHgEL,IA
AX,C/FmNvB,C;O;KqNhS1C,C;oFAeA,yB;MvOkKI,6B;MuOpTJ,gD;MAkJA,4B;QAM8C,OAlJO,atOyShB,CDc
E,cAAU,cAAL,GAAiB,GAAtB,CCdF,MsOzSgB,C;O;KA4IrD,C;oGAQA,yB;MtH1GA,4B;MsH0GA,4B;QAMsD,
OtH1GK,YjHiNpB,ciOpHe,GAAY,GhH7FP,CgH6FN,GAA6C,EAA7C,I;O;KMOrD,C;sGAQA,yB;MNbA,kE;MM
aA,4B;QAMuD,ONbkB,sBjO4GlC,ciO5GgB,GAAW,GAAO,C;O;KMOzE,C;8FAQA,yB;MAAA,0D;MvOyMA,0
B;MAAA,+B;MuOzMA,4B;QAOqD,OvO6MZ,eAAW,OuO7MS,kBvO0GnB,cAAL,GAAiB,GuO1GO,CvO6MT,C
AAX,C;O;KuOpNzC,C;4FASA,yB;MAAA,wD;MvOgMA,0B;MAAA,+B;MuOhMA,4B;QAOoD,OvOoMX,eAA
W,OuOpMQ,iBvOiGlB,cAAL,GAAiB,GuOjGM,CvOoMR,CAAX,C;O;KuO3MzC,C;gFAUA,yB;MAAA,4C;MvO
+JA,+B;MuO/JA,sC;QAa2D,OvO4JjB,euO5J0B,WvO6D7B,cuO7D6B,EAAW,QAAX,CvO4J1B,C;O;KuOzK1C,C

;kFAeA,yB;MAAA,8C;MvOgJA,+B;MuOhJA,sC;QAa4D,OvO6IlB,euO7I2B,YvO8C9B,cuO9C8B,EAAY,QAAZ,
CvO6I3B,C;O;KuO1J1C,C;oFAeA,yB;MrO0FI,6B;MqOrTJ,gD;MA2NA,4B;QAM+C,OA3NM,atOyShB,CCeE,cA
AU,cAAL,GAAiB,KAAtB,CDfF,MsOzSgB,C;O;KAqNrD,C;oGAQA,yB;MtHnLA,4B;MsHmLA,4B;QAMuD,Ot
HnLI,Y/G4NlB,c+NrDpC,GAAY,KhHvK0C,CgHuKvD,GAA+C,EAA/C,I;O;KMMJ,C;sGAQA,yB;MNZA,kE;M
MYA,4B;QAMwD,ONZoB,sB/N6CnC,c+N7Ce,GAAW,KAAS,C;O;KMM5E,C;8FAQA,yB;MAAA,0D;MrOiIA,4
B;MAAA,iC;MqOjIA,4B;QAOuD,OrOqIZ,gBAAY,QqOrIQ,kBrOkCrB,cAAL,GAAiB,KqOlCS,CrOqIR,CAAZ,C;
O;KqO5I3C,C;4FASA,yB;MAAA,wD;MrOwHA,4B;MAAA,iC;MqOxHA,4B;QAOsD,OrO4HX,gBAAY,QqO5H
O,iBrOyBpB,cAAL,GAAiB,KqOzBQ,CrO4HP,CAAZ,C;O;KqOnI3C,C;gFAUA,yB;MAAA,4C;MrOmGA,iC;Mq
OnGA,sC;QAa6D,OrOgGhB,gBqOhG0B,WrOD9B,cqOC8B,EAAW,QAAX,CrOgG1B,C;O;KqO7G7C,C;kFAeA,
yB;MAAA,8C;MrOoFA,iC;MqOpFA,sC;QAa8D,OrOiFjB,gBqOjF2B,YrOhB/B,cqOgB+B,EAAY,QAAZ,CrOiF3
B,C;O;KqO9F7C,C;ICtRA,qC;MAEI,SvOuIoD,cuOvI3C,CvOuI2C,EuOvIvC,CvOuIuC,C;MuOtIpD,SvOsIoD,cuO
tI3C,CvOsI2C,EuOtIvC,CvOsIuC,C;MuOrIpD,OvOmDkE,YuOnDvD,EvOmDwE,KAAjB,EuOnDjD,EvOmD8E,K
AA7B,CuOnDvD,KAAX,GvOkFsD,SuOlFjC,EvOkF2C,KAAK,GuOlF3C,EvOkFuD,KAAZ,IAAf,CuOlFtD,GvOq
EqD,SAAU,CAaT,SuOlFpB,EvOkF8B,KAAK,GuOlF9B,EvOkF0C,KAAZ,IAAf,CAbS,MAAK,GuOrExB,CvOqE
mC,KAAX,IAAf,C;K;IuOlEzD,qC;MACI,StNwIsD,esNxI7C,CtNwI6C,EsNxIzC,CtNwIyC,C;MsNvItD,StNuIsD,e
sNvI7C,CtNuI6C,EsNvIzC,CtNuIyC,C;MsNtItD,OtNqDmE,asNrDxD,EtNqD0E,KAAlB,EsNrDlD,EtNqDgF,KAA
9B,CsNrDxD,KAAX,GtN+EwD,UsN/EnC,EtN+E8C,KAAK,UsN/E9C,EtN+E0D,KAAZ,CAAhB,CsN/ExD,GtNk
EuD,UAAW,CAaV,UsN/EtB,EtN+EiC,KAAK,UsN/EjC,EtN+E6C,KAAZ,CAAhB,CAbU,MAAK,KsNlE3B,CtNk
EsC,KAAX,CAAhB,C;K;IsN/D3D,uD;MAmBI,WAAO,CAAP,C;QAD8E,OvOwBZ,YuOvBlD,KvOuBmE,KAAjB
,EuOvBzC,GvOuBsE,KAA7B,CuOvBlD,KAD8D,GAChD,GADgD,GvOuDxB,SuOtDf,GvOsDyB,KAAK,GuOtDx
B,mBAAiB,GAAjB,EAAsB,KAAtB,EvOqXV,SuOrXuC,IvOqXvC,CuOrXU,CvOsDoC,KAAZ,IAAf,C;auOrDtD,
WAAO,CAAP,C;QAF8E,OvOwBZ,YuOtBlD,KvOsBmE,KAAjB,EuOtBzC,GvOsBsE,KAA7B,CuOtBlD,KAF8D,
GAEhD,GAFgD,GvO0CzB,SuOxCd,GvOwCwB,KAAK,GuOxCvB,mBAAiB,KAAjB,EAAwB,GAAxB,EvOoXV,
SuOpXwC,CAAC,IAAD,IvOoXxC,CuOpXU,CvOwCkC,KAAX,IAAf,C;;QuOvC7C,MAAa,gCAAyB,eAAzB,C;K
;IAGzB,uD;MAmBI,sBAAO,CAAP,C;QADkF,OtNQf,asNPnD,KtNOqE,KAAlB,EsNP1C,GtNOwE,KAA9B,CsNP
nD,KADkE,GACpD,GADoD,GtNkC1B,UsNjCjB,GtNiC4B,KAAK,UsNjC3B,mBAAiB,GAAjB,EAAsB,KAAtB,E
tN4WP,UsN5WoC,ItN4WpC,CsN5WO,CtNiCuC,KAAZ,CAAhB,C;asNhCxD,sBAAO,CAAP,C;QAFkF,OtNQf,as
NNnD,KtNMqE,KAAlB,EsNN1C,GtNMwE,KAA9B,CsNNnD,KAFkE,GAEpD,GAFoD,GtNqB3B,UsNnBhB,GtN
mB2B,KAAK,KsNnB1B,mBAAiB,KAAjB,EAAwB,GAAxB,EtN2WP,UsN3WsC,IAAD,atN2WrC,CsN3WO,CtN
mBqC,KAAX,CAAhB,C;;QsNlB/C,MAAa,gCAAyB,eAAzB,C;K;ItOlDC,sB;MAEtB,8B;MAFmG,gB;K;IAEnG,4
B;MAAA,gC;MACI,iBAGqC,WAAO,CAAP,C;MAErC,iBAGqC,WAAO,MAAP,C;MAErC,kBAGmC,C;MAEnC,
iBAGkC,E;K;;;IAnBtC,wC;MAAA,uC;QAAA,sB;;MAAA,gC;K;wGAsBA,iB;MAM0D,OAAa,0BAuPjC,SAAL,G
AAiB,KAvPqB,EAAU,KFsP3C,KAAL,GAAiB,GEtPqB,C;K;oGAEvE,iB;MAOoE,OAAa,0BA8O3C,SAAL,GAAi
B,KA9O+B,EAAU,KA8OrD,KAAL,GAAiB,KA9O+B,C;K;wGAEjF,yB;MAqQA,6B;MDtQA,8C;MCCA,wB;QA
MyD,ODAS,YAAiB,CCwQhD,cAAU,SAAL,GAAiB,KAAtB,CDxQgD,MAAjB,ECAe,KDAc,KAA7B,C;O;KCNl
E,C;wGAQA,yB;MAuQA,aAS6D,0B;MAT7D,+B;MgBxQA,gD;MhBCA,wB;QAM0D,OgBAS,aAAkB,ChB0QhD
,eAAW,oBAAL,SAAK,CAAL,YAAN,CgB1QgD,MAAlB,EhBAgB,KgBAc,KAA9B,C;O;KhBNnE,C;8FAQA,yB;
MAqPA,6B;MArPA,wB;QAEsD,ODMD,cAAU,CCsP5B,cAAU,SAAL,GAAiB,KAAtB,CDtP4B,MAAK,GAAW,
CDqP5C,cE3PsC,KF2P5B,KAAL,GAAiB,GAAtB,CCrP4C,MAAX,IAAf,C;O;KCRrD,C;8FAGA,yB;MAkPA,6B;
MAlPA,wB;QAEuD,ODGF,cAAU,CCsP5B,cAAU,SAAL,GAAiB,KAAtB,CDtP4B,MAAK,GAAW,CCsP5C,cAzP
uC,KAyP7B,KAAL,GAAiB,KAAtB,CDtP4C,MAAX,IAAf,C;O;KCLrD,C;8FAGA,yB;MA+OA,6B;MA/OA,wB;Q
AEqD,ODAA,cAAU,CCsP5B,cAAU,SAAL,GAAiB,KAAtB,CDtP4B,MAAK,GCAI,KDAO,KAAX,IAAf,C;O;KC
FrD,C;8FAGA,yB;MAsPA,aAS6D,0B;MAT7D,+B;MAtPA,wB;QAEuD,OgBAA,eAAW,ChB6P7B,eAAW,oBAA
L,SAAK,CAAL,YAAN,CgB7P6B,MAAK,KhBAI,KgBAO,KAAX,CAAhB,C;O;KhBFvD,C;gGAIA,yB;MAwOA,
6B;MAxOA,wB;QAEuD,ODMD,cAAU,CCyO7B,cAAU,SAAL,GAAiB,KAAtB,CDzO6B,MAAK,GAAY,CDwO9
C,cE9OwC,KF8O9B,KAAL,GAAiB,GAAtB,CCxO8C,MAAZ,IAAf,C;O;KCRtD,C;gGAGA,yB;MAqOA,6B;MAr
OA,wB;QAEwD,ODGF,cAAU,CCyO7B,cAAU,SAAL,GAAiB,KAAtB,CDzO6B,MAAK,GAAY,CCyO9C,cA5Oy
C,KA4O/B,KAAL,GAAiB,KAAtB,CDzO8C,MAAZ,IAAf,C;O;KCLtD,C;gGAGA,yB;MAkOA,6B;MAlOA,wB;Q

AEsD,ODAA,cAAU,CCyO7B,cAAU,SAAL,GAAiB,KAAtB,CDzO6B,MAAK,GCAK,KDAO,KAAZ,IAAf,C;O;K
CFtD,C;gGAGA,yB;MAyOA,aAS6D,0B;MAT7D,+B;MAzOA,wB;QAEwD,OgBAA,eAAW,ChBgP9B,eAAW,oB
AAL,SAAK,CAAL,YAAN,CgBhP8B,MAAK,UhBAK,KgBAO,KAAZ,CAAhB,C;O;KhBFxD,C;gGAIA,yB;MA2
NA,6B;MA3NA,wB;QAEuD,ODMD,cAAe,YAAL,CC4N7B,cAAU,SAAL,GAAiB,KAAtB,CD5N6B,MAAK,EA
AY,CD2N9C,cEjOwC,KFiO9B,KAAL,GAAiB,GAAtB,CC3N8C,MAAZ,CAAf,C;O;KCRtD,C;gGAGA,yB;MAw
NA,6B;MAxNA,wB;QAEwD,ODGF,cAAe,YAAL,CC4N7B,cAAU,SAAL,GAAiB,KAAtB,CD5N6B,MAAK,EAA
Y,CC4N9C,cA/NyC,KA+N/B,KAAL,GAAiB,KAAtB,CD5N8C,MAAZ,CAAf,C;O;KCLtD,C;gGAGA,yB;MAqN
A,6B;MArNA,wB;QAEsD,ODAA,cAAe,YAAL,CC4N7B,cAAU,SAAL,GAAiB,KAAtB,CD5N6B,MAAK,ECAK,
KDAO,KAAZ,CAAf,C;O;KCFtD,C;gGAGA,yB;MA4NA,aAS6D,0B;MAT7D,+B;MA5NA,wB;QAEwD,OgBAA,
eAAW,ChBmO9B,eAAW,oBAAL,SAAK,CAAL,YAAN,CgBnO8B,MAAK,UhBAK,KgBAO,KAAZ,CAAhB,C;O
;KhBFxD,C;4FAIA,yB;MA8MA,6B;MDxMA,4C;MCNA,wB;QAEqD,ODMD,WC+MjB,cAAU,SAAL,GAAiB,K
AAtB,CD/MiB,ED8MjB,cEpNoC,KFoN1B,KAAL,GAAiB,GAAtB,CC9MiB,C;O;KCRpD,C;4FAGA,yB;MA2MA
,6B;MDxMA,4C;MCHA,wB;QAEsD,ODGF,WC+MjB,cAAU,SAAL,GAAiB,KAAtB,CD/MiB,EC+MjB,cAlNqC,
KAkN3B,KAAL,GAAiB,KAAtB,CD/MiB,C;O;KCLpD,C;4FAGA,yB;MAwMA,6B;MDxMA,4C;MCAA,wB;QA
EoD,ODAA,WC+MjB,cAAU,SAAL,GAAiB,KAAtB,CD/MiB,ECAkB,KDAlB,C;O;KCFpD,C;4FAGA,yB;MA+M
A,aAS6D,0B;MAT7D,+B;MgB/MA,8C;MhBAA,wB;QAEsD,OgBAA,YhBsNjB,eAAW,oBAAL,SAAK,CAAL,Y
AAN,CgBtNiB,EhBAmB,KgBAnB,C;O;KhBFtD,C;4FAIA,yB;MAiMA,6B;MDnLA,kD;MCdA,wB;QAMqD,ODc
D,cCsLjB,cAAU,SAAL,GAAiB,KAAtB,CDtLiB,EDqLjB,cEnMoC,KFmM1B,KAAL,GAAiB,GAAtB,CCrLiB,C;
O;KCpBpD,C;4FAOA,yB;MA0LA,6B;MDnLA,kD;MCPA,wB;QAMsD,ODOF,cCsLjB,cAAU,SAAL,GAAiB,KA
AtB,CDtLiB,ECsLjB,cA7LqC,KA6L3B,KAAL,GAAiB,KAAtB,CDtLiB,C;O;KCbpD,C;4FAOA,yB;MAmLA,6B;
MDnLA,kD;MCAA,wB;QAMoD,ODAA,cCsLjB,cAAU,SAAL,GAAiB,KAAtB,CDtLiB,ECAkB,KDAlB,C;O;KC
NpD,C;4FAOA,yB;MAsLA,aAS6D,0B;MAT7D,+B;MgBtLA,oD;MhBAA,wB;QAMsD,OgBAA,ehByLjB,eAAW,
oBAAL,SAAK,CAAL,YAAN,CgBzLiB,EhBAmB,KgBAnB,C;O;KhBNtD,C;sGAQA,yB;MAoKA,6B;MDxMA,4
C;MCoCA,wB;QAMiD,ODxCG,WC+MjB,cAAU,SAAL,GAAiB,KAAtB,CD/MiB,ED8MjB,cEtKqC,KFsK3B,KA
AL,GAAiB,GAAtB,CC9MiB,C;O;KCkCpD,C;sGAOA,yB;MA6JA,6B;MDxMA,4C;MC2CA,wB;QAMkD,OD/CE
,WC+MjB,cAAU,SAAL,GAAiB,KAAtB,CD/MiB,EC+MjB,cAhKsC,KAgK5B,KAAL,GAAiB,KAAtB,CD/MiB,C
;O;KCyCpD,C;sGAOA,yB;MAsJA,6B;MDxMA,4C;MCkDA,wB;QAMgD,ODtDI,WC+MjB,cAAU,SAAL,GAAiB
,KAAtB,CD/MiB,ECsDmB,KDtDnB,C;O;KCgDpD,C;sGAOA,yB;MAyJA,aAS6D,0B;MAT7D,+B;MgB/MA,8C;
MhBsDA,wB;QAMkD,OgB1DI,YhBsNjB,eAAW,oBAAL,SAAK,CAAL,YAAN,CgBtNiB,EhB0DoB,KgB1DpB,C
;O;KhBoDtD,C;4FAQA,yB;MAuIA,6B;MDnLA,kD;MDiPJ,0B;MAAA,+B;MErMI,wB;QAQ6C,OFwMR,eAAW,
OCtPI,cCsLjB,cAAU,SAAL,GAAiB,KAAtB,CDtLiB,EDqLjB,cEvI4B,KFuIlB,KAAL,GAAiB,GAAtB,CCrLiB,C
A4Lf,KD0DW,CAAX,C;O;KEhNrC,C;4FASA,yB;MA8HA,6B;MDnLA,kD;MCkPJ,4B;MAAA,iC;MA7LI,wB;Q
AQ+C,OAgMR,gBAAY,QDvPC,cCsLjB,cAAU,SAAL,GAAiB,KAAtB,CDtLiB,ECsLjB,cA/H8B,KA+HpB,KAA
L,GAAiB,KAAtB,CDtLiB,CAsMb,KCiDY,CAAZ,C;O;KAxMvC,C;4FASA,yB;MAqHA,6B;MDnLA,kD;MC8DA
,wB;QAQ2C,ODhES,cCsLjB,cAAU,SAAL,GAAiB,KAAtB,CDtLiB,ECgES,KDhET,C;O;KCwDpD,C;4FASA,yB;
MAsHA,aAS6D,0B;MAT7D,+B;MgBtLA,oD;MhBgEA,wB;QAQ6C,OgBlES,ehByLjB,eAAW,oBAAL,SAAK,CA
AL,YAAN,CgBzLiB,EhBkEU,KgBlEV,C;O;KhB0DtD,C;4EAUA,yB;MAAA,4B;MAAA,iC;MAAA,mB;QAM2C,
uBAAY,QAAL,SAAK,KAAZ,C;O;KAN3C,C;4EAQA,yB;MAAA,4B;MAAA,iC;MAAA,mB;QAM2C,uBAAY,Q
AAL,SAAK,KAAZ,C;O;KAN3C,C;oGAQA,yB;MAAA,8C;MAkFA,6B;MAlFA,wB;QAE+D,0BAyF5B,cAAU,S
AAL,GAAiB,KAAtB,CAzF4B,EAyF5B,cAzFqD,KAyF3C,KAAL,GAAiB,KAAtB,CAzF4B,C;O;KAF/D,C;0GAI
A,yB;MAAA,yC;MA8EA,6B;MA9EA,wB;QAQkE,aA+E/B,cAAU,SAAL,GAAiB,KAAtB,CA/E+B,EA+E/B,cA/E
mD,KA+EzC,KAAL,GAAiB,KAAtB,CA/E+B,C;O;KARlE,C;4FAUA,yB;MAAA,iC;M4L7NJ,4B;M5L6NI,wB;Q
AEqD,uB4L5NiC,Q5L4N1B,IAAK,K4L5NX,G5L4NoB,KAAM,K4L5NM,C5L4NjC,C;O;KAFrD,C;0FAGA,yB;
MAAA,iC;M4L3NJ,4B;M5L2NI,wB;QAEoD,uB4L1NgC,Q5L0NzB,IAAK,K4L1NX,G5L0NmB,KAAM,K4L1N
M,C5L0NhC,C;O;KAFpD,C;4FAGA,yB;MAAA,iC;M4LzNJ,4B;M5LyNI,wB;QAEqD,uB4LxNiC,Q5LwN1B,IAA
K,K4LxNX,G5LwNoB,KAAM,K4LxNM,C5LwNjC,C;O;KAFrD,C;4EAGA,yB;MAAA,iC;M4LvNJ,4B;M5LuNI,
mB;QAEkC,uB4LtNsB,QAAP,C5LsNR,S4LtNe,C5LsNtB,C;O;KAFlC,C;kFAIA,yB;MAAA,0B;MAAA,mB;QAU
mC,OAAK,OAAL,SAAK,C;O;KAVxC,C;oFAWA,Y;MASqC,gB;K;gFACrC,Y;MASiC,OAAK,SAAL,GAAiB,K;

K;kFAClD,yB;MAAA,aASqD,0B;MATrD,mB;QASmC,OAAK,oBAAL,SAAK,CAAL,Y;O;KATnC,C;oFAWA,y
B;MF+DJ,0B;MAAA,+B;ME/DI,mB;QASqC,OFiEE,eAAW,OEjEb,SFiEa,CAAX,C;O;KE1EvC,C;sFAUA,Y;MA
EuC,W;K;kFACvC,yB;MAAA,6B;MAAA,mB;QASmC,qBAAU,SAAL,GAAAiB,KAAtB,C;O;KATnC,C;oFAUA,y
B;MAAA,aAS6D,0B;MAT7D,+B;MAAA,mB;QASqC,sBAAW,oBAAL,SAAK,CAAL,YAAN,C;O;KATrC,C;oFA
WA,Y;MAMqC,OApDC,SAAL,GAAiB,K;K;sFAqDlD,Y;MAMuC,OA3DD,SAAL,GAAiB,K;K;gCA6DlD,Y;MA
AyC,OAAQ,CA7DX,SAAL,GAAiB,KA6DD,Y;K;;;;;gCArVrD,Y;MAAA,c;MAGuG,qD;MAHvG,a;K;8BAAA,iB;
MAAA,2IAGuG,oCAHvG,G;K;0EAyVA,yB;MAAA,iC;MAAA,4B;QAW4C,uBAAY,SAAZ,C;O;KAX5C,C;4EA
YA,yB;MAAA,iC;MAAA,4B;QAU6C,uBAAO,SAAP,C;O;KAV7C,C;4EAWA,yB;MAAA,4B;MAAA,iC;MAAA,
4B;QAW2C,uBAAY,QAAL,SAAK,CAAZ,C;O;KAX3C,C;4EAYA,yB;MAAA,4B;MAAA,iC;MAAA,4B;QAW4
C,uBAAY,QAAL,SAAK,SAAZ,C;O;KAX5C,C;IkCzXA,8B;MACqB,sB;K;wCAKjB,iB;MAM8C,OlCgWL,WkCh
WK,aAAQ,KAAR,ClCgWL,C;K;wCkC9VzC,wB;MAOI,aAAQ,KAAR,IAAiB,KlCsPgB,K;K;mFkClPP,Y;MAAQ
,OAAA,YAAQ,O;K;qCAE9C,Y;MAC+E,gCAAS,YAAT,C;K;IAEzD,qC;MAAC,oB;MACnB,eAAoB,C;K;6CACp
B,Y;MAAyB,sBAAQ,YAAM,O;K;0CACvC,Y;MAAoD,Q;MAA9B,IAAI,eAAQ,YAAM,OAAlB,C;QAAA,OlC2
Ue,WkC3US,aAAM,mBAAN,EAAM,2BAAN,OlC2UT,C;;QkC3UwC,MAAM,2BAAuB,YAAM,WAA7B,C;K;;2
CAGvF,mB;MAIS,Q;MAAL,IAAI,eAAC,0EAAD,SAAJ,C;QAAkC,OAAO,K;MAEzC,OAAe,WAAR,YAAQ,EA
AS,OlCiOS,KkCjOlB,C;K;gDAGnB,oB;MACY,Q;MAA2B,gBAA3B,gE;MAA2B,c;;QjBioDvB,U;QADhB,IAAI,
wCAAsB,mBAA1B,C;UAAqC,aAAO,I;UAAP,e;;QACrB,6B;QAAhB,OAAgB,gBAAhB,C;UAAgB,2B;UiBjoD6B
,2BjBioDR,OiBjoDQ,S;UAAA,W;YAAwB,oBAAR,YAAQ,EjBioDhC,OjBp6CA,KkC7NgC,C;;UjBioD/C,IAAI,O
AAJ,C;YAAyB,aAAO,K;YAAP,e;;;QAC/C,aAAO,I;;;MiBloDH,iB;K;oCAGJ,Y;MAAkC,OAAA,IAAK,QAAQ,O
AAb,KAAqB,C;K;;IA9CvD,uC;MAAA,qD;MACgC,wBAAK,eAAW,IAAX,CAAL,C;MADhC,Y;K;;;;qCAPJ,Y;M
AAA,OAKqB,sDALrB,M;K;qCAAA,Y;MAAA,c;MAKqB,wD;MALrB,a;K;mCAAA,iB;MAAA,2IAKqB,0CALrB
,G;K;kFAwDA,yB;MAAA,2C;MAWwC,0C;QAAA,wB;UAAW,OAAA,aAAK,KAAL,ClC4MV,K;S;O;MkCvNzC
,6B;QAWI,OAAO,qBAAY,gCAAW,IAAX,GAAAiB,wBAAjB,CAAZ,C;O;KAXX,C;oFAcA,oB;MAGwE,e;K;IqM
3ExE,sC;MAQ2D,OAAa,WAAb,SzOkRjB,KAAL,GAAAiB,GyOlRkB,EAAS,KAAT,C;K;IAExE,sC;MAQ4D,OAA
a,WAAb,SvOyQlB,KAAL,GAAAiB,KuOzQmB,EAAS,KAAT,C;K;IAGzE,sC;MAQ0D,OAAc,WxO2R5B,oBwO3R
c,SxO2RnB,KAAK,CAAL,iBwO3RiC,EAAS,KAAT,C;K;IAExE,sC;MAOgD,uBAAc,SvNmRvB,KuNnRS,EAA6
B,WAAW,KAAX,CAA7B,C;K;IAGhD,8B;MAMqC,Q;MAAA,0DAAmB,kBAAkB,SAAlB,C;K;IAExD,qC;MAO
+C,Q;MAAA,0CAAc,KAAd,oBAAwB,kBAAkB,SAAlB,C;K;IAGvE,+B;MAMuC,Q;MAAA,2DAAoB,kBAAkB,
SAAlB,C;K;IAE3D,sC;MAOiD,Q;MAAA,2CAAe,KAAf,oBAAyB,kBAAkB,SAAlB,C;K;IAE1E,6B;MAMmC,Q;
MAAA,yDAAkB,kBAAkB,SAAlB,C;K;IAErD,oC;MAO6C,Q;MAAA,yCAAa,KAAb,oBAAuB,kBAAkB,SAAlB,
C;K;IAEpE,8B;MAMqC,Q;MAAA,0DAAmB,kBAAkB,SAAlB,C;K;IAExD,qC;MAO+C,Q;MAAA,0CAAc,KAA
d,oBAAwB,kBAAkB,SAAlB,C;K;IAMvE,kC;MAM4C,kCAAsB,EAAtB,C;K;IAE5C,2C;MAS0C,IAAvB,I;MAA
A,sBAAL,SAAK,EAAa,KAAb,C;MAAL,iB;QAA4B,OAAO,I;;MAA7C,UAAU,I;MACV,IxO/EkE,YwO+E9D,Gx
O/E+E,KAAjB,EAA6B,CDuQ5D,SyOxLzB,6BAAM,UzOwL6B,KAAL,GAAAiB,GAAtB,CCvQ4D,MAA7B,CwO+
E9D,IAAJ,C;QAA2B,OAAO,I;MAClC,OzOwPqC,UAAW,OyOxPzC,GxO8L8B,KD0DW,CAAX,C;K;IyOrPzC,m
C;MAM8C,mCAAuB,EAAvB,C;K;IAE9C,4C;MAS0C,IAAvB,I;MAAA,sBAAL,SAAK,EAAa,KAAb,C;MAAL,i
B;QAA4B,OAAO,I;;MAA7C,UAAU,I;MACV,IxOrGkE,YwOqG9D,GxOrG+E,KAAjB,EAA6B,CCwQ5D,SuOnK
zB,8BAAO,UvOmK4B,KAAL,GAAAiB,KAAtB,CDxQ4D,MAA7B,CwOqG9D,IAAJ,C;QAA4B,OAAO,I;MACnC,
OvOmOuC,WAAY,QuOnO5C,GxOkLgC,KCiDY,CAAZ,C;K;IuOhO3C,iC;MAM0C,iCAAqB,EAArB,C;K;IAE1C
,0C;MASI,WAAW,KAAX,C;MAEA,aAAa,SAAK,O;MACAlB,IAAI,WAAU,CAAd,C;QAAiB,OAAO,I;MAExB,Y
AAkB,4BAAK,U;MACvB,S;MAEA,gBAAgB,qBAAK,CAAL,C;MAChB,IAAI,YAAY,EAAhB,C;QACI,IAAI,W
AAU,CAAV,IAAe,cAAa,EAAhC,C;UAAqC,OAAO,I;QAC5C,QAAQ,C;;QAER,QAAQ,C;;MAGZ,uBAAuB,mB;
MAEvB,qBAAqB,gB;MACrB,axOiNmC,SwOjNtB,KxOiNsB,C;MwOhNnC,aAAa,W;MACb,aAAU,KAAV,MAA
sB,MAAtB,M;QACI,YAAY,QAAQ,qBAAK,CAAL,CAAR,EAAiB,KAAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAA
e,OAAO,I;QACtB,IxOnJ8D,YwOmJ1D,MxOnJ2E,KAAjB,EwOmJjD,cxOnJ8E,KAA7B,CwOmJ1D,IAAJ,C;UACI
,IAAI,+CAAkB,gBAAlB,QAAJ,C;YACI,iBxO5FwC,WwO4FvB,KxO5FuB,EwO4Ff,MxO5Fe,C;YwO8FxC,IxOvJ
sD,YwOuJlD,MxOvJmE,KAAjB,EwOuJzC,cxOvJsE,KAA7B,CwOuJlD,IAAJ,C;cACI,OAAO,I;;;YAGX,OAAO,I;
;;QAIf,SxOnHkD,SAAe,YwOmHjE,MxOnH4D,KAAK,EwOmHvD,MxOnHmE,KAAZ,CAAf,C;QwOqHlD,mBA

AmB,M;QACnB,SxOhJiD,SwOgJjD,MxOhJ2D,KAAK,GAAW,CA4U5C,SwO5LrB,KxO4LqB,CA5U4C,MAAX,I
AAf,C;QwOiJjD,IxOnK8D,YwOmK1D,MxOnK2E,KAAjB,EwOmKjD,YxOnK8E,KAA7B,CwOmK1D,IAAJ,C;U
AA2B,OAAO,I;;MAGtC,OAAO,M;K;IAGX,kC;MAM4C,kCAAsB,EAAtB,C;K;IAE5C,2C;MASI,WAAW,KAAX
,C;MAEA,aAAa,SAAK,O;MAClB,IAAI,WAAU,CAAd,C;QAAiB,OAAO,I;MAExB,YAAmB,6BAAM,U;MACzB
,S;MAEA,gBAAgB,qBAAK,CAAL,C;MAChB,IAAI,YAAY,EAAhB,C;QACI,IAAI,WAAU,CAAV,IAAe,cAAa,E
AAhC,C;UAAqC,OAAO,I;QAC5C,QAAQ,C;;QAER,QAAQ,C;;MAIZ,uBAAuB,gD;MAEvB,qBAAqB,gB;MACr
B,avNoJqC,UAAW,oBuNpJnC,KvNoJmC,CAAX,C;MuNnJrC,aAAa,2B;MACb,aAAU,KAAV,MAAsB,MAAtB,
M;QACI,YAAY,QAAQ,qBAAK,CAAL,CAAR,EAAiB,KAAjB,C;QAEZ,IAAI,QAAQ,CAAZ,C;UAAe,OAAO,I;
QACtB,IvN5M+D,auN4M3D,MvN5M6E,KAAlB,EuN4MlD,cvN5MgF,KAA9B,CuN4M3D,IAAJ,C;UACI,IAAI,+
CAAkB,gBAAlB,QAAJ,C;YACI,iBvN1J0C,YuN0JzB,KvN1JyB,EuN0JjB,MvN1JiB,C;YuN4J1C,IvNhNuD,auNg
NnD,MvNhNqE,KAAlB,EuNgN1C,cvNhNwE,KAA9B,CuNgNnD,IAAJ,C;cACI,OAAO,I;;;YAGX,OAAO,I;;;QAI
f,SvNjLoD,UuNiLpD,MvNjL+D,KAAK,UuNiL1D,MvNjLsE,KAAZ,CAAhB,C;QuNmLpD,mBAAmB,M;QACnB
,SvN9MmD,UuN8MnD,MvN9M8D,KAAK,KAAW,CjBgR7C,UAAW,oBAAL,CAyDR,SwO3HrB,KxO2HqB,CA
zDQ,MAAK,CAAL,iBAAN,CiBhR6C,MAAX,CAAhB,C;QuN+MnD,IvN5N+D,auN4N3D,MvN5N6E,KAAlB,Eu
N4NlD,YvN5NgF,KAA9B,CuN4N3D,IAAJ,C;UAA2B,OAAO,I;;MAGtC,OAAO,M;K;IhO9RX,6B;MACkD,OAA
uB,0BAAtB,KAAO,WAAe,EAAU,KAAO,WAAjB,C;K;IACzE,8B;MACqD,OAAC,gCAAuB,iBAAU,gCAAV,C;
K;IAE7E,4B;MACoD,OR2ZZ,SAvGI,oBQpTS,ERoTd,KAAK,CAAL,iBQpTc,KRoTT,oBQpTuB,ERoT5B,KAAK
,CAAL,iBQpTc,CR2ZH,QAAV,C;K;IQ1ZxC,+B;MACuD,ORyZf,SAvGI,oBQlTY,ERkTjB,KAAK,CAAL,iBQlTi
B,QRkTZ,oBQlT0B,ERkT/B,KAAK,CAAL,iBQlTiB,CRyZN,QAAV,C;K;IQpZxC,6B;MAEI,eAAe,ES4SoB,K;M
T3SnC,cAAc,ES2SqB,K;MT1SnC,IAAI,qBAAU,CAAd,C;QACI,OS6C+D,aT7CpD,ES6CsE,KAAlB,ET7C/C,ES6
C6E,KAA9B,CT7CpD,IAAJ,GAAa,aAAb,GAA2B,a;;MAItC,IAAI,uBAAY,CAAhB,C;QACI,OAAO,UAAM,aAA
W,OAAX,CAAN,C;;MAIX,eAAiB,4BAAc,CAAd,CAAD,KAAoB,OAApB,CAAD,WAAkC,CAAlC,C;MACf,UA
AU,kBAAW,kBAAW,OAAX,CAAX,C;MACV,OAAO,UAAM,iCSkCsD,aAAkB,CTlCzD,UAAM,GAAN,CSkCy
D,MAAlB,EAA8B,CTlCvD,UAAM,OAAN,CSkCuD,MAA9B,CTlCvC,KAAJ,GAAkC,CAAlC,GAAyC,CAApD,
EAAN,C;K;IAIX,gC;MAKe,Q;MAHX,eAAe,ESwRoB,K;MTvRnC,cAAc,ESuRqB,K;MTtRnC,IAAI,qBAAU,CA
Ad,C;QACW,ISyBwD,aTzBpD,ESyBsE,KAAlB,ETzB/C,ESyB6E,KAA9B,CTzBpD,IAAJ,C;UACH,S;;UAEA,OS
gDgD,UThDhD,ESgD2D,KAAK,UThD3D,ESgDuE,KAAZ,CAAhB,C;;QTnDpD,W;;MAQJ,IAAI,uBAAY,CAAh
B,C;QACI,OAAO,UAAM,gBAAW,OAAX,CAAN,C;;MAIX,eAAiB,4BAAc,CAAd,CAAD,KAAoB,OAApB,CAA
D,WAAkC,CAAlC,C;MACf,UAAU,kBAAW,kBAAW,OAAX,CAAX,C;MACV,OAAO,UAAM,aSUsD,aAAkB,C
TV9D,UAAM,GAAN,CSU8D,MAAlB,EAA8B,CTV5D,UAAM,OAAN,CSU4D,MAA9B,CTV5C,KAAJ,GAAkC,
OAAlC,KAAN,CAAN,C;K;IAGX,yB;MAEI,IAAE,QAAF,CAAE,CAAF,C;QADyC,OAC5B,W;;QACb,SRkTuC,a
QlTlC,4BAAK,URkT0C,KAAb,CQlTvC,C;UAFyC,OAEP,4BAAK,U;;UACvC,SRiTuC,aQjTlC,4BAAK,URiT0C,
KAAb,CQjTvC,C;YAHyC,OAGP,4BAAK,U;eACvC,SAAK,UAAL,C;YAJyC,OR4VN,SQxVX,YAAF,CAAE,CR
wVW,C;;YQ5VM,ORgBY,SAAU,CA4U5B,SQvVP,YAAnB,IAAI,UAAe,CRuVO,CA5U4B,MAAK,GAAW,CA4
U5C,SQvVY,URuVZ,CA5U4C,MAAX,IAAf,C;;;;K;IQRzD,0B;MAEI,IAAE,QAAF,CAAE,CAAF,C;QAD2C,OA
C9B,2B;;QACb,SS4SuC,cT5SlC,6BAAM,US4S0C,KAAd,CT5SvC,C;UAF2C,OAER,6BAAM,U;;UACzC,SS2Su
C,cT3SlC,6BAAM,US2S0C,KAAd,CT3SvC,C;YAH2C,OAGR,6BAAM,U;eACzC,4C;YAJ2C,OSkWL,UT9Vd,uB
AAF,CAAE,CS8Vc,C;;YTlWK,OSUY,UAAW,CAwV5B,UT3VF,uBAA3B,IAAI,oBAAuB,CS2VE,CAxV4B,MA
AK,KAAW,CTHzB,gCSGyB,MAAX,CAAhB,C;;;;K;ITC3D,yB;MAC4C,QAAC,CAAqB,GAAf,UAAP,IAAmC,C
AAC,MAAO,EAAW,IAAJ,EAAf,IAAgC,C;K;IAE/G,0B;MAC8C,OAAC,qBAAO,EAAP,CAAW,WAAZ,GAAyB
,IAAzB,GAAiC,YAAjC,W;K;IAG9C,0B;MAA8C,uBAAc,CAAd,EAAiB,EAAjB,C;K;IAE9C,kC;MACI,IAAI,gB
AAK,CAAT,C;QAAY,OAAS,WAAF,CAAE,EAAS,IAAT,C;MAErB,eAAiB,qBAAO,CAAP,CAAD,yBAAa,IAAb
,EAAD,WAAwB,CAAxB,C;MACf,UAAU,WAAI,sCAAW,IAAX,EAAJ,C;MACV,IAAI,kBAAO,IAAX,C;QACI,
uCAAO,IAAP,E;QACA,4CAAY,CAAZ,E;;MAEJ,OAAgB,WAAT,QAAS,EAAS,IAAT,CAAT,GAA8B,WAAJ,G
AAI,EAAS,IAAT,C;K;IiO1FzC,qC;K;;;
;;
;;
;;

;;;
;;;
;;;
;;;
;;;
;;;
;;;
;;;
;;;
;;;
;;;
;;Y5IsEA,iB;;K;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;
;;;
;;
;;
;;;
;;
;;;
;;S6IjE8B,iB;QAGD,iB;gCxJmBqB,I;IW8C9C,aAAyB,OAAO,OAAQ,KAAI,W
AAY,IAAG,OAAO,SAA5C,IAAyD,CAAC,CAAC,OAAO,SAAS,K;aAC7F,MAAJ,GAAY,iBAAgB,OAAH,OAA
b,CAAZ,GAAoD,gC;uDG1EJ,mC,EAAuB,wB;yB2I0BnC,iJ;8BAMK,2M;8BAKA,qQ;e5HsXS,MAAW,KG1ZlC,
GH0ZkC,EG1Z1B,GH0Z0B,C;eAAX,MAAW,KGzZlC,GHyZkC,EGzZ1B,GHyZ0B,C;sBWrVnC,gBAA8B,CAA9
B,O;oCjG2Pc,eAAmB,oCAAnB,C;yB4G5LL,C;gCAKY,eAAiB,OAAL,GAAK,CAAjB,EAAgC,OAAL,GAAK,C
AAhC,EAA+C,OAAL,GAAK,CAA/C,E;uBAwDd,K;qB4C9GK,C;yBACI,C;sBACH,C;kBACJ,C;iBACD,C;mBA
CE,C;sB1Ks0CH,O;;;;uB6C/0CzB,WsKmCkC,yBtKnClC,C;;;;CpEjFX,E;"}

Found
 in path(s):
* /opt/cola/permits/1830512329_1698119948.787533/0/kotlin-1-8-10-3-tgz/package/kotlin.js.map
No license file was found, but licenses were detected in source scan.

// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//   http://www.apache.org/licenses/LICENSE-2.0
// distributed under the License is distributed on an "AS-IS" BASIS,

Found in path(s):
* /opt/cola/permits/1830512329_1698119948.787533/0/kotlin-1-8-10-3-tgz/package/kotlin.js

1.27 lifecycle-extensions 2.6.1

1.27.1 Available under license :

List of 3rd party licenses:
----
* javapoet.jar (com.squareup:javapoet:1.8.0)

****** LICENSE:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management
of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written
communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense,
and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You

meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution,
then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying
the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks,
service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by
applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole
responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache
License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---

* kotlin-stdlib.jar (org.jetbrains.kotlin:kotlin-stdlib:1.1.1)

****** NOTICE:

```
==========================================================================
==  NOTICE file corresponding to the section 4 d of                     ==
==  the Apache License, Version 2.0,                                    ==
==  in this case for the Kotlin Compiler distribution.                  ==
==========================================================================
```

Kotlin Compiler
Copyright 2010-2015 JetBrains s.r.o and respective authors and developers

****** LICENSE:

```
/*
* Copyright 2010-2017 JetBrains s.r.o.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
```

* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

****** LICENSE:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION,
AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form
for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not
include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is
conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make,
have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part
of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise
complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its
Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but
not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to
apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES
OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---

* auto-common.jar (com.google.auto:auto-common:0.6)

****** LICENSE:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control
with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under
the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to
submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do
not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation
of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations
and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright
notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---

* annotations.jar (org.jetbrains:annotations:13.0)

****** LICENSE:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS
FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean
the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative
Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication
that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent
license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet
the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of
the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor
provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---

* guava.jar (com.google.guava:guava:18.0)

****** LICENSE:

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available
under the License, as indicated by a
copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized
to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive,
no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit)
alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices
that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have
executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8.
Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability
obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page"

as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

```
/*
* Copyright (C) 2017 The Android Open Source Project
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*      http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

1.28 libusb 1.0.25-pre

1.28.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest
you first think carefully about whether
this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients
all the rights that we gave
you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of
any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits
its criteria of freedom. The Lesser General
Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use
the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without
limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's

complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based

on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or

linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables
containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable

source code for the Library including whatever
changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user
installs one, as long as the modified version is
interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot

distribute
so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people
have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public
License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by

the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively

convey the exclusion of warranty; and each file should have at least the
"copyright" line
and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should
also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!