

NOTE: SCENARIO IS WHAT THE PUZZLE TAKER SEES.

SCENARIO:

You are developing a secure library for compression/decompression operations. All methods in the library require that the size of the original file (uncompressed) can be determined and be at most 100MB, otherwise the library's methods throw an exception. The decompress method takes a String value as the path of a zip file, and a String value as the path of destination directory, and decompresses the zip file into the given directory. Consider the snippet of code below and answer the following questions, assuming that the code has all required permissions to execute and there is no IOException.

```
01  // OMITTED: Import whatever is needed.
02  public final class ZipUtils {
03      private static final int BUFFER = 0x1000; // 4KB
04      private static final int MAX_FILE_SIZE = 0x6400000; // 100MB
05
06      public static void decompress (String path, String directory)
07          throws IOException {
08          FileInputStream fis = new FileInputStream(path);
09          ZipInputStream zis = new ZipInputStream(fis);
10          try {
11              ZipEntry entry;
12              while ((entry = zis.getNextEntry()) != null) {
13                  if (entry.getSize() < 0)
14                      throw new IllegalStateException("Size of file cannot be
15 determined");
16                  if (entry.getSize() > MAX_FILE_SIZE)
17                      throw new IllegalStateException("File is too large.");
18
19                  // OMITTED: Store the "entry" in the "directory"
20              }
21          }
22          finally {
23              //OMITTED: close the streams
24          }
25      }
26  }
```

Questions:

1. What does the decompress method do when executed?
2. What will happen if the decompress method is called with a zip file containing an entry (named big.txt) larger than 100MB?
 - a. big.txt won't be decompressed and the method will throw an exception.
 - b. big.txt may or may not be decompressed.
 - c. big.txt will be decompressed.

- d. The program will crash after the invocation of the decompress method.
- e. None of the above

[Other statistical questions will be imported here while creating survey.]

NOTE: ANSWER IS TO BE SHOWN TO THE PUZZLE TAKER AT THE END OF THE SESSION.

ANSWER:

a

The file (big.txt) will not be decompressed, since we are checking if the `getSize` method is able to determine the size of the file by checking if the return value is greater than 0. The `getSize` method may return -1 if the size of the entry (file) is unknown. Thus, the above code throws an exception if the file size cannot be determined or if the size exceeds 100MB.

NOTE: THE REST OF THIS DOCUMENT CONTAINS EXTRA INFORMATION FOR THE PROJECT RESEARCHERS. IT IS NOT TO BE SHOWN TO THE PUZZLE TAKERS.

TAGS:

java, zipinputstream

CATEGORIES:

Blindspot - NO

Type - Compression

Number of distinct functions - 5

Number of total functions - 5

Blindspot function - NA

Function call omitted - NA

Blindspot type - NA

Number of Parameters in the blindspot function - NA

Cyclomatic complexity - 5

NAME:

ZipEntry class, `getSize` method - Returns the uncompressed size of the entry data, or -1 if not known.

DESCRIPTION:

Java provides the `java.util.zip` package for zip-compatible data compression. It provides classes that enable you to read, create, and modify ZIP and GZIP files formats.

`ZipInputStream` functions the same as `InputStream`, and `ZipEntry` is basically used to represent a Zip file entry.

REFERENCES:

1. [ZipEntry#getSize](#)
2. <https://goo.gl/hi29lb>