NAME AND LAB GROUP:

1. ☐3 points☐ Objects of the class `ObjectOutputStream` represent binary files. If an exception of the class `IOException` is thrown when trying to close an object of the class `ObjectOutputStream`, this means that the file is not closed properly.

   You have to write a Java method with the profile:

   ```
   public static boolean tryToClose( ObjectOutputStream os )
   ```

   for trying to close the object `os` of the class `ObjectOutputStream`. The method should catch the exceptions of the class `IOException` and return `true` if the file was closed with no error, `false` otherwise.

   Solution:

   ```java
   public static boolean tryToClose( ObjectOutputStream os )
   {
       try {
           os.close();
           return true;
       }
       catch( IOException e ) {
           return false;
       }
   }
   ```

2. ☐2 points☐ We need storing and loading objects of the class `QueueIntLinked` into/from binary files. For it we will use objects of the class `ObjectOutputStream` in order to manage binary files.

   You have to do the required changes in the header of the class or classes that you consider in order to be able to perform the writing and reading operations of objects of the class `QueueIntLinked`.

   Solution:

   For writing an object of the class `QueueIntLinked` into a binary file all their attributes must be serializables, no only the objects of the own class.

   ```java
   // Changes to the header of the class NodeInt
   import java.io.Serializable;
   public class NodeInt implements Serializable {

   // Changes in the header of the class QueueIntLinked
   import java.io.Serializable;
   public class QueueIntLinked implements Serializable {
   ```

3. 5 points We need to include a new operation in the class `Concordance` for getting the longest word in the text used for construction the concordance. If there are several words with the maximum length, then it will be considered as valid that the method returns either of them. If the concordance has no words the method must return an empty string.

Remember, the attributes of the class `Concordance` are the following:

```
private NodeCnc first;
private int     size;
private boolean sorted;
private String  delimiters;
```

and the attributes of the class `NodeCnc` are:

```
String          word;
QueueIntLinked numbersOfLine;
NodeCnc         next;
```

You have to add a new method into the class `Concordance` with the following profile:

public String largestWord()

Solution:

```
public String largestWord()
{
    NodeCnc temp = first;
    String result = "";
    while( temp != null ) {
        if ( temp.word.length() > result.length() ) result = temp.word;
        temp = aux.next;
    }
    return result;
}
```