Assignment 1

In this assignment, you will practice activities covering the concepts taken in the course so far, through two main tasks as follows:

Task 1

Below is one faulty program. It includes test inputs that result in failure. Answer the following questions about this program.

Program:
```
public int findLast (int[] x, int y) {
// Effects: if x==null throw NullPointerException
// else return the index of the last element in x that equals y
// If no such element exists, return -1
    for (int i =x.lenght-1; i > 0; i--)
    {
        if (x[i] == y)
        {
            return i;
        }
    }
    return -1;
}

// test x=[2,3,5] y=2
// expected output =0
```

a) Explain what is wrong with the given code. Describe the fault precisely by proposing a modification to the code.

b) If possible, give a test case that does not execute the fault. If not, briefly explain why not.

c) If possible, give a test case that executes the fault, but does not result in an error state. If not, briefly explain why not.

d) If possible give a test case that results in an error, but not a failure. If not, briefly explain why not. Hint: Don't forget about the program counter.

e) For the given test case, describe the first error state. Be sure to describe the complete state

Task 1 deliverables:

1. A report showing a detailed answer for each step from (a) to (e). This report can be a word document, or a set of compressed scanned pictures.

Task 2

Objectives

- Studying implementation of an open source project.
- Using Input space-partitioning & boundary-analysis techniques to design test cases.
- Learn unit testing and code coverage.

Software Testing – Winter 2021

Description

- In this task, you will study the open source code of "java-math-library".
- There is no JavaDoc for this code, but the code is commented in the same style, so you should read carefully and try to grasp as much as you can. Furthermore, you may refer to the JavaDocs of the parent classes from the Java SDK.
- You are to study the following classes (de.tilman_neumann.util.StringUtil and de.tilman_neumann.util. Multiset_HashMapImpl)
- Once you study those APIs implementations/documentations, you are required to use input space partitioning along with boundary values analysis to design unit tests for their methods.

Task 2 deliverables:

2. A report of all the test cases you have developed for the classes mentioned above, with full reasoning of the partitions, the boundary values, and the chosen coverage criteria with complete explanation. In the written report, you should discuss how you are designing the test cases. This report can be a word document, or a set of compressed scanned pictures.
3. A testing package, with a test class for each class mentioned above (please follow java coding/naming standards).

Things to consider for task 2

- You can use JUnit or TestNG or any other java unit testing library if you please.
- Failing reasons for each unit test case should specifically be one reason.
- You should add your test package/classes to the included project.
- Make the test design document as detailed as possible.

General Assignment Guidelines
- This assignment is in groups of two **at most**.
- All the deliverables from both tasks should be placed in one folder, and should be named as stud1ID-stud2ID-Assignment1.zip
- Failing to name the zip file as mentioned above, marks will be deducted.
- In case of cheating, a faculty-based regulation will be applied.

Due date and submission
Assignment 1 is due on Monday, May 17th 10th at 11:55 PM (Cairo Local time). Submission needs to be done through the course's Blackboard only.

Late submission policy: Late submissions is allowed with a penalty for two days only. A 50% late submission penalty will be applied after the original deadline.