**Midterm Revision**

**Course**: Data Structures & Algorithms

# **Formation**

* Students conduct the test individually in computers of the laboratory.
* Submit your work following the instruction of the supervisor.
* For each task, students are provided base source code files, then programming in these files.
* Students take your own responsibility for identifiers (class names, function names, file names, etc.) and the structure of the project folder in case you modify them differently from requirements.

# **Requirement**

Create a folder named **<Student ID>** (for example, 522K0001)

Copy the provided source code folders to the **<Student ID>** folder above

Conduct the following requirements

After completion, students delete bytecode files (.class), compress the project folder as

**<Student ID>.zip**, and then submit following the instruction.

## **Question 1 (4.0 points)**

|  |  |
| --- | --- |
| **<<interface>>**  **ListInterface** |  |
|  |
| + size(): int  + sumEven(): int  + countKey(k: int): int |

Students implements designated methods in MyLinkedList.java

* size(): return the number of nodes. If the list is empty, then return 0.
* sumEven(): return the summation of even keys in the list. If the list is empty or there are no even keys, then return 0.
* countKey(k): return the number of occurences of the given key **k**. If the list is empty, then return 0.

Notes

* Do not perform inputting and printing the designated methods.
* Students can add the **main** method to test your work; however, if it causes any compilation errors, then the whole question is scored 0.0 points.

## **Question 2 (3.0 points)**

Students are allowed to use

java.util.Stack

java.util.Queue

java.util.LinkedList

Implement the method below in Question2.java

public static String reverse(String str)

to reverse the given string str using a Stack.

For example,

* str = “abcde”
* returned value: “edcba”

Notes

* Do not perform inputting and printing the designated methods.
* Students can add the **main** method to test your work; however, if it causes any compilation errors, then the whole question is scored 0.0 points.

## **Question 3 (3.0 points)**

Implement the **recursive** method below in Question3.java

public static int sumDigits(int a)

to compute the summation of digits of a positive integer a.

For example,

* a = 2417
* returned value: 14 (2+4+1+7)

Notes

* Do not perform inputting and printing the designated methods.
* Students can add the **main** method to test your work; however, if it causes any compilation errors, then the whole question is scored 0.0 points.

**-- END --**