Programming II
Assignment 1
Due: 09.03.2023 by 9:00am in LMS Assignment 1 folder
Note: There are 8 questions in this assignment. Note that making a function static makes it possible to call that function from the main function without having to create an object to call that. For example,

```
public class A{
        public static void main(String [] argos){
                foo();
        }
        public static void foo(){
                System.out.println("this function can be called from main without having" +
                        "to create an object as can be seen");
        }
}
```

The above is a convenient way to test your functions/methods for beginners of Java.

1.

(*Check password*) Some websites impose certain rules for passwords. Write a method that checks whether a string is a valid password. Suppose the password rules are as follows:

- A password must have at least eight characters.
- A password consists of only letters and digits.
- A password must contain at least two digits.

Write a program that prompts the user to enter a password and displays `Valid Password` if the rules are followed or `Invalid Password` otherwise.

2.

(*Sum series*) Write a method to compute the following series:

$$m(i) = \frac{1}{2} + \frac{2}{3} + \ldots + \frac{i}{i+1}$$

Write a test program that displays the following table:

i	m(i)
1	0.5000
2	1.1667
...	
19	16.4023
20	17.3546

3.

(*Display an integer reversed*) Write a method with the following header to display an integer in reverse order:

```
public static void reverse(int number)
```

For example, `reverse(3456)` displays `6543`. Write a test program that prompts the user to enter an integer and displays its reversal.

4.

(*Sum the digits in an integer*) Write a method that computes the sum of the digits in an integer. Use the following method header:

```
public static int sumDigits(long n)
```

For example, `sumDigits(234)` returns `9` (2 + 3 + 4). (*Hint*: Use the `%` operator to extract digits, and the `/` operator to remove the extracted digit. For instance, to extract 4 from 234, use `234 % 10` (= 4). To remove 4 from 234, use `234 / 10` (= 23). Use a loop to repeatedly extract and remove the digit until all the digits are extracted. Write a test program that prompts the user to enter an integer and displays the sum of all its digits.

5.

(*Sort characters in a string*) Write a method that returns a sorted string using the following header:

```
public static String sort(String s)
```

For example, `sort("acb")` returns `abc`.

Write a test program that prompts the user to enter a string and displays the sorted string.

6.

(*Partition of a list*) Write the following method that partitions the list using the first element, called a *pivot*.

```
public static int partition(int[] list)
```

After the partition, the elements in the list are rearranged so that all the elements before the pivot are less than or equal to the pivot and the elements after the pivot are greater than the pivot. The method returns the index where the pivot is located in the new list. For example, suppose the list is {5, 2, 9, 3, 6, 8}. After the partition, the list becomes {3, 2, 5, 9, 6, 8}. Implement the method in a way that takes at most `list.length` comparisons. Write a test program that prompts the user to enter a list and displays the list after the partition. Here is a sample run. Note that the first number in the input indicates the number of the elements in the list. This number is not part of the list.

7.

(*Pattern recognition: consecutive four equal numbers*) Write the following method that tests whether the array has four consecutive numbers with the same value.

```
public static boolean isConsecutiveFour(int[] values)
```

Write a test program that prompts the user to enter a series of integers and displays if the series contains four consecutive numbers with the same value. Your program should first prompt the user to enter the input size—i.e., the number of values in the series. Here are sample runs:

```
Enter the number of values: 8 Enter
Enter the values: 3 4 5 5 5 5 4 5 Enter
The list has consecutive fours
```

```
Enter the number of values: 9 Enter
Enter the values: 3 4 5 5 6 5 5 4 5 Enter
The list has no consecutive fours
```

8.

(*Sorted?*) Write the following method that returns true if the list is already sorted in increasing order.

```
public static boolean isSorted(int[] list)
```

Write a test program that prompts the user to enter a list and displays whether the list is sorted or not. Here is a sample run. Note that the first number in the input indicates the number of the elements in the list. This number is not part of the list.

```
Enter list: 8 10 1 5 16 61 9 11 1 Enter
The list is not sorted
```

```
Enter list: 10 1 1 3 4 4 5 7 9 11 21 Enter
The list is already sorted
```

Deliverables:

1. One pdf file that contains the source code for all the questions. **Make sure you comment your code. That is, explain important pieces of code or lines.** Please mark your programs in it with the proper numbers. The first page of the pdf file should have a cover page for the answers with your full name and student number. Also mark the question numbers given below for which you implemented the programs for.

   1 2 3 4 5 6 7 8

   on your cover page.

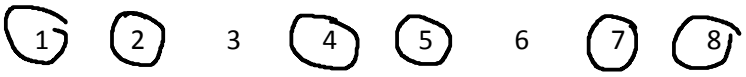
2. Record your explanation of your solutions to questions 7 and 8. Each recording should not last more than 5 minutes.
3. Zip the pdf file with two video recording of explanations for questions 7 and 8 and submit them to LMS (Assignment 1 folder).

A sample pdf file is given on the next page.

Programming II

Assignment 1

Ahmet Programci



1. Your implementation for Question 1

   ….

2. Your implementation for Question 2

   ….

   and so forth.