Programming Fundamentals I – Laboratory Exam

Ordinary Period 2022/23 – 16/01/2023

Duration: 60 minutes

Implement a program that, given a matrix of integers in the range [0, 9], checks and displays the coincidences in the value of the sum of elements in a row with the value of the sum of elements in a column.

The program must show all the cases in which the above occurs, indicating:

- The row number and the column number (taking into account that the rows and columns start at 0) where the sum match occurs along with the value of the sum shared by the row and column number
- The values of the numbers in the row
- The values of the numbers in the column

If there is no match, the program will show a message indicating this.

NOTE: You must use as a basis the code of the java file available at Virtual Campus (at the beginning of the Laboratory section), in which there are several methods implemented to create and show the matrix. You will have to:

- Implement the necessary methods for the program to carry out the functionality indicated above.
- Finish implementing the main method to make the calls to the corresponding methods.

Examples of execution:

01

```
Introduce the number of rows for the matrix [2, 6]: 3
Introduce the number of columns for the matrix [2, 6]: 4

1 5 5 9
5 4 9 1
5 8 6 9

The row 0 and the column 2 have a matching sum: 20
The values of the row 0 are: 1 5 5 9
The values of the column 2 are: 5 9 6

The row 1 and the column 3 have a matching sum: 19
The values of the row 1 are: 5 4 9 1
The values of the column 3 are: 9 1 9
```

02

```
Introduce the number of rows for the matrix [2, 6]: 3
Introduce the number of columns for the matrix [2, 6]: 4

3 0 2 0
9 3 9 3
6 6 5 4

There are no matchings of sums between rows and columns
```

Evaluation:

- To pass the laboratory exam it is mandatory that there are no compilation errors in your solution.
- The program must work correctly, according to the specification above.
- The program must be correctly modularized and parameterized, dividing its functionality into the auxiliary methods considered necessary.
- Global variables (defined at class level) are not allowed, except for the Scanner used to read from the keyboard.
- The code must be easily readable (name of variables and representative modes, correct indentation and spacing format, etc.).

Delivery:

- The **.java file must be uploaded** to the corresponding upload task in Campus Virtual before 20:15.
- You must **use the following name** for the file: LabGroup_1stSurname_2ndSurname_Name and use the same name as the name of the class. For example, the file would be: B1_Garcia_Lopez_Ana.java