SESSION 6: MODULARIZATION (I)

GOAL:

- Practice the concepts studied in theory about modular programming (organizing the code into methods).

EXERCISE:

Implement a modularized program containing methods to perform each of the following tasks:

- Read from the keyboard an integer value within the range [*LB*, *UB*]. The values of *LB* and *UB* will be the input parameters of the method, as well as a String with the message that will be prompted to the user to ask for the value. The return value will be the integer read from keyboard, once it has been assured that it is valid.
- Read from the keyboard an integer value greater or equal than another value *LB*. The input parameters of the method are the value of *LB* and a String with the message that will be prompted to the user when asking for the value. The return value will be the integer read from keyboard, once it has been assured that it is valid.
- Assign the value 1 to *X* random positions in a matrix (2D array) of integer numbers. The value of *X* and the matrix will be the only parameters for the method. The method will not return anything, but it will assign the value 1 to *X* positions in the array passed as a parameter. This assignment shall be done randomly, ensuring that *X* distinct positions are assigned the value 1 always.
- Count the number of positions in a matrix of integers with the value 0. The matrix will be the only parameter for the method. The method will return the number of elements in the array that have the value 0.
- Calculate the number of elements that have value 1 surrounding a position in a matrix of integers. The parameters for the method Will be the array where the amount has to be calculated and the position around which the method has to calculate the amount of values 1. This position will be given by two integers *R* and *C*, representing the row and column for the position to check. The method will return the amount of times the value 1 appears in the elements of the matrix that immediately surround the position given by *R* and *C.*
- Print on the standard output the content of a matrix of integers passed as a parameter. The matrix to be shown will be the only parameter in the method.

The comparisons and assignments with the values 0 and 1 must be carried out by declaring these values as constants and using them in the corresponding methods.

Additionally, the program will include the code necessary to test the correct operation of each of the methods created. The values to be used as parameters in the method calls may have been obtained previously as a result of the call to another method. An example is shown below:

EXAMPLE OF EXECUTION:

```
Test for the method that reads a value within a range:
Enter a number in the range [2, 10]: 3
The number read was 3
===================================

Test for the method that reads a value greater than or equal to other
integer given:
Enter a number (greater than or equal to 2): 6
The number read was 6
===================================

Test for the method that assign the value 1 randomly and the method
that shows the content of a matrix:
1 1 1
1 0 0
===================================

Test for the method that counts the number of zeros in a matrix:
The number of zeros is 2
===================================

Test for the method that counts the number of ones surrounding a
position in a matrix:
The amount of ones surrounding the position [1][2] is: 2

End of the program
```