**Exercise 1**

The aim of this exercise is to display the following diagram:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A |  |  |  |  |  |
| B | B |  |  |  |  |
| A | A | A |  |  |  |
| B | B | B | B |  |  |
| A | A | A | A | A |  |
| B | B | B | B | B | B |

To do this:

1. Implement the class A that has two static and public variables *counter* and *size*. The variable *size* is initialized with a positive value of your choice (for example, the *size* of the diagram above is 6). The constructor of the class A does the following:
   * Increments by 1 the variable *counter.*
   * Displays a row of the diagram with the letter ‘A’ according to the value of the *counter*.
     + For example: if *counter* = 3 it displays: A A A
   * Creates an object of the class B **if** the variable *counter* is < to the variable *size*.
2. Implement the class B. The constructor of the class B does the following:
   * Increments by 1 the variable *counter* of the class A*,*
   * Displays a row of the diagram with the letter ‘B’ according to the value of the *counter* of the class A.
     + For example: if *counter* = 4 it displays: B B B B
   * Creates an object of the class A **if** the variable *counter* of the class A is < to the variable *size* of the class A.
3. Write a method “main” in the class A that allows, by only one instruction, the construction of the requested diagram.

***Exercise 2.***

Write a method permitting to insert an integer value in an array of integers and shift all the next elements. The method signature is the following:

***public static bool insert (int val, int pos, int [] T)***

**val:** is the integer value to insert

**pos:** is the position where we will insert the value val.

**T:** is the array of integers in which we will insert the value val.

Remark: the method ‘***insert’*** returns ‘true’ if the value has been inserted in the array and returns ‘false’ otherwise (example the position exceeds the array size).

Example of insertion and shifting:

Initially the array T :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | 7 | 14 | 8 | 3 | 11 |

After insertion and shifting, val= 6, position=2:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | 7 | 6 | 14 | 8 | 3 |

***Exercise 3.***

Create in C# a class ‘**Circle’** containing 3 private instance variables (coordinates x and y of the center and the radius), the get and set methods and a constructor to initialize them, and override the method ToString.

Define 2 methods in the class ‘**Circle’** to return the perimeter and the area of the circle.

In a class **Program** which contains a method Main, create two instances c1 and c2 of type circle, the information about the circles is given by the user, then calculate and display their coordinates, radius, areas and perimeters.

**N.B :** The constant is defined by the static variable Math.PI.