Introduction to C++ programming
Exam – Part 2 (version A)

Exercise 1. (7 punti)
Write a C++ function that displays on the screen the following drawing:

```
xxxxxxx
xx   xx
x x x x
x  x  x
x x x x
xx   xx
xxxxxxx
```

Observe that the number of lines in the drawing is odd. The function should be capable of displaying any such drawing consisting of any odd number of lines. Write a simple test program that ask the user to enter the number of lines for the drawing to be displayed and then displays the corresponding drawing.

Exercise 2. Define a C++ class called **`PointVector`** for representing vectors of points in the bi-dimensional Euclidian space. A vector of points can be represented as an array of real numbers containing an even number of elements. For example the array **`v=`**{**`1.5,1,4.5,1,4.5,5`**} represents 3 points with coordinates **`(1.5,1)`**, **`(4.5,1)`** and **`(4.5,5)`**. A user of the **`PointVector`** class should be able to:

- create vectors of points of arbitrary even size n
- display a vector through the **`cout`** command and with the following format:

  `vector=[(1.5, 1), (4.5,1), (4.5,5)]`

- display the point with max x-coordinate
- display the distance between each pair of points in the vector with the following manner

  ```
  point 1 - point 2: 3.0
  point 1 - point 3: 5.0
  point 2 - point 3: 4.0
  ```

  assuming a vector of points **`v=`**{**`1.5,1,4.5,1,4.5,5`**}.

- to determine the sub-vector of points with increasing x-coordinate and return the sum of their x-coordinate. For example for vector

  `v=[(2.5, 1), (1.5,1), (4.5,5), (5,5), (1,2)]`

  the largest sub-vector of elements with increasing x-coordinate contained in **`v`** is **`[(1.5,1), (4.5,5), (5,5)]`** and its sum is $1.5 + 4.5 + 5 = 11$.

Write a simple **`main`** to test the **`PointVector`** class by creating a **`PointVector`** object and display it on the screen, display yhe distance for each pair of points, and display the largest subvector of points with increasing x-coordinates.