Polygon Bounding Box

Description

A polygon is convex if all of its internal angles are less than 180° (and none of the edges cross each other). Figure 1 shows an example.

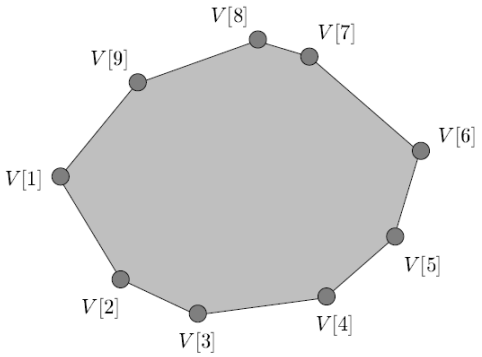

Figure 1 Convex Polygon Example

We represent a convex polygon as an array V[1…N] where each element of the array represents a vertex of the polygon in the form of a coordinate pair (x,y).

We are told that the first vertex in the array V[] is the vertex with the minimum x-coordinate and that the vertices are ordered counterclockwise, as in the figure. You may also assume that the x-coordinates of the vertices are all distinct, as are the y-coordinates of the vertices.

Design the fastest algorithm **to find the bounding box of such polygon in an efficient way** (i.e. **minX, maxX, minY, maxY**).

Complexity

Complexity of your algorithm Should be **less than O(N)**

Input: Already Implemented

The first line of input is an integer T (T < =10), that indicates the number of test cases. Each case consists of one integer that represents the number of points for a convex polygon (N), followed by the points coordinates starting from the leftmost side point. N is an integer in the range 100,000 to 1,000,000.

Output: Already Implemented

The result of the computation: **minX, maxX, minY & maxY** in one line **respectively** separated by spaces.

Implemented Struct:

1. **BoundingBox:** is the data type that shall be returned from BoundingBoxOfConvexPolygon after deducting the minX, maxX, minY, maxY.

```
struct BoundingBox
 {
     public double minX;
     public double minY;
     public double maxX;
     public double maxY;
 }
```

2. **Point:** is the data type used to carry each point coordinates (X and Y).

```
 struct Point
 {
     public double X;
     public double Y;
 }
```

Function: **Implement it!**

```
static BoundingBox RequiredFunction(Point [] Points, int N)
```

It takes an array Points of N distinct coordinates sorted anticlockwise starting from the leftmost point and the array size (N). It should return the minX, maxX, minY, maxY through the **BoundingBox** Struct.

`PROBLEM_CLASS.cs includes this method.`

Test Cases

#	Input: Convex Polygon Points	Output: Bounding Box Points
1	4.5 1 5.5 1.5 7 3 5 2	4.5 7 1 3
2	11 6 12 5 13 5.5 14 5.8 13.5 6.5 12.5 7	11 14 5 7

C# Help

Creating 1D array

```
int [] array1D = new int [size]
```

Creating 2D array

```
int [,] array2D = new int [size1, size2]
```

Getting the size of 1D array

```
int size = array1D.GetLength(0);
```

Getting the size of 2D array

```
int size1 = array2D.GetLength(0);

int size2 = array2D.GetLength(1);
```

Sorting single array

Sort the given array "items" in ascending order

```
Array.Sort(items);
```

Sorting parallel arrays

Sort the first array "master" and re-order the 2nd array "slave" according to this sorting

```
Array.Sort(master, slave);
```