Given a list of integers, find the longest increasing contiguous sequence in it. If there are several answers, return the first one.

**Example:**

LargestSequence([2,3,6,7,5,3,2,1]) = [2,3,6,7]

* **[input] array.integer data**
* **[output] array.integer**
  + The list containing the longest increasing sequence.

<https://codefights.com/challenge/u2M3LtCH536HeaZoo>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int[] LargestSequence(int[] data)

{

List<List<int>> ls = new List<List<int>>();

List<int> largest = new List<int>();

for (int i = 0; i < data.Length; i++)

{

for (int j = i; j < data.Length; j++)

{

List<int> actual = new List<int>();

for (int k = i; k <= j; k++)

{

actual.Add(data[k]);

}

bool esInc = true;

for (int k = 0; k + 1 < actual.Count; k++)

{

if (actual[k + 1] < actual[k])

{

esInc = false;

break;

}

}

if (esInc && actual.Count > largest.Count)

{

//copio actual en largest

largest = new List<int>();

for (int k = 0; k < actual.Count; k++)

{

largest.Add(actual[k]);

}

}

if (esInc )

{

ls.Add(actual);

}

}

}

//foreach (List<int> lista in ls)

//{

// for (int i = 0; i < lista.Count; i++)

// {

// Console.Write(lista[i] + " ");

// }

// Console.WriteLine();

//}

return largest.ToArray();

}

static void Main(string[] args)

{

int[] arr = { 2, 3, 6, 7, 5, 3, 2, 1 };

// int[] arr = { 1, 2, 3, 4, 5,6,7,8,9 };

int[] res = LargestSequence(arr);

for (int i = 0; i < res.Length; i++)

{

Console.Write(res[i] + " ");

}

Console.ReadLine();

}

}

}