Some people are standing in a row in a park. There are trees between them which cannot be moved. Your task is to rearrange the people by their heights in a non-descending order without moving the trees. People can be very tall!

Example

For a = [-1, 150, 190, 170, -1, -1, 160, 180], the output should be  
sortByHeight(a) = [-1, 150, 160, 170, -1, -1, 180, 190].

Input/Output

* **[execution time limit] 3 seconds (cs)**
* **[input] array.integer a**

If a[i] = -1, then the ith position is occupied by a tree. Otherwise a[i] is the height of a person standing in the ith position.

*Guaranteed constraints:*  
1 ≤ a.length ≤ 1000,  
-1 ≤ a[i] ≤ 1000.

* **[output] array.integer**
  + Sorted array a with all the trees untouched.

**[C#] Syntax Tips**

// Prints help message to the console

// Returns a string

**string** **helloWorld**(**string** name) {

Console.Write("This prints to the console when you Run Tests");

**return** "Hello, " + name;

}

<https://app.codesignal.com/arcade/code-arcade/sorting-outpost/D6qmdBL2NYz49XHwM/description>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

class Principal

{

//static int[][] starRotation(int[][] matrix, int width, int[] center, int t)

//{

//}

static int[] sortByHeight(int[] a)

{

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

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

{

if (a[i] != -1)

{

alturas.Add(a[i]);

}

}

alturas.Sort();

int indice = 0;

int[] ans = new int[a.Length];

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

{

if(a[i] != -1)

{

ans[i] = alturas[indice++];

}

else

{

ans[i] = -1;

}

}

return ans;

}

public static void Main(string[] args)

{

int[] a = { -1, 150, 190, 170, -1, -1, 160, 180 };

int[] ans = sortByHeight(a);

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

{

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

}

Console.ReadLine();

}

}