Given an array of strings, sort them in the order of increasing lengths. If two strings have the same length, their relative order must be the same as in the initial array.

Example

For

inputArray = ["abc",

"",

"aaa",

"a",

"zz"]

the output should be

sortByLength(inputArray) = ["",

"a",

"zz",

"abc",

"aaa"]

Input/Output

* **[execution time limit] 3 seconds (cs)**
* **[input] array.string inputArray**

A non-empty array of strings.

*Guaranteed constraints:*  
3 ≤ inputArray.length ≤ 100,  
0 ≤ inputArray[i].length ≤ 100.

* **[output] array.string**

**[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/QQB7f8ouAqY6jf7xi/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 string[] sortByLength(string[] inputArray)

{

for (int i = 1; i < inputArray.Length; i++)

{

int indice = i;

while (indice > 0 && inputArray[indice - 1].Length > inputArray[indice].Length)

{

string temp = inputArray[indice - 1];

inputArray[indice - 1] = inputArray[indice];

inputArray[indice] = temp;

indice--;

}

}

return inputArray;

}

public static void Main(string[] args)

{

string[] inputArray = {"abc",

"",

"aaa",

"a",

"zz" };

//["",

//"a",

//"zz",

//"abc",

//"aaa"]

string[] s = sortByLength(inputArray);

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

{

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

}

Console.ReadLine();

}

}