**7 kyu**

**Sort array by string length**

1052587% of1,037523 of6,369[stevehopkinson](https://www.codewars.com/users/stevehopkinson)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/sort-array-by-string-length/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/sort-array-by-string-length/solutions/csharp)

[Forks (2)](https://www.codewars.com/kata/sort-array-by-string-length/forks/csharp)

[Discourse (15)](https://www.codewars.com/kata/sort-array-by-string-length/discuss/csharp)

* Add to Collection
* |
* Share this kata:

Write a function that takes an array of strings as an argument and returns a sorted array containing the same strings, ordered from shortest to longest.

For example, if this array were passed as an argument:

["Telescopes", "Glasses", "Eyes", "Monocles"]

Your function would return the following array:

["Eyes", "Glasses", "Monocles", "Telescopes"]

All of the strings in the array passed to your function will be different lengths, so you will not have to decide how to order multiple strings of the same length.

<https://www.codewars.com/kata/sort-array-by-string-length/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static string[] SortByLength(string[] array)

{

for (int i = 0; i < array.Length - 1; i++)

{

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

{

if (array[i].Length > array[j].Length)

{

string temp = array[i];

array[i] = array[j];

array[j] = temp;

}

}

}

return array;

}

static void Main(string[] args)

{

string[] s = { "Beg", "Life", "I", "To" };

foreach(string item in SortByLength(s))

{

Console.Write(item + " ");

}

Console.ReadLine();

}

}

}