**7 kyu**

**Ordered Count of Characters**

361392% of 22741 of1,182[suic](https://www.codewars.com/users/suic" \o "This kata's Sensei)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/ordered-count-of-characters/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/ordered-count-of-characters/solutions/csharp)

[Forks (5)](https://www.codewars.com/kata/ordered-count-of-characters/forks/csharp)

[Discourse (31)](https://www.codewars.com/kata/ordered-count-of-characters/discuss/csharp)

* Add to Collection
* |
* Share this kata:

Count the number of occurrences of each character and return it as a list of tuples in order of appearance.

Example:

Kata.orderedCount("abracadabra") == new List<Tuple<char, int>> () { new Tuple<char, int>('a', 5), new Tuple<char, int>('b', 2), new Tuple<char, int>('r', 2), new Tuple<char, int>('c', 1), new Tuple<char, int>('d', 1)}

<https://www.codewars.com/kata/ordered-count-of-characters/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static List<Tuple<char, int>> OrderedCount(string input)

{

// Implement me!

Dictionary<char, int> dic =

new Dictionary<char, int>();

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

{

if (dic.ContainsKey(input[i]))

{

dic[input[i]]++;

}

else

{

dic[input[i]] = 1;

}

}

List<Tuple<char, int>> lista =

new List<Tuple<char, int>>();

foreach (KeyValuePair<char, int> kvp in dic)

{

if (kvp.Value > 0)

{

lista.Add(new Tuple<char, int>(kvp.Key, kvp.Value));

}

}

return lista;

}

static void Main(string[] args)

{

List<Tuple<char, int>> lista = OrderedCount("abracadabra");

foreach(Tuple<char, int> item in lista)

{

Console.WriteLine(item.Item1 + " " + item.Item2);

}

Console.ReadLine();

}

}

}