**7 kyu**

**Guess the Word: Count Matching Letters**

7387% of 12191 of573[GrahamD](https://www.codewars.com/users/GrahamD)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/guess-the-word-count-matching-letters/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/guess-the-word-count-matching-letters/solutions/csharp)

[Discourse (16)](https://www.codewars.com/kata/guess-the-word-count-matching-letters/discuss/csharp)

* Add to Collection
* |
* Share this kata:

Consider a game, wherein the player has to guess a target word. All the player knows is the length of the target word.

To help them in their goal, the game will accept guesses, and return the number of letters that are in the correct position.

Write a method that, given the correct word and the player's guess, returns this number.

For example, here's a possible thought process for someone trying to guess the word "dog":

CountCorrectCharacters("dog", "car"); //0 (No letters are in the correct position)

CountCorrectCharacters("dog", "god"); //1 ("o")

CountCorrectCharacters("dog", "cog"); //2 ("o" and "g")

CountCorrectCharacters("dog", "cod"); //1 ("o")

CountCorrectCharacters("dog", "bog"); //2 ("o" and "g")

CountCorrectCharacters("dog", "dog"); //3 (Correct!)

The caller should ensure that the guessed word is always the same length as the correct word, but since it could cause problems if this were not the case, you need to check for this eventuality:

//Throw an InvalidOperationException if the two parameters are of different lengths.

You may assume, however, that the two parameters will always be in the same case.

<https://www.codewars.com/kata/guess-the-word-count-matching-letters/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public int CountCorrectCharacters(string correctWord, string guess)

{

if(correctWord.Length != guess.Length)

{

throw new InvalidOperationException("error");

}

int n = correctWord.Length;

int cont = 0;

for (int i = 0; i < n; i++)

{

if(correctWord[i] == guess[i])

{

cont++;

}

}

return cont;

}

static void Main(string[] args)

{

Console.ReadLine();

}

}

}