**7 kyu**

**Common Substrings**

37795% of 7086 of703[Shivo](https://www.codewars.com/users/Shivo)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/common-substrings/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/common-substrings/solutions/csharp)

[Forks (3)](https://www.codewars.com/kata/common-substrings/forks/csharp)

[Discourse (45)](https://www.codewars.com/kata/common-substrings/discuss/csharp)

* Add to Collection
* |
* Share this kata:

Given 2 strings, your job is to find out if there is a substring that appears in both strings. You will return true if you find a substring that appears in both strings, or false if you do not. We only care about substrings that are longer than one letter long.

#Examples:

\*Example 1\*

SubstringTest("Something","Fun"); //Returns false

\*Example 2\*

SubstringTest("Something","Home"); //Returns true

In the above example, example 2 returns true because both of the inputs contain the substring "me". (so**ME**thing and ho**ME**)  
In example 1, the method will return false because something and fun contain no common substrings. (We do not count the 'n' as a substring in this Kata because it is only 1 character long)

#Rules: Lowercase and uppercase letters are the same. So 'A' == 'a'.  
We only count substrings that are > 1 in length.

#Input: Two strings with both lower and upper cases.

#Output: A boolean value determining if there is a common substing between the two inputs.

<https://www.codewars.com/kata/common-substrings/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp2

{

class Program

{

//static int LCSubStr(string X, string Y,

// int m, int n)

//{

// // Create a table to store lengths of

// // longest common suffixes of substrings.

// // Note that LCSuff[i][j] contains length

// // of longest common suffix of X[0..i-1]

// // and Y[0..j-1]. The first row and first

// // column entries have no logical meaning,

// // they are used only for simplicity of

// // program

// int[,] LCStuff = new int[m + 1, n + 1];

// // To store length of the longest common

// // substring

// int result = 0;

// // Following steps build LCSuff[m+1][n+1]

// // in bottom up fashion

// for (int i = 0; i <= m; i++)

// {

// for (int j = 0; j <= n; j++)

// {

// if (i == 0 || j == 0)

// LCStuff[i, j] = 0;

// else if (X[i - 1] == Y[j - 1])

// {

// LCStuff[i, j] =

// LCStuff[i - 1, j - 1] + 1;

// result = Math.Max(result,

// LCStuff[i, j]);

// if (result > 1) return true;

// }

// else

// LCStuff[i, j] = 0;

// }

// }

// //return result;

// return false;

//}

public static bool SubstringTest(string str1, string str2)

{

string X = str1.ToLower();

string Y = str2.ToLower();

int m = X.Length;

int n = Y.Length;

int[,] LCStuff = new int[m + 1, n + 1];

// To store length of the longest common

// substring

int result = 0;

// Following steps build LCSuff[m+1][n+1]

// in bottom up fashion

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

{

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

{

if (i == 0 || j == 0)

LCStuff[i, j] = 0;

else if (X[i - 1] == Y[j - 1])

{

LCStuff[i, j] =

LCStuff[i - 1, j - 1] + 1;

result = Math.Max(result,

LCStuff[i, j]);

if (result > 1) return true;

}

else

LCStuff[i, j] = 0;

}

}

//return result;

return false;

}

static void Main(string[] args)

{

String X = "OldSite:GeeksforGeeks.org";

String Y = "NewSite:GeeksQuiz.com";

int m = X.Length;

int n = Y.Length;

Console.WriteLine(SubstringTest(X, Y));

Console.ReadLine();

}

}

}