A *ciphertext alphabet* is obtained from the [plaintext alphabet](keyword://plaintext-alphabet) by means of rearranging some characters. For example "bacdef...xyz" will be a simple ciphertext alphabet where a and b are rearranged.

A *substitution cipher* is a method of encoding where each letter of the *plaintext alphabet* is replaced with the corresponding (i.e. having the same index) letter of some *ciphertext alphabet*.

Given two strings, check whether it is possible to obtain them from each other using some (possibly, different) *substitution ciphers*.

**Example**

* For string1 = "aacb" and string2 = "aabc", the output should be  
  isSubstitutionCipher(string1, string2) = true.

Any *ciphertext alphabet* that starts with acb... would make this transformation possible.

* For string1 = "aa" and string2 = "bc", the output should be  
  isSubstitutionCipher(string1, string2) = false.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] string string1**

A string consisting of lowercase characters.

*Constraints:*  
1 ≤ string1.length ≤ 10.

* **[input] string string2**

A string consisting of lowercase characters of the same length as string1.

*Constraints:*  
string2.length = strin1.length.

* **[output] boolean**

<https://codefights.com/arcade/code-arcade/mirror-lake/rNrF4v5etMdFNKD3s>

------------SOLUCION MAS SIMPLE--------------

static bool isSubstitutionCipher(string string1, string string2)

{

int n = string1.Length;

Dictionary<char, char> diccio = new Dictionary<char, char>();

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

{

if (diccio.ContainsKey(string1[i]))

{

if (string2[i] != diccio[string1[i]])

return false;

}

else

{

/\*

si no contiene el key pero YA CONTIENE el value

es porque ya otro key, lo tiene, por lo tanto

a un string1[i] no le corresponde un solo string2[i] y retorna falso

Ej si:

string string1= "aaxyaa";

string string2 = "aazzaa";

primero la asocia a 'x' la 'z'

luego comprueba a 'y' de string1

pero como ya contiene ese value (asociado a 'x')

entonces retorna falso

\*/

if (diccio.ContainsValue(string2[i]))

{

return false;

}

diccio[string1[i]] = string2[i];

}

}

return true;

}

----OTRA SOLUCION---

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static bool isSubstitutionCipher(string string1, string string2)

{

int n = string1.Length;

Dictionary<char, char> diccio = new Dictionary<char, char>();

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

if (diccio.ContainsKey(string1[i]))

{

if (string2[i] != diccio[string1[i]])

return false;

}

else

diccio[string1[i]] = string2[i];

Dictionary<char, char> diccio2 = new Dictionary<char, char>();

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

if (diccio2.ContainsKey(string2[i]))

{

if (string1[i] != diccio2[string2[i]])

return false;

}

else

diccio2[string2[i]] = string1[i];

return true;

}

static void Main(string[] args)

{

//string string1 = "aabc";

//string string2 = "aacb";

string string1 = "aaxyaa";

string string2 = "aazzaa";

Console.WriteLine(isSubstitutionCipher(string1, string2));

Console.ReadLine();

}

}

}