**7 kyu**

**Unique string characters**

12991% of 13549 of570[KenKamau](https://www.codewars.com/users/KenKamau)

C#

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

Details

[Solutions](https://www.codewars.com/kata/unique-string-characters/solutions/csharp)

[Forks (2)](https://www.codewars.com/kata/unique-string-characters/forks/csharp)

[Discourse (6)](https://www.codewars.com/kata/unique-string-characters/discuss/csharp)

* Add to Collection
* |
* Share this kata:

In this Kata, you will be given two strings a and b and your task will be to return the characters that are not common in the two strings.

For example:

solve("xyab","xzca") = "ybzc"

--The first string has 'yb' which is not in the second string.

--The second string has 'zc' which is not in the first string.

Notice also that you return the characters from the first string concatenated with those from the second string.

More examples in the tests cases.

Good luck!

Please also try [Simple remove duplicates](https://www.codewars.com/kata/5ba38ba180824a86850000f7)

<https://www.codewars.com/kata/unique-string-characters/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

//static int[] solve(int[] arr)

//{

// Dictionary<int, int> frec = new Dictionary<int, int>();

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

// {

// if (frec.ContainsKey(arr[i]))

// {

// frec[arr[i]]++;

// }

// else

// {

// frec[arr[i]] = 1;

// }

// }

//}

public static string Solve(string a, string b)

{

Dictionary<char, int> da = new Dictionary<char, int>();

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

{

if (da.ContainsKey(a[i]))

{

da[a[i]]++;

}

else

{

da[a[i]] = 1;

}

}

Dictionary<char, int> db = new Dictionary<char, int>();

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

{

if (db.ContainsKey(b[i]))

{

db[b[i]]++;

}

else

{

db[b[i]] = 1;

}

}

string ans = "";

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

{

if(da.ContainsKey(a[i]) && !db.ContainsKey(a[i]))

{

ans += a[i];

}

if (!da.ContainsKey(a[i]) && db.ContainsKey(a[i]))

{

ans += a[i];

}

}

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

{

if (da.ContainsKey(b[i]) && !db.ContainsKey(b[i]))

{

ans += b[i];

}

if (!da.ContainsKey(b[i]) && db.ContainsKey(b[i]))

{

ans += b[i];

}

}

return ans;

}

static void Main(string[] args)

{

//solve([3, 4, 4, 3, 6, 3]) = [4, 6, 3]

// Assert.That(Kata.Solve("xyabb", "xzca"), Is.EqualTo("ybbzc"));

// "nxrimrpikdtuedeok"

// was:

// "nxrriimpkkddtueeo"

//"racrcrzupispujufpipbmj"

//"rrracczuuuppppiisjjfbm"

Console.WriteLine(Solve("xyabb", "xzca"));

Console.ReadLine();

}

}

}