**7 kyu**

**Operations With Sets**

8377% of 7931 of454[raulbc777](https://www.codewars.com/users/raulbc777)

C#

* [TRAIN](https://www.codewars.com/kata/operations-with-sets/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/operations-with-sets/solutions/csharp)

[Discourse (38)](https://www.codewars.com/kata/operations-with-sets/discuss/csharp)

* |
* Share this kata:

We need a function that receives two arrays arr1 and arr2, each of them, with elements that occur only once. We need to know:

(1) Number of elements that are present in both arrays

(2) Number of elements that are present in only one array

(3) Number of remaining elements after extracting to arr1, the elements that are present in arr2.

(4) Number of remaining elements after extracting to arr2, the elements that are present in arr1.

Explaining the features of this function, let's name it process\_2arrays()(ruby and python), process2Arrays()(javascript), we show graphically how it should operate:

process\_2arrays(arr1, arr2) == [(1), (2), (3), (4)] # (data required above)

Let's see some cases:

arr1 = [1, 2 ,3,4, 5 ,6 ,7 ,8 ,9]

arr2 = [2, 4, 6, 8, 10, 12, 14]

process\_2arrays(arr1, arr2) --------> [4, 8, 5, 3]

(1) ---> 4 # because the elements present in both arryas are: 2, 4, 6 and 8 (4 values)

(2) ---> 8 # beacause elements present in only one array are: 1, 3, 5, 7, 9, 10, 12, and 14 (8 values)

(3) ---> 5 # elements remaning of arr1 are: 1, 3, 5, 7, 9

(4) ---> 3 # elements remaning of arr2 are: 10, 12, 14

No doubt, an easy kata to warm up before doing the more complex ones. Enjoy it!

<https://www.codewars.com/kata/operations-with-sets/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static int[] Process2Arrays(int[] arr1, int[] arr2)

{

//int max = Math.Max(arr1.Max(), arr2.Max());

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

List<int> lista = new List<int>();

lista.AddRange(arr1);

lista.AddRange(arr2);

for (int i = 0; i < lista.Count; i++)

{

if (frec.ContainsKey(lista[i]))

{

frec[lista[i]]++;

}

else

{

frec[lista[i]] = 1;

}

}

int ambos = 0;

int solo\_en\_uno = 0;

foreach(KeyValuePair<int,int > kvp in frec)

{

if(kvp.Value > 1)

{

ambos++;

}

if(kvp.Value == 1)

{

solo\_en\_uno++;

}

}

// Console.WriteLine(ambos + " " + solo\_en\_uno + " " + (arr1.Length-ambos) + " " + (arr2.Length-ambos));

return new int[] { ambos ,solo\_en\_uno , (arr1.Length - ambos) , (arr2.Length - ambos) };

}

static void Main(string[] args)

{

int[] arr1 = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

int[] arr2 = { 2, 4, 6, 8, 10, 12, 14 };

int[] res = Process2Arrays(arr1, arr2);

Console.WriteLine(res[0] + " " + res[1] + " " + res[2] + " " + res[3]);

Console.ReadLine();

}

}

}