**6 kyu**

**Challenge Fun #17: Merge Arrays**

12198% of 4612 of98[myjinxin2015](https://www.codewars.com/users/myjinxin2015)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/challenge-fun-number-17-merge-arrays/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/challenge-fun-number-17-merge-arrays/solutions/csharp)

[Forks (3)](https://www.codewars.com/kata/challenge-fun-number-17-merge-arrays/forks/csharp)

[Discourse (9)](https://www.codewars.com/kata/challenge-fun-number-17-merge-arrays/discuss/csharp)

* Add to Collection
* |
* Share this kata:

**Task**

You have two sorted arrays a and b, merge them to form new array of unique items.

If an item is present in both arrays, it should be part of the resulting array if and only if it appears in both arrays the same number of times.

**Example**

For a = [1, 3, 40, 40, 50, 60, 60, 60] and b = [2, 40, 40, 50, 50, 65], the result should be [1, 2, 3, 40, 60, 65].

Number 40 appears 2 times in both arrays,

thus it is in the resulting array.

Number 50 appears once in array a and twice in array b,

therefore it is not part of the resulting array.

**Input/Output**

* [input] integer array a

A sorted array.

1 ≤ a.length ≤ 500000

* [input] integer array b

A sorted array.

1 ≤ b.length ≤ 500000

* [output] an integer array

The resulting sorted array.

<https://www.codewars.com/kata/challenge-fun-number-17-merge-arrays/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

//public int DigitalRoot(long n)

//{

// // Your awesome code here!

//}

public static int[] MergeArrays(int[] a, int[] b)

{

//coding and coding..

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

int i, j;

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

{

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

else da[a[i]] = 1;

}

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

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

{

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

else db[b[i]] = 1;

}

i = 0;

j = 0;

int[] keys\_a = new HashSet<int>(da.Keys.ToArray()).ToArray();

int[] keys\_b = new HashSet<int>(db.Keys.ToArray()).ToArray();

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

while (i < keys\_a.Length && j < keys\_b.Length)

{

if (keys\_a[i] < keys\_b[j])

{

ans.Add(keys\_a[i]);

i++;

}

else if (keys\_b[j] < keys\_a[i])

{

ans.Add(keys\_b[j]);

j++;

}

else if(keys\_a[i] == keys\_b[j])

{

if(da[keys\_a[i]] == db[keys\_b[j]])

{

ans.Add(keys\_a[i]);

}

i++;

j++;

}

}

while(i < keys\_a.Length)

{

ans.Add(keys\_a[i]);

i++;

}

while(j < keys\_b.Length)

{

ans.Add(keys\_b[j]);

j++;

}

return ans.ToArray();

}

static void Main(string[] args)

{

//a = [1, 3, 40, 40, 50, 60, 60, 60]

//b = [2, 40, 40, 50, 50, 65],

//[1, 2, 3, 40, 60, 65].

int[] a = { 1, 3, 40, 40, 50, 60, 60, 60 };

int[] b = { 2, 40, 40, 50, 50, 65 };

foreach(int elem in MergeArrays(a,b) )

{

Console.Write(elem + " ");

}

Console.ReadLine();

}

}

}