You created an awesome array a, but some hooligan removed one element from it, shuffled it and returned it to you as array b.

Given arrays a and b, your task is to figure out what element was removed from a.

**Example**

For a =[1, 2, 3, 4] and b = [3, 1, 2],  
the output should be  
findOnlyDiffChar(a, b) = 4.

* **[time limit] 3000ms (cs)**
* **[input] array.integer a**

*Constraints:*  
1 ≤ a.length ≤ 4000,  
1 ≤ a[i] ≤ 1000.

* **[input] array.integer b**

Array consisting of exactly the same elements as a, except for a single missing element.

*Constraints:*  
b.length = a.length - 1.

* **[output] integer**

The missing element.

<https://codefights.com/challenge/RsQYfmimvek4JiiiG>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication2

{

class Program

{

static int findOnlyDiffChar(int[] a, int[] b)

{

Dictionary<int, int> frec\_a = a.GroupBy(x => x)

.ToDictionary(x => x.Key, x => x.Count());

Dictionary<int, int> frec\_b = b.GroupBy(x => x)

.ToDictionary(x => x.Key, x => x.Count());

int ans = 0;

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

{

if (!frec\_b.ContainsKey(kvp.Key))

{

ans = kvp.Key;

break;

}

else

{

if (kvp.Value > frec\_b[kvp.Key])

{

ans = kvp.Key;

break;

}

}

}

return ans;

}

static void Main(string[] args)

{

//int[] a= {1, 2, 3, 4};

//int[] b = { 3, 1, 2 };

int[] a = { 100, 99, 100, 99, 100, 99, 100 };

int[] b = { 99, 100, 99, 100, 99, 100 };

Console.WriteLine(findOnlyDiffChar(a, b));

Console.ReadLine();

}

}

}

------------OTRAS SOLUCIONES------------

int findOnlyDiffChar(int[] a, int[] b)

{

return a.Sum() - b.Sum();

}

int findOnlyDiffChar(int[] a, int[] b)

{

var l = a.ToList();

foreach (var x in b)

l.Remove(x);

return l[0];

}

int findOnlyDiffChar(int[] a, int[] b)

{

Array.Sort(a);

Array.Sort(b);

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

{

if (a[i] != b[i])

{

return a[i];

}

}

return a[a.Length - 1];

}