**7 kyu**

**Array element parity**

9194% of 13074 of502[KenKamau](https://www.codewars.com/users/KenKamau)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/array-element-parity/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/array-element-parity/solutions/csharp)

[Forks (1)](https://www.codewars.com/kata/array-element-parity/forks/csharp)

[Discourse (12)](https://www.codewars.com/kata/array-element-parity/discuss/csharp)

* Add to Collection
* |
* Share this kata:

In this Kata, you will be given an array of integers whose elements have both a negative and a positive value, except for one integer that is either only negative or only positive. Your task will be to find that integer.

For example,

solve[1,-1,2,-2,3] = 3 --> 3 only has a positive ocurrence.

solve([-3,1,2,3,-1,-4,-2]) = -4 --> -4 only has a negative occurence

solve([1,-1,2,-2,3,3]) = 3 --> the integer that is only positive or only negative my appear more than once.

<https://www.codewars.com/kata/array-element-parity/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static int Solve(List<int> arr)

{

//throw new NotImplementedException();

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

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

{

diccio[arr[i]] = i;

}

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

{

if (!diccio.ContainsKey(-arr[i])) return arr[i];

}

return -1;

}

static void Main(string[] args)

{

Console.WriteLine( Solve( new int[] { 1, -1, 2, -2, 3, 3 }.ToList() ));

//= -4-- > -4 only has a negative occurence

Console.ReadLine();

}

}

}

------------------------

[Unnamed](https://www.codewars.com/users/Unnamed), [mvdoyle](https://www.codewars.com/users/mvdoyle), [bayshaffer](https://www.codewars.com/users/bayshaffer)

**using System;**

**using System.Collections.Generic;**

**using System.Linq;**

**public static class Kata**

**{**

**public static int Solve(List<int> list)**

**{**

**return list.Distinct().Sum();**

**}**

**}**