**Ones and Zeros**

2174487% *of* 2,1581,500 *of* 13,964[user4386369](https://www.codewars.com/users/user4386369)

C#

* [Train Again](https://www.codewars.com/kata/ones-and-zeros/train/csharp)
* [Next Kata](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/ones-and-zeros/solutions/csharp)

[Forks (4)](https://www.codewars.com/kata/ones-and-zeros/forks/csharp)

[Discourse (150)](https://www.codewars.com/kata/ones-and-zeros/discuss/csharp)

* Add to Collection
* |
* Share this kata:

Given an array of one's and zero's convert the equivalent binary value to an integer.

Eg: [0, 0, 0, 1] is treated as 0001 which is the binary representation of 1

Examples:

Testing: [0, 0, 0, 1] ==> 1

Testing: [0, 0, 1, 0] ==> 2

Testing: [0, 1, 0, 1] ==> 5

Testing: [1, 0, 0, 1] ==> 9

Testing: [0, 0, 1, 0] ==> 2

Testing: [0, 1, 1, 0] ==> 6

Testing: [1, 1, 1, 1] ==> 15

Testing: [1, 0, 1, 1] ==> 11

<https://www.codewars.com/kata/ones-and-zeros/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp116

{

class Program

{

public static int binaryArrayToNumber(int[] BinaryArray)

{

//Code here

int sum = 0;

int potencia = 0;

for(int i = BinaryArray.Length-1; i>=0; i--)

{

sum += ((int)(Math.Pow(2, potencia++)) \* BinaryArray[i]);

}

return sum;

}

static void Main(string[] args)

{

int[] bin = { 1, 0, 1, 1 };

Console.WriteLine(binaryArrayToNumber(bin));

Console.ReadLine();

}

}

}