**6 kyu**

**Find Numbers with Same Amount of Divisors**

13399% of 4311 of137[raulbc777](http://www.codewars.com/users/raulbc777)

C#

* [TRAIN AGAIN](http://www.codewars.com/kata/find-numbers-with-same-amount-of-divisors/train/csharp)
* [NEXT KATA](http://www.codewars.com/trainer/csharp)

Details

[Solutions](http://www.codewars.com/kata/find-numbers-with-same-amount-of-divisors/solutions/csharp)

[Forks (1)](http://www.codewars.com/kata/find-numbers-with-same-amount-of-divisors/forks/csharp)

[Discourse (34)](http://www.codewars.com/kata/find-numbers-with-same-amount-of-divisors/discuss/csharp)

* Add to Collection
* |
* Share this kata:

The integers 14 and 15, are contiguous (1 the difference between them, obvious) and have the same number of divisors.

14 ----> 1, 2, 7, 14 (4 divisors)

15 ----> 1, 3, 5, 15 (4 divisors)

The next pair of contiguous integers with this property is 21 and 22.

21 -----> 1, 3, 7, 21 (4 divisors)

22 -----> 1, 2, 11, 22 (4 divisors)

We have 8 pairs of integers below 50 having this property, they are:

"[[2, 3], [14, 15], [21, 22], [26, 27], [33, 34], [34, 35], [38, 39], [44, 45]]"

Let's see now the integers that have a difference of 3 between them. There are seven pairs below 100:

"[[2, 5], [35, 38], [55, 58], [62, 65], [74, 77], [82, 85], [91, 94]]"

Let's name, diff, the difference between two integers, next and prev, (diff = next - prev) and nMax, an upper bound of the range.

We need a special function, count\_pairsInt(), that receives two arguments, diff and nMax and outputs the amount of pairs of integers that fulfill this property, all of them being smaller (not smaller or equal) than nMax.

Let's see it more clearly with examples.

count\_pairsInt(1, 50) -----> 8 (See case above)

count\_pairsInt(3, 100) -----> 7 (See case above)

Happy coding!!!

<http://www.codewars.com/kata/find-numbers-with-same-amount-of-divisors/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp2

{

class Program

{

static int cantidad\_divisores(int n)

{

int sqr = (int)Math.Sqrt(n);

int cont = 0;

for (int i = 1; i <= sqr; i++)

{

if (n % i == 0)

{

cont++;

}

}

cont \*= 2;

if ((sqr \* sqr) == n) cont = cont - 1;

return cont;

}

public static int CountPairsInt(int diff, long nMax)

{

int cont = 0;

for (int i = 1; i + diff < nMax; i++)

{

if (cantidad\_divisores(i) == cantidad\_divisores(i + diff))

{

//Console.WriteLine(i + " " + (i + diff));

cont++;

}

}

return cont;

}

static void Main(string[] args)

{

//Console.WriteLine(cantidad\_divisores(20));

Console.WriteLine(CountPairsInt(7, 1500)); //189

// Console.WriteLine(CountPairsInt(3,100));

Console.ReadLine();

}

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication12;

/\*\*

\*

\* @author Usuario

\*/

public class JavaApplication12 {

static int cantidad\_divisores(int n)

{

int sqr = (int)Math.sqrt(n);

int cont = 0;

for (int i = 1; i <= sqr; i++)

{

if (n % i == 0)

{

cont++;

}

}

cont \*= 2;

if ((sqr \* sqr) == n) cont = cont - 1;

return cont;

}

public static int CountPairsInt(int diff, long nMax)

{

int cont = 0;

for (int i = 1; i + diff < nMax; i++)

{

if (cantidad\_divisores(i) == cantidad\_divisores(i + diff))

{

cont++;

}

}

return cont;

}

public static void main(String[] args) {

// TODO code application logic here

}

}