**Find the next perfect square!**

2413888% of 3,2252,059 of23,059[kphurley](http://www.codewars.com/users/kphurley)

C#

* [TRAIN AGAIN](http://www.codewars.com/kata/find-the-next-perfect-square/train/csharp)
* [NEXT KATA](http://www.codewars.com/trainer/csharp)

Details

[Solutions](http://www.codewars.com/kata/find-the-next-perfect-square/solutions/csharp)

[Forks (4)](http://www.codewars.com/kata/find-the-next-perfect-square/forks/csharp)

[Discourse (61)](http://www.codewars.com/kata/find-the-next-perfect-square/discuss/csharp)

* Add to Collection
* |
* Share this kata:

You might know some pretty large perfect squares. But what about the NEXT one?

Complete the findNextSquare method that finds the next integral perfect square after the one passed as a parameter. Recall that an integral perfect square is an integer n such that sqrt(n) is also an integer.

If the parameter is itself not a perfect square, than -1 should be returned. You may assume the parameter is positive.

**Examples:**

findNextSquare(121) --> returns 144

findNextSquare(625) --> returns 676

findNextSquare(114) --> returns -1 since 114 is not a perfect

<http://www.codewars.com/kata/find-the-next-perfect-square/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp112

{

class Program

{

public static long FindNextSquare(long num)

{

// your code here

long sqr = (long)Math.Sqrt(num);

if (sqr \* sqr != num) return -1;

return (sqr + 1) \* (sqr + 1);

}

static void Main(string[] args)

{

Console.WriteLine(FindNextSquare(121));

Console.ReadLine();

}

}

}