**6 kyu**

**Sum of Digits / Digital Root**

64113189% of4,9023,214 of37,043[user578387](https://www.codewars.com/users/user578387)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/sum-of-digits-slash-digital-root/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/sum-of-digits-slash-digital-root/solutions/csharp)

[Forks (14)](https://www.codewars.com/kata/sum-of-digits-slash-digital-root/forks/csharp)

[Discourse (513)](https://www.codewars.com/kata/sum-of-digits-slash-digital-root/discuss/csharp)

* Add to Collection
* |
* Share this kata:

In this kata, you must create a digital root function.

A digital root is the *recursive sum of all the digits in a number.* Given *n*, take the sum of the digits of *n*. If that value has two digits, continue reducing in this way until a single-digit number is produced. This is only applicable to the natural numbers.

Here's how it works:

digital\_root(16)

=> 1 + 6

=> 7

digital\_root(942)

=> 9 + 4 + 2

=> 15 ...

=> 1 + 5

=> 6

digital\_root(132189)

=> 1 + 3 + 2 + 1 + 8 + 9

=> 24 ...

=> 2 + 4

=> 6

digital\_root(493193)

=> 4 + 9 + 3 + 1 + 9 + 3

=> 29 ...

=> 2 + 9

=> 11 ...

=> 1 + 1

=> 2

<https://www.codewars.com/kata/sum-of-digits-slash-digital-root/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static int DigitalRoot(long n)

{

// Your awesome code here!

long sum = 0;

long aux = n;

do

{

sum = 0;

while (aux > 0)

{

sum += (aux % 10);

aux /= 10;

}

aux = sum;

} while (sum >= 10);

return (int)sum;

}

static void Main(string[] args)

{

Console.WriteLine(DigitalRoot(942));

Console.ReadLine();

}

}

}