**8 kyu**

**Collatz Conjecture (3n+1)**

401391% of 40496 of2,211[Goncalerta](https://www.codewars.com/users/Goncalerta" \o "This kata's Sensei)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/collatz-conjecture-3n-plus-1/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/collatz-conjecture-3n-plus-1/solutions/csharp)

[Forks (1)](https://www.codewars.com/kata/collatz-conjecture-3n-plus-1/forks/csharp)

[Discourse (16)](https://www.codewars.com/kata/collatz-conjecture-3n-plus-1/discuss/csharp)

* Add to Collection
* |
* Share this kata:

The Collatz conjecture (also known as 3n+1 conjecture) is a conjecture that applying the following algorithm to any number we will always eventually reach one:

[This is writen in pseudocode]

if(number is even) number = number / 2

if(number is odd) number = 3\*number + 1

#Task

Your task is to make a function hotpo that takes a positive n as input and returns the number of times you need to perform this algorithm to get n = 1.

#Examples

hotpo(1) returns 0

(1 is already 1)

hotpo(5) returns 5

5 -> 16 -> 8 -> 4 -> 2 -> 1

hotpo(6) returns 8

6 -> 3 -> 10 -> 5 -> 16 -> 8 -> 4 -> 2 -> 1

hotpo(23) returns 15

23 -> 70 -> 35 -> 106 -> 53 -> 160 -> 80 -> 40 -> 20 -> 10 -> 5 -> 16 -> 8 -> 4 -> 2 -> 1

#References

* Collatz conjecture wikipedia page: <https://en.wikipedia.org/wiki/Collatz_conjecture>

<https://www.codewars.com/kata/collatz-conjecture-3n-plus-1/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static uint Hotpo(uint n)

{

uint cont = 0;

while (n > 1)

{

if(n % 2 ==0)

{

n /= 2;

}

else

{

n = n \* 3 + 1;

}

cont++;

}

return cont;

}

static void Main(string[] args)

{

Console.ReadLine();

}

}

}