Author

[SourMongoose](https://codefights.com/profile/SourMongoose)

https://codefights.com/img/coins_new.png2000

The *hailstone sequence* is generated from a number n by the following set of rules:

* if n is even, divide it by 2 to get n/2;
* if n is odd, multiply it by 3 and add 1 to get 3n + 1.

Let the *length* of a *hailstone sequence* be the number of steps it takes to reach 1.

For example, the *length* of the *hailstone sequence* starting with 6 equals 8, since the sequence is generated is follows:  
6 → 3 → 10 → 5 → 16 → 8 → 4 → 2 → 1.

Given a length l, find the minimum positive integer X such that the *length* of the sequence starting with X equals l.

**Example**

For l = 8, the output should be  
hailstoneLength(l) = 6.

* **[input] integer l**

The *length* of the *halistone sequence*, 0 ≤ l ≤ 350.

* **[output] integer**

The minimum number X such that the length of the sequence starting with X has*length* l.

<https://codefights.com/challenge/kKsosHXWQq64ZoqRE>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int sec(int n)

{

int len=0;

while (n > 1)

{

// Console.Write(n + " ");

if (n % 2 == 0)

{

n = n / 2;

}

else

{

n = 3 \* n + 1;

}

len++;

}

return len;

}

static int hailstoneLength(int l)

{

int n = 1;

while (true )

{

if (sec(n) == l)

{

return n;

}

n++;

}

}

static void Main(string[] args)

{

Console.WriteLine(hailstoneLength(8));

Console.ReadLine();

}

}

}