Consider the following algorithm for the given positive integer n:

1. print n
2. if n = 1 then STOP;
3. if n is odd then   
   n = 3 \* n + 1  
   else  
   n = n / 2;
4. goto 2

For the given integer n, return the second largest value that will be printed by the algorithm above.

**Example**

For n = 22, the output should be  
secondLargest(n) = 40.

The following values will be printed:  
22 11 34 16 52 26 13 40 20 10 5 16 8 4 2 1.

The second largest number in this sequence is40, which is the answer.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] integer n**

An integer.

*Constraints:*  
2 ≤ n ≤ 1000.

* **[output] integer**

The second largest number that will be printed.

<https://codefights.com/challenge/qQ762Pzrgm5KaBMT6/main?utm_source=challengeOfTheDay&utm_medium=email&utm_campaign=email_notification>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int secondLargest(int n)

{

if (n <= 2) return 1;

List<int> lista = new List<int>();

lista.Add(n);

while (n>1)

{

if (n % 2 != 0)

{

n = (3 \* n) + 1;

}

else

{

n = n / 2;

}

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

lista.Add(n);

}

lista.Sort();

return lista[lista.Count - 2];

}

static void Main(string[] args)

{

// Console.WriteLine( count2sI(502) );

Console.WriteLine(secondLargest(22));

Console.ReadLine();

}

}

}