

Attendance War

Time: 1s | Memory: 256 MB

Problem Statement

Imagine you are late to your Mathematics class, and the professor gives you the challenge to solve a problem and if you could solve it before the class ends, you are allowed to the class. Being a programming enthusiast, you offer to solve that mathematical problem using one of your favorite programming languages. The professor smirks and agrees to that.

The problem goes something like this:

The professor gives you an integer (x). It can be any positive integer.

If the integer is odd, you have to perform $3x+1$ on the integer.

If the integer is even, you have to perform $x/2$ on the integer.

However, the problem does not end here. You have to keep on performing the operation until you reach number 1.

The professor guarantees that no matter what number you start with, you will always end in 1 if you apply the above even-odd condition.

For example:

If the professor gives the number 5, the following is the sequence of operation:

5 - Odd: So $3x+1 = 3 * 5 + 1 = 16$ (note you have to continue with the answer as input for next operation until you reach 1 - which is guaranteed by the professor)

16 - Even: So $x / 2 = 16 / 2 = 8$

8 - Even: $8 / 2 = 4$

4 - Even: $4 / 2 = 2$

2 = Even: $2 / 2 = 1$

1 = Stop

Your program ends here, as the output has reached 1.

Now the number of steps it took to reach 1 is the answer. In the above example, when 5 was the initial input, the number of steps it took to reach 1 was 5 steps.

So the answer is 5.

Input Format

The input consists of a single line containing one integer n ($1 \leq n \leq 10^6$)

Output Format

Print a single integer: the number of steps needed to reach 1 from n .

If $n = 1$, print 0.

Sample Input I

1

Sample Output I

0

Sample Input II

6

Sample Output II

8

Sample Input III

27

Sample Output III

111