952. Digital Problem

* [Description](http://www.lintcode.com/en/problem/digital-problem/" \l "description)
* [Notes](http://www.lintcode.com/en/problem/digital-problem/#note)
* [Testcase](http://www.lintcode.com/en/problem/digital-problem/#testcase)
* [Judge](http://www.lintcode.com/en/problem/digital-problem/#judge)

[Discuss](http://www.lintcode.com/discuss)

Give a conversion rule to convert number n：  
1. n is an odd number: n = 3 \* n + 1  
2. n is an even number: n = n / 2  
3. After several conversions, n will become 1.

Given a number n, find the times of converting to 1.

 Notice

* 1 <= n <= 1000000

Have you met this question in a real interview?

Yes

**Example**

Given n = 2, return 1.

Explanation:

2→1

Given n = 3, return 7.

Explanation:

3→10→5→16→8→4→2→1

<http://www.lintcode.com/en/problem/digital-problem/#>

public int digitConvert(int n) {

// Write your code here

int cont =0;

while(n != 1) {

if(n%2!=0) {

n= 3\*n+1;

}else {

n/=2;

}

cont++;

}

return cont;

}