

将数字变成 0 的操作次数

给你一个非负整数 `num`，请你返回将它变成 0 所需要的步数。 如果当前数字是偶数，你需要把它除以 2；否则，减去 1。

示例 1：

输入：`num = 14`

输出：6

解释：

步骤 1) 14 是偶数，除以 2 得到 7。

步骤 2) 7 是奇数，减 1 得到 6。

步骤 3) 6 是偶数，除以 2 得到 3。

步骤 4) 3 是奇数，减 1 得到 2。

步骤 5) 2 是偶数，除以 2 得到 1。

步骤 6) 1 是奇数，减 1 得到 0。

示例 2：

输入：`num = 8`

输出：4

解释：

步骤 1) 8 是偶数，除以 2 得到 4。

步骤 2) 4 是偶数，除以 2 得到 2。

步骤 3) 2 是偶数，除以 2 得到 1。

步骤 4) 1 是奇数，减 1 得到 0。

示例 3：

输入：`num = 123`

输出：12

```
class Solution {
public:
    int numberOfSteps(int num) {
        int step=0;
        while(num)
        {
            if(num%2==0)
            {
                num=num/2;
            }
            else
            {
                num=num-1;
            }
            step++;
        }
        return step;
    }
};
```