

7. Reverse Integer (LeetCode)

<https://leetcode.com/problems/reverse-integer/description/>

The screenshot shows the LeetCode problem page for '7. Reverse Integer'. The problem is marked as 'Solved' with a green checkmark. The difficulty is 'Medium'. The description states: 'Given a signed 32-bit integer x , return x with its digits reversed. If reversing x causes the value to go outside the signed 32-bit integer range $[-2^{31}, 2^{31} - 1]$, then return 0. Assume the environment does not allow you to store 64-bit integers (signed or unsigned).' The interface shows 11.8K votes, 12.9K views, and a 'Testcase' tab with 'Accepted' status. Performance metrics are displayed: Runtime is 1 ms (Beats 97.85% of users with Java) and Memory is 39.12 MB (Beats 97.16% of users with Java). The 'Code' tab on the right shows a Java solution:

```
1 class Solution {
2     public int reverse(int x) {
3         int result=0;
4         int remainder=0;
5
6         while(x!=0){
7             remainder=x%10;
8             if((result>(Integer.MAX_VALUE/10)) || (result<(Integer.MIN_VALUE/10))
9         ){
10             return 0;
11         }
12         result=result*10+remainder;
13         x=x/10;
14         return result;
15     }
16 }
17 }
```

1009. Complement of base 10 Integer (LeetCode)

<https://leetcode.com/problems/complement-of-base-10-integer/description/>

The screenshot shows the LeetCode problem page for '1009. Complement of Base 10 Integer'. The problem is marked as 'Solved' with a green checkmark. The difficulty is 'Easy'. The description states: 'The **complement** of an integer is the integer you get when you flip all the 0's to 1's and all the 1's to 0's in its binary representation. For example, The integer 5 is "101" in binary and its **complement** is "010" which is the integer 2. Given an integer n , return its complement.' The interface shows 2.2K votes, 108 views, and a 'Testcase' tab with 'Accepted' status. Performance metrics are displayed: Runtime is 10 ms (Beats 5.72% of users with Java) and Memory is 40.00 MB (Beats 6.82% of users with Java). The 'Code' tab on the right shows a Java solution:

```
1 class Solution {
2     public int bitwiseComplement(int n) {
3         String string="";
4         if(n==0){
5             return 1;
6         }
7         if(n==1){
8             return 0;
9         }
10        while(n!=0){
11            int bit=n&1;
12            if(bit==0){
13                bit=1;
14            }else{
15                bit=0;
16            }
17            string=bit+string;
18            n=n>>1;
19        }
20        System.out.println(string);
21        int decimal=Integer.parseInt(string,2);
22        return decimal;
23    }
24 }
25 }
```

231. Power of Two (LeetCode)

<https://leetcode.com/problems/power-of-two/description/>

231. Power of Two Solved

Easy Topics Companies

Given an integer `n`, return `true` if it is a power of two. Otherwise, return `false`.

An integer `n` is a power of two, if there exists an integer `x` such that `n == 2x`.

Example 1:

Input: `n = 1`

5.9K 379

Testcase Result

Accepted

Editorial + Solution

Runtime	Details	Memory	Details
1 ms		39.45 MB	
Beats 86.54% of users with Java		Beats 67.46% of users with Java	

```
1 class Solution {
2     public boolean isPowerOfTwo(int n) {
3         int count=0;
4         if(n==0 || n<0){
5             return false;
6         }
7         while(n>0){
8             int bit=n&1;
9             if(bit==1){
10                 count++;
11             }
12             if(count>1){
13                 return false;
14             }
15             n=n>>1;
16         }
17         return true;
18     }
19 }
```

Saved to local Ln 4, Col 23