LeetCode Problem Solving Session

15 January 2022 18:53

Q1- Reverse Integer

Example test cases : I/P :- 123
O/P :- 321

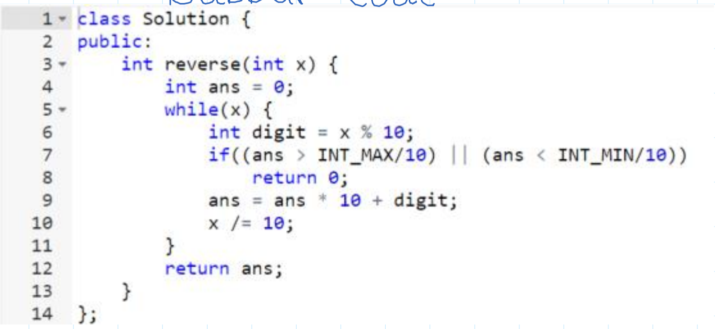
I/P :- -123
O/P :- -321

I/P :- -2147483646
O/P :- 0

(Because the reversed integer exceeds the range of int.)

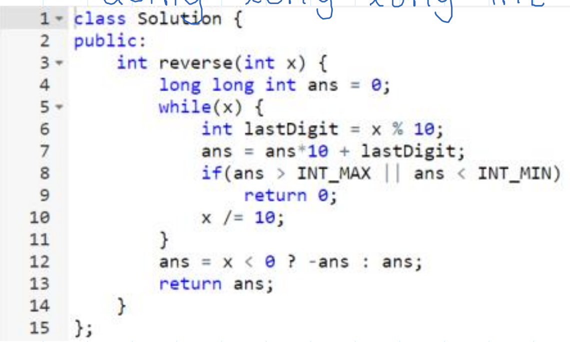
Hint : Check for overflow in every iteration before or after updating ans.

Babbar Code



```
class Solution {
public:
    int reverse(int x) {
        int ans = 0;
        while(x) {
            int digit = x % 10;
            if((ans > INT_MAX/10) || (ans < INT_MIN/10))
                return 0;
            ans = ans * 10 + digit;
            x /= 10;
        }
        return ans;
    }
};
```

Using long long int



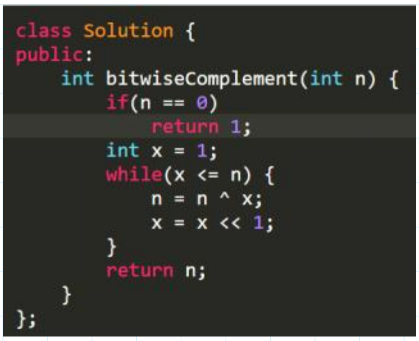
```
class Solution {
public:
    int reverse(int x) {
        long long int ans = 0;
        while(x) {
            int lastDigit = x % 10;
            ans = ans*10 + lastDigit;
            if(ans > INT_MAX || ans < INT_MIN)
                return 0;
            x /= 10;
        }
        ans = x < 0 ? -ans : ans;
        return ans;
    }
};
```

Q2 Complement of Base-10 integer.

Approach-1 : $n = 5 = 101$

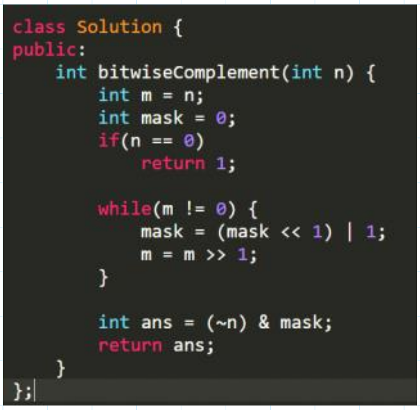
$\begin{array}{r} 101 \\ \wedge\quad 1 \\ \hline 100 \end{array}$, $\begin{array}{r} 100 \\ \wedge\quad 1 \\ \hline 110 \end{array}$, $\begin{array}{r} 110 \\ \wedge\ 1 \\ \hline 010 \end{array}$ $\Rightarrow$ $\textcircled{2}$

Let $x = 1$, till $x <= n$, do $n \wedge x$ and then $x << 1$.



```
class Solution {
public:
    int bitwiseComplement(int n) {
        if(n == 0)
            return 1;
        int x = 1;
        while(x <= n) {
            n = n ^ x;
            x = x << 1;
        }
        return n;
    }
};
```

Approach - 2 : Babbar Method. (Video dekho)

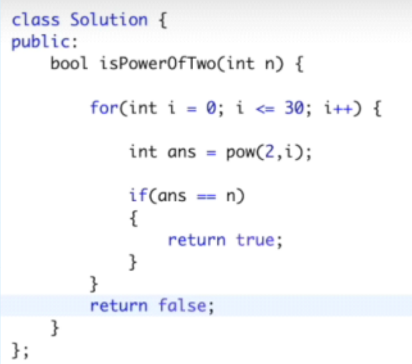


```
class Solution {
public:
    int bitwiseComplement(int n) {
        int m = n;
        int mask = 0;
        if(n == 0)
            return 1;

        while(m != 0) {
            mask = (mask << 1) | 1;
            m = m >> 1;
        }

        int ans = (~n) & mask;
        return ans;
    }
};
```

Q3 Power of Two.

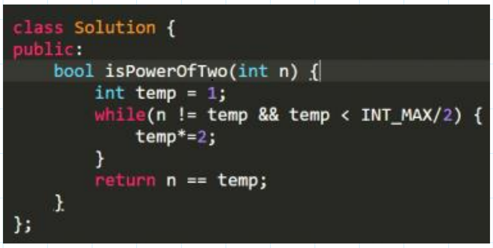
Hint : Any negative integer is not a power of 2.

① → Brute Force



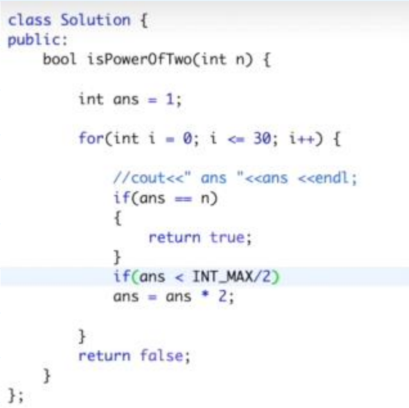
```
class Solution {
public:
    bool isPowerOfTwo(int n) {

        for(int i = 0; i <= 30; i++) {

            int ans = pow(2,i);

            if(ans == n)
            {
                return true;
            }
        }
        return false;
    }
};
```

② Improved Brute Force



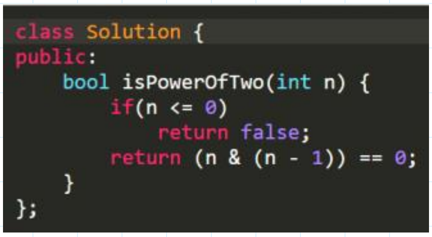
```
class Solution {
public:
    bool isPowerOfTwo(int n) {
        int temp = 1;
        while(n != temp && temp < INT_MAX/2) {
            temp*=2;
        }
        return n == temp;
    }
};
```

OR



```
class Solution {
public:
    bool isPowerOfTwo(int n) {

        int ans = 1;

        for(int i = 0; i <= 30; i++) {

            //cout<<" ans "<<ans <<endl;
            if(ans == n)
            {
                return true;
            }
            if(ans < INT_MAX/2)
            ans = ans * 2;

        }
        return false;
    }
};
```

③ Homework (Most efficient)



```
class Solution {
public:
    bool isPowerOfTwo(int n) {
        if(n <= 0)
            return false;
        return (n & (n - 1)) == 0;
    }
};
```

Hint : n & (n-1) flips the rightmost SET bit.

```
Eg:  n = 18      =>    000....010010

  18 & (17)     =>    000....010010
                   &  000....010001
                      -------------
                      000....010000
```