Experiment-4

Student Name: Chaitanya Gaba **UID:** 22BCS15984
Branch: BE-CSE **Section/Group:** KPIT-901/B
Semester: 6th **Date of Performance:** 18/01/25
Subject Name: Advanced Programming Lab - 2 **Subject Code:** 22CSP-351

1. **Aim:** Divide and Conquer
   1. Problem: 190. Reverse Bits.
   2. Problem: 191. Number of 1 Bits

2. **Objective:**
   1. Problem 190. To reverse the bits of a given 32-bit unsigned integer. Implement an optimized bit manipulation technique to reverse bits.
   2. Problem 191.Number of 1 Bits. To count the number of set bits (1s) in a given 32-bit unsigned integer.

3. **Implementation/Code:**

1.)

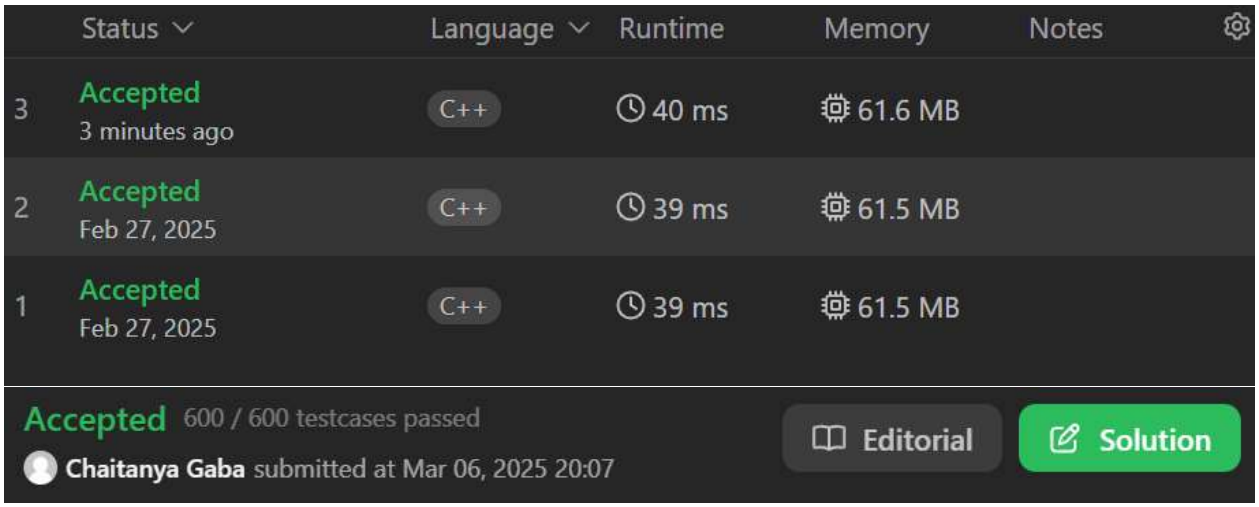
```
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        uint32_t result = 0;
        for (int i = 0; i < 32; ++i) {
            result = (result << 1) | (n & 1);
            n >>= 1;
        }
        return result;
    }
};
```

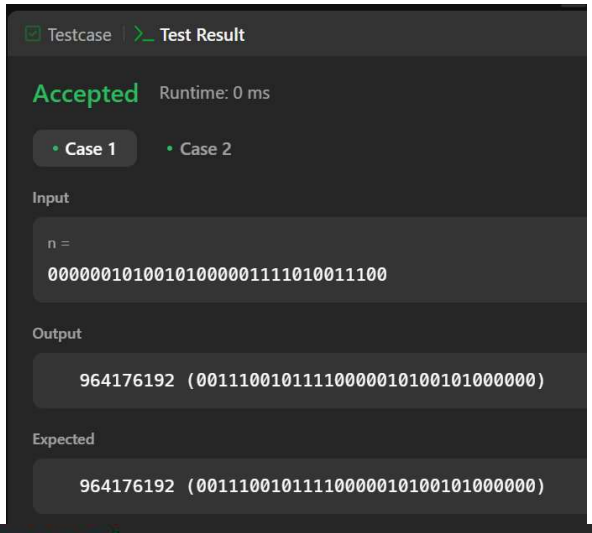
2.)

```
class Solution {
public:
    int hammingWeight(int n) {
        int count = 0;
        while (n) {
            n &= (n - 1);
            count++;
        }
        return count;
    }
};
```

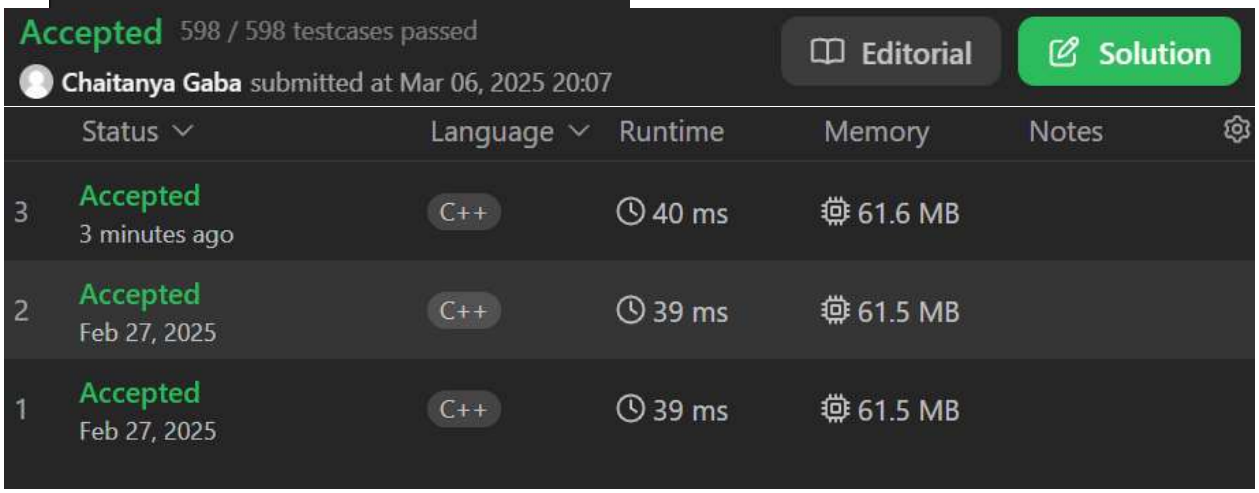
4. Output:

1.




Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
n =
00000010100101000001111010011100
Output
964176192 (00111001011110000010100101000000)
Expected
964176192 (00111001011110000010100101000000)

2.


Accepted 598 / 598 testcases passed
Chaitanya Gaba submitted at Mar 06, 2025 20:07
Editorial
Solution
Status
Language
Runtime
Memory
Notes
3
Accepted
3 minutes ago
C++
40 ms
61.6 MB
2
Accepted
Feb 27, 2025
C++
39 ms
61.5 MB
1
Accepted
Feb 27, 2025
C++
39 ms
61.5 MB

5. **Time Complexity:**
   1. O(1)
   2. O(k)

6. **Space Complexity:**
   1. O(1)
   2. O(1)

7. **Learning Outcome:**
   1. Understanding bitwise operations for reversing bits.
   2. Implementing lookup tables for faster computation.
   3. Utilizing intrinsic functions for hardware-level optimization.
   4. Mastering bitwise AND & shifting for counting bits.
   5. Exploring hardware-accelerated functions.
   6. Optimizing solutions using precomputed tables for fast lookups.