Experiment 3

Student Name: Amit Kumar Sahu **UID:** 22BCS50073

Branch: CSE **Section/Group:**22BCS_NTPP_IOT_602/A

Semester: 6th **DateofPerformance:**03/02/2025

Subject Name: AP LAB - 2 **SubjectCode:**22CSP-351

Problem-1

1. **Aim:**

   Find the longest substring where every letter appears in both uppercase and lowercase. Return the earliest occurrence if multiple exist; return an empty string if none exist.

2. **Objective:**

   - Identify the longest contiguous substring where each letter appears in both uppercase and lowercase.
   - Return the earliest such substring if multiple exist; otherwise, return an empty string.

3. **Implementation:**

```
class Solution {
public:
    string longestNiceSubstring(string s) {
        int n = s.size();
        for (int len = n; len > 0; len--) {
            for (int i = 0; i + len <= n; i++) {
                string sub = s.substr(i, len);
                unordered_set<char> st(sub.begin(), sub.end());
                bool nice = true;
                for (char c : sub) {
                    if (!st.count(tolower(c)) || !st.count(toupper(c))) {
                        nice = false;
                        break;
                    }
```

```
            }
            if (nice) return sub;
        }
    }
    return "";
  }
};
```

4. Output:

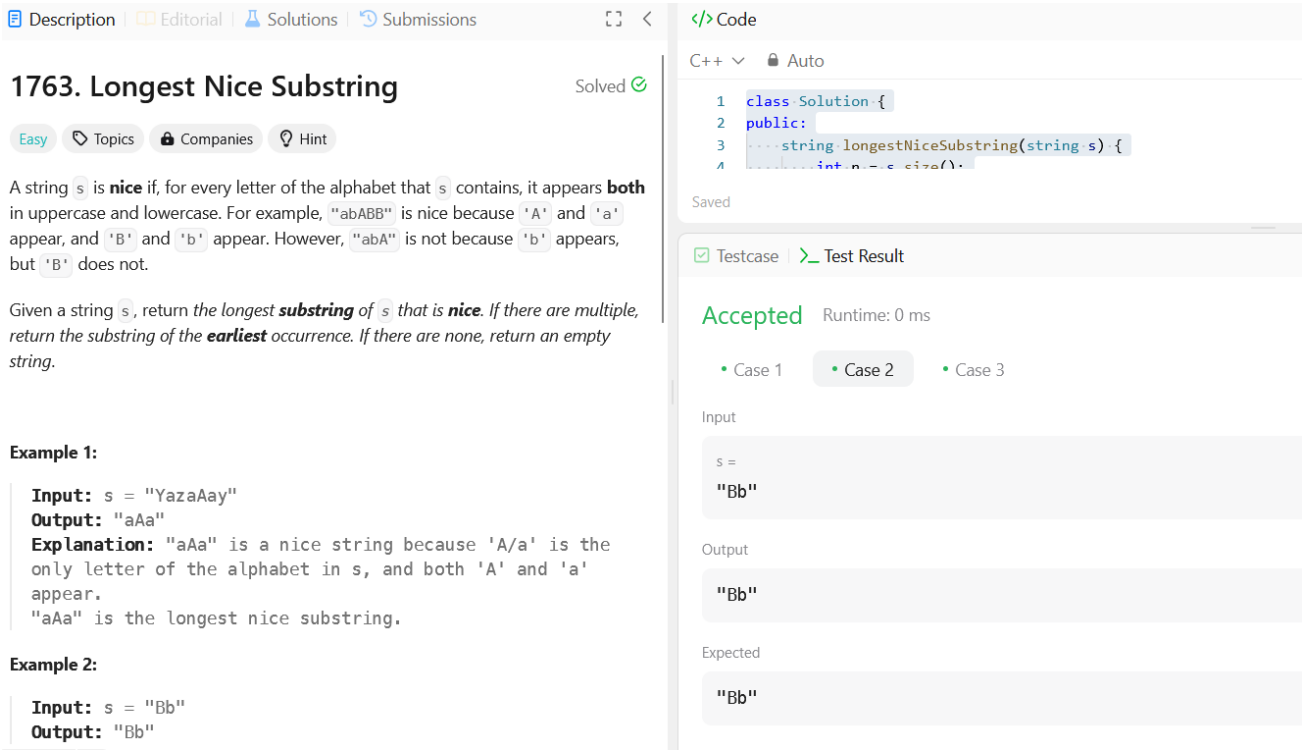


Fig: Longest Nice Substring.

Problem-2

1. Aim:

Reverse the bits of a given 32-bit unsigned integer and return the resulting value.

2. Objective:

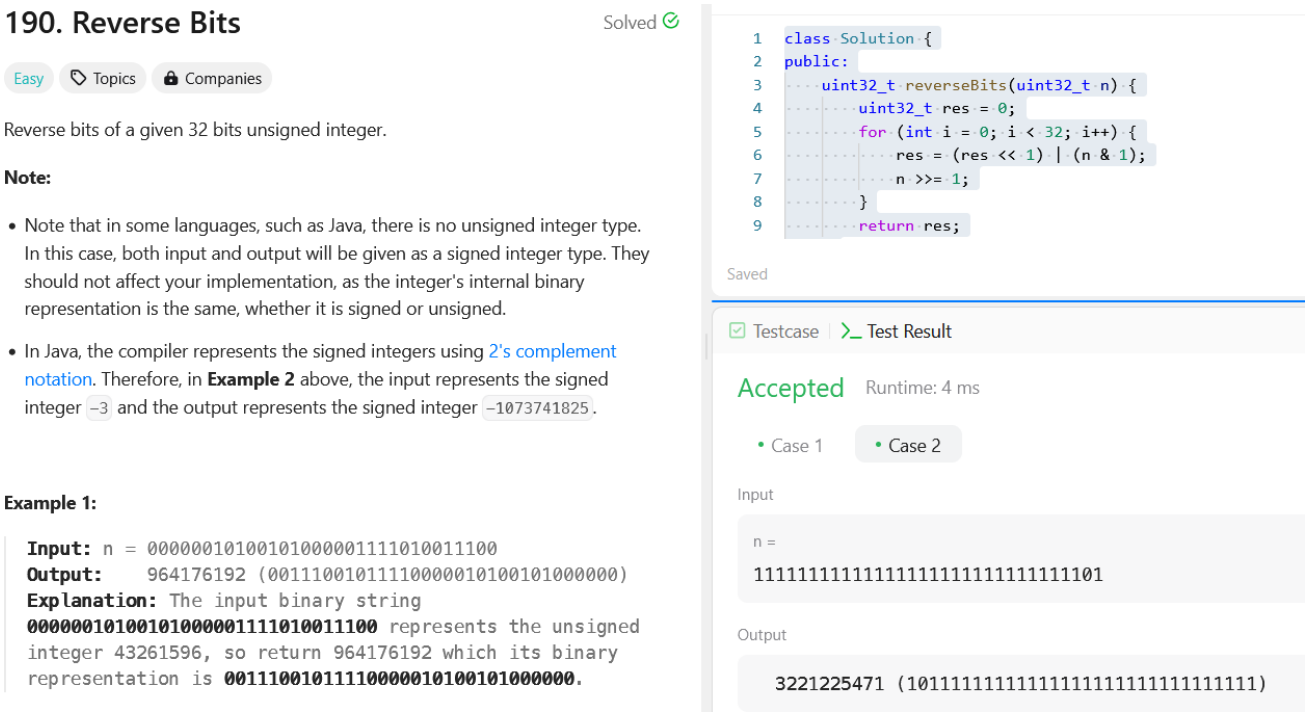
1 Process the 32-bit integer by reversing its binary representation.

2 Return the corresponding integer value of the reversed binary.

3. Implementation:

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

4. Output:



190. Reverse Bits

Solved

Easy | Topics | Companies

Reverse bits of a given 32 bits unsigned integer.

Note:

- Note that in some languages, such as Java, there is no unsigned integer type. In this case, both input and output will be given as a signed integer type. They should not affect your implementation, as the integer's internal binary representation is the same, whether it is signed or unsigned.
- In Java, the compiler represents the signed integers using 2's complement notation. Therefore, in **Example 2** above, the input represents the signed integer `-3` and the output represents the signed integer `-1073741825`.

Example 1:

Input: n = 00000010100101000001111010011100
Output: 964176192 (00111001011110000010100101000000)
Explanation: The input binary string **00000010100101000001111010011100** represents the unsigned integer 43261596, so return 964176192 which its binary representation is **00111001011110000010100101000000**.

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

Saved

Testcase | Test Result

Accepted Runtime: 4 ms

Case 1 | Case 2

Input

n =
11111111111111111111111111111101

Output

3221225471 (10111111111111111111111111111111)

Fig: Reverse Bits.