Assignment -4

Name :Atul **Uid-22Bbcs15834**

Section-605-B **Subject:AP**

Q1. Longest Nice Substring

```
class Solution {
public:
    string longestNiceSubstring(string s) {
        if (s.size() < 2) return "";
        unordered_set<char> st(s.begin(), s.end());
        for (int i = 0; i < s.size(); i++) {
            if (st.count(tolower(s[i])) && st.count(toupper(s[i]))) continue;
            string left = longestNiceSubstring(s.substr(0, i));
            string right = longestNiceSubstring(s.substr(i + 1));
            return left.size() >= right.size() ? left : right;
        }
        return s;
    }
};
```

Q2. Reverse Bits

```
class Solution {
public:
```

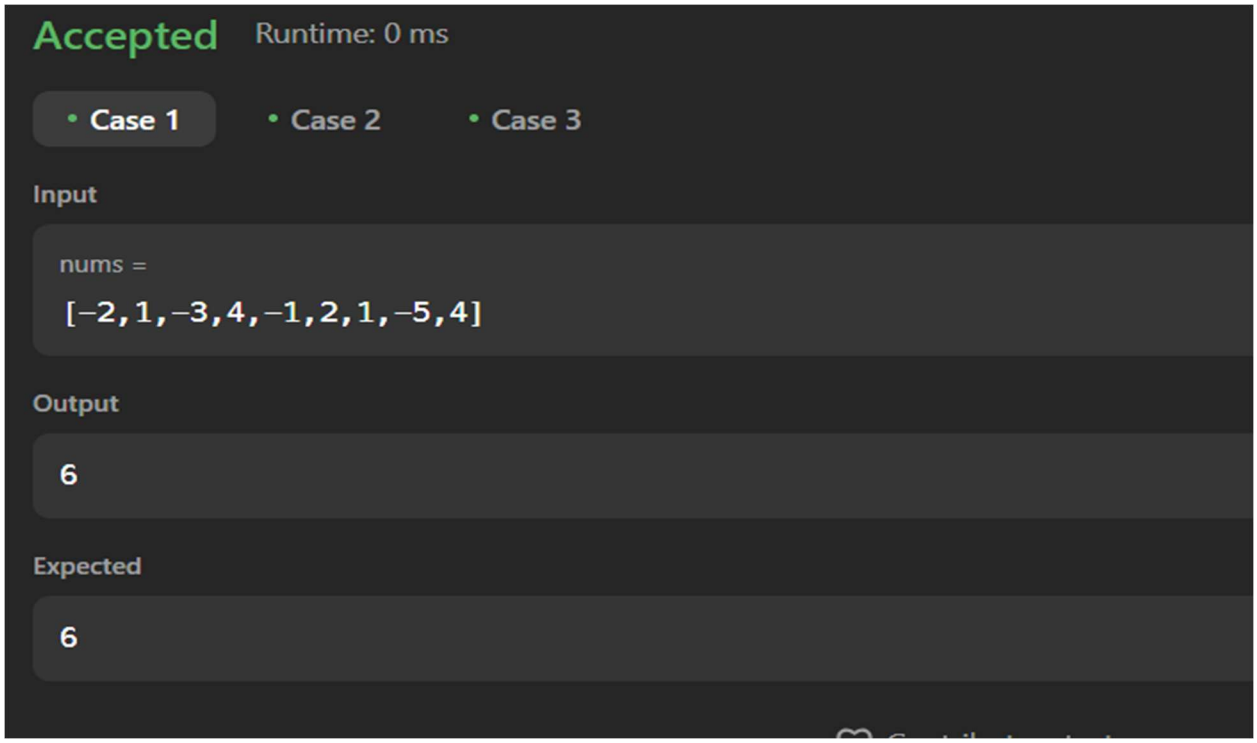
```
    uint32_t reverseBits(uint32_t n) {
        uint32_t ans = 0;
        for (int i = 0; i < 32; i++) {
            ans = (ans << 1) | (n & 1);
            n >>= 1;
        }
        return ans;
    }
};
```

Q3. Number of 1 Bits

```
class Solution {
public:
    int hammingWeight(uint32_t n) {
        int count = 0;
        while (n) {
            count += n & 1;
            n >>= 1;
        }
        return count;
    }
};
```

Q4. Maximum Subarray

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int maxSum = nums[0], currSum = nums[0];
        for (int i = 1; i < nums.size(); i++) {
            currSum = max(nums[i], currSum + nums[i]);
            maxSum = max(maxSum, currSum);
        }
        return maxSum;
    }
};
```



Q5. Search a 2D Matrix II

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int row = 0, col = matrix[0].size() - 1;
        while (row < matrix.size() && col >= 0) {
            if (matrix[row][col] == target) return true;
            else if (matrix[row][col] > target) col--;
            else row++;
        }
        return false;
    }
};
```

Q6. Super Pow

```
class Solution {
private:
    int modPow(int a, int b, int mod) {
        int result = 1;
        a %= mod;
        while (b > 0) {
            if (b % 2) result = (result * a) % mod;
            a = (a * a) % mod;
            b /= 2;
```

```
        }
        return result;
    }
public:
    int superPow(int a, vector<int>& b) {
        int mod = 1337, exp = 0;
        for (int digit : b) exp = (exp * 10 + digit) % 1140;
        return modPow(a, exp, mod);
    }
};
```