

**Name : Aditya Kumar Ishwar**

**UID : 22BCS14310**

**SEC : IOT-605(b)**

Ques no. 1 :- Longest Nice Substring (1763)

Solution :-

```
class Solution {
public:
    string longestNiceSubstring(string s) {
        string output = "";
        int count = 0;
        for(int i = 0; i < s.length(); i++){
            int smallMask = 0;
            int largeMask = 0;
            char ch = s[i];
            int chint = 0;
            if(ch >= 65 && ch <= 90){
                chint = ch - 'A';
                largeMask = 1 << chint;
            }
            else{
                chint = ch - 'a';
                smallMask = 1 << chint;
            }
            for(int j = i + 1; j < s.length(); j++){
                ch = s[j];
                if(ch >= 65 && ch <= 90){
                    chint = ch - 'A';
```

```

        largeMask |= 1<<chint;
    }
    else{
        chint = ch-'a';
        smallMask |= 1<<chint;
    }
    //checking for nice
    if((smallMask^largeMask) == 0){
        if(count<j-i+1){
            count = j-i+1;
            string temp(s.begin()+i,s.begin()+j+1);
            output = temp;
        } } }
    return output;}
};

```

☒ Testcase
 ☒ Test Result

Accepted

Runtime: 0 ms

• Case 1

• Case 2

• Case 3

Input

s =  
 "YazaAay"

Output

"aAa"

Expected

"aAa"

[!\[\]\(6af8fb3374762cb7dc918a112e102b36\_img.jpg\)
 Contribute a testcase](#)

Ques no. 2 :- Reverse Bits (190)

Solution :-

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

Testcase | Test Result

**Accepted** Runtime: 0 ms

• Case 1 • Case 2

**Input**

n =  
00000010100101000001111010011100

**Output**

964176192 (00111001011110000010100101000000)

**Expected**

964176192 (00111001011110000010100101000000)

[Contribute a testcase](#)

Ques no. 3:- Number of 1 Bits(191)

Solution :-

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

The screenshot shows a test result interface with a dark theme. At the top, there are two tabs: 'Testcase' and 'Test Result', with 'Test Result' being the active tab. Below the tabs, the status 'Accepted' is displayed in green, followed by 'Runtime: 0 ms'. There are three buttons labeled 'Case 1', 'Case 2', and 'Case 3', with 'Case 1' being the selected one. Under the 'Input' section, there is a text box showing 'n =' followed by the value '11'. Under the 'Output' section, there is a text box showing the value '3'. Under the 'Expected' section, there is a text box showing the value '3'. At the bottom of the interface, there is a link that says 'Contribute a testcase' with a heart icon.

Testcase > Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

n =  
11

Output

3

Expected

3

♥ Contribute a testcase

Ques no.4 :- Maximum Subarray (52)

Solution :- class Solution {

public:

```
int maxSubArray(vector<int>& nums) {  
    int cs=0;  
    int ms= INT_MIN;  
    for(int i=0; i<nums.size(); i++){  
        cs += nums[i];  
        ms = max(cs , ms);  
        if (cs <0){  
            cs=0;  
        }  
    } return ms;  
};
```

Testcase

Test Result

Accepted

Runtime: 0 ms

Case 1

Case 2

Case 3

Input

nums =  
[-2,1,-3,4,-1,2,1,-5,4]

Output

6

Expected

6

♥

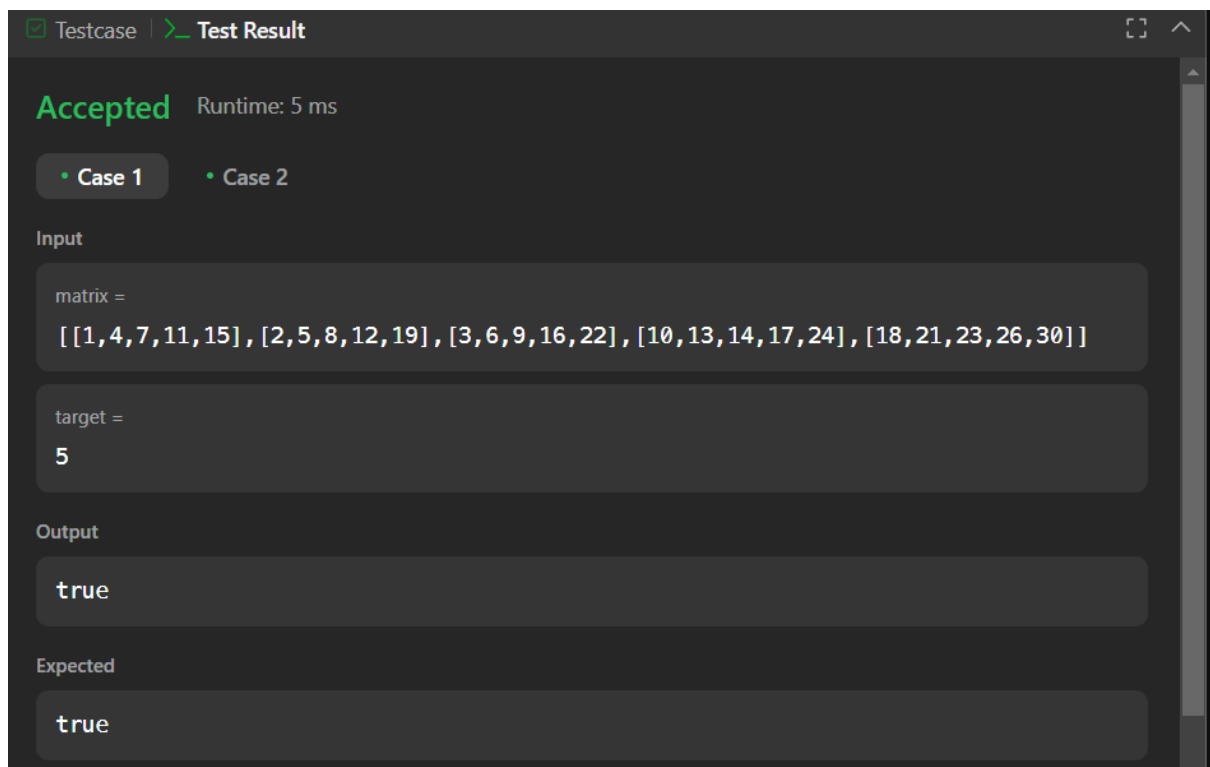
Contribute a testcase

Ques no. 5 :- Search a 2D Matrix II (240)

Solution :-

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int n = matrix.size(), m = matrix[0].size();
        int row = 0, col = m - 1;

        while (row < n && col >= 0) {
            if (matrix[row][col] == target) return true;
            else if (matrix[row][col] < target) row++;
            else col--;
        }
        return false;
    }
};
```



Ques no. 6 :- Super Pow(372)

Solution :-

```
class Solution {  
    const int base = 1337;  
  
    int powmod(int a, int k) //a^k mod 1337 where 0 <= k <= 10{  
        a %= base;  
  
        int result = 1;  
  
        for (int i = 0; i < k; ++i)  
            result = (result * a) % base;  
  
        return result; }  
  
public:  
  
    int superPow(int a, vector<int>& b) {  
        if (b.empty()) return 1;  
  
        int last_digit = b.back();  
  
        b.pop_back();  
  
        return powmod(superPow(a, b), 10) * powmod(a, last_digit) % base;  
    }  
};
```

The screenshot shows a 'Test Result' window with a dark theme. At the top, it says 'Accepted' in green and 'Runtime: 0 ms'. Below this, there are three tabs: 'Case 1' (selected), 'Case 2', and 'Case 3'. Under the 'Case 1' tab, the 'Input' section shows 'a =' followed by a text box containing '2', and 'b =' followed by a text box containing '[3]'. The 'Output' section shows a text box containing '8'. The 'Expected' section shows a text box containing '8'. At the bottom, there is a link that says 'Contribute a testcase' with a heart icon.