



**CHANDIGARH  
UNIVERSITY**  
Discover. Learn. Empower.

**NAAC  
GRADE A+**  
Accredited University

## **UNIVERSITY INSTITUTE OF ENGINEERING**

**Department of Computer Science & Engineering**

**(BE-CSE/IT-6<sup>th</sup> Sem)**



**Subject Name:** Advanced Programming Lab - 2

**Subject Code:** 22CSP-351

**Submitted to:**

Mr. Vishal

**Submitted by:**

Name: Sanya Singh

UID: 23BCS14374

Section: FL\_IOT\_604

Group: A

## ASSIGNMENT-4

1763.[Longest Nice Substring](#)

```
class Solution {  
  
public:  
  
bool check(string s){  
    for(int i=0;i<s.size();i++) {  
        char c = s[i];  
        if(c<=90) {  
            c += 32;}  
        else {  
            c -= 32;}  
        if(s.find(c)==string::npos) return false;  
    }  
    return true;  
}  
  
string longestNiceSubstring(string s) {  
    string ans = "";  
    for(int i=0;i<s.size();i++){  
        string res = "";  
        res += s[i];  
        for(int j = i+1;j<s.size();j++){  
            res += s[j];  
            if(check(res) and res.size()>ans.size()) ans = res;  
        }  
    }  
    return ans;  
}  
};
```

**Accepted** 73 / 73 testcases passed  
 Sanya Singh submitted at Feb 05, 2025 20:46

**Runtime**  
 22 ms | Beats 21.14%

**Memory**  
 14.52 MB | Beats 22.83%

**Code**

```

1 class Solution {
2 public:
3 bool check(string s){
4     for(int i=0;i<s.size();i++) {
5         char c = s[i];
6         if(c<=90) {
7             c += 32;}
8         else {
9             c -= 32;}
10        if(s.find(c)==string::npos) return false;
11    }
12    return true;
13 }
14

```

**Testcase** | Test Result

Case 1 Case 2 Case 3 +

s =

"YazaAay"

## 190. [Reverse Bits](#)

```

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

```

The screenshot displays a LeetCode submission page for the problem "Number of 1 Bits". The submission is marked as "Accepted" with 600/600 testcases passed. The runtime is 0 ms, beating 100.00% of solutions, and the memory usage is 7.79 MB, beating 62.75%. A bar chart shows the runtime distribution across different time intervals. The code editor on the right shows a C++ solution using a loop to count the number of 1s in the binary representation of the input number.

**Runtime:** 0 ms | Beats 100.00%  
**Memory:** 7.79 MB | Beats 62.75%

**Code:**

```

1 class Solution {
2 public:
3     uint32_t reverseBits(uint32_t n) {
4         uint32_t res = 0;
5         for (int i = 0; i < 32; i++) {
6             res = res << 1;
7             res += (n & 1);
8             n = n >> 1;
9         }
10        return res;
11    }
12 };

```

**Testcase:** Runtime: 2 ms  
 Accepted  
 Case 1 Case 2  
 Input: n =

## 191. [Number of 1 Bits](#)

```

class Solution {
public:
    int hammingWeight(int n) {
        int count = 0;
        for(int i = 31; i >= 0; i--){
            if(((n >> i) & 1) == 1)
                count++;
        }
        return count;
    }
};

```

**Accepted** 598 / 598 testcases passed  
 Submitted at Feb 05, 2025 20:59

**Runtime**  
 0 ms | Beats 100.00%  
 Analyze Complexity

**Memory**  
 8.19 MB | Beats 80.44%

**Code**

```

1 class Solution {
2 public:
3     int maxSubArray(vector<int>& nums) {
4         int sum = 0;
5         int maxi = nums[0];
6         int n = nums.size();
7         for(int i = 0; i < n; i++){
8             sum = sum + nums[i];
9             maxi = max(sum, maxi);
10            if(sum < 0){
11                sum = 0;
12            }
13        }
14        return maxi;
15    }
16 };
  
```

**Testcase** | **Test Result**

**Accepted** Runtime: 0 ms

Case 1 Case 2 Case 3

Input

n =

11

### 53. Maximum Subarray

```

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

The screenshot displays a C++ code editor interface. On the left, a sidebar shows the submission status: 'Accepted' with 210/210 testcases passed. It also shows the runtime as 0 ms (Beats 100.00%) and memory usage as 70.65 MB (Beats 99.93%). The main editor area shows a C++ code snippet for a 'maxSubArray' function. The bottom panel shows the 'Testcase' and 'Test Result' sections, both indicating 'Accepted'.

```

1 class Solution {
2 public:
3     int maxSubArray(vector<int>& nums) {
4         int sum=0;
5         int maxi=nums[0];
6         int n=nums.size();
7         for(int i=0;i<n;i++){
8             sum=sum+nums[i];
9             maxi=max(sum,maxi);
10            if(sum<0){
11                sum=0;
12            }
13        }
14        return maxi;
15    }
16 };

```

## 240. [Search a 2D Matrix II](#)

```
class Solution {
```

```
public:
```

```
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
```

```
        int row=matrix.size();
```

```
        int col=matrix[0].size();
```

```
        int rowindex=0;
```

```
        int colindex=col-1;
```

```
        while(rowindex<row && colindex>=0){
```

```
            int element=matrix[rowindex][colindex];
```

```
            if(target==element){
```

```
                return 1;
```

```
            }
```

```
            if(element<target){
```

```
                rowindex++;
```

```

    }

    else{

        colindex--;

    }

}

return 0;

}

};

```

The screenshot displays a C++ IDE interface. On the left, a sidebar indicates the solution is 'Accepted' with 130/130 testcases passed. It shows a runtime of 57 ms (39.04% beats) and memory usage of 17.37 MB (100.00% beats). The main editor shows a C++ code snippet for a searchMatrix function. The bottom panel shows 'Testcase' and 'Test Result' for 'Case 1' and 'Case 2', both marked as 'Accepted' with a runtime of 3 ms.

372. [Super Pow](#)

```

class Solution {
    const int base = 1337;

    int powmod(int a, int k)
    {
        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 displays a C++ code editor interface. The top bar includes navigation icons, a 'Problem List' tab, and buttons for 'Run', 'Submit', and 'Premium'. The main editor area shows the following C++ code:

```
10 }  
11 public:  
12  
13 int superPow(int a, vector<int>& b) {  
14     if (b.empty()) return 1;  
15     int last_digit = b.back();  
16     b.pop_back();  
17     return powmod(superPow(a, b), 10) * powmod(a, last_digit) % base;  
18 }  
19 };
```

Below the code editor, the 'Testcase' tab is active, showing 'Accepted' status and 'Runtime: 0 ms'. The 'Test Result' section displays three cases: Case 1, Case 2, and Case 3. The 'Input' field shows 'a = 2'.

On the left side, the 'All Submissions' tab is active, showing 'Accepted 57 / 57 testcases passed' and 'Solution' button. The submission details for 'Sanya Singh' (submitted at Feb 05, 2025 21:02) are shown. The 'Runtime' section indicates '3 ms | Beats 31.12%' and 'Analyze Complexity' link. The 'Memory' section shows '15.25 MB | Beats 52.85%'. A bar chart below the memory section shows the distribution of runtime across different time intervals (0ms to 10ms).