Assingment -4

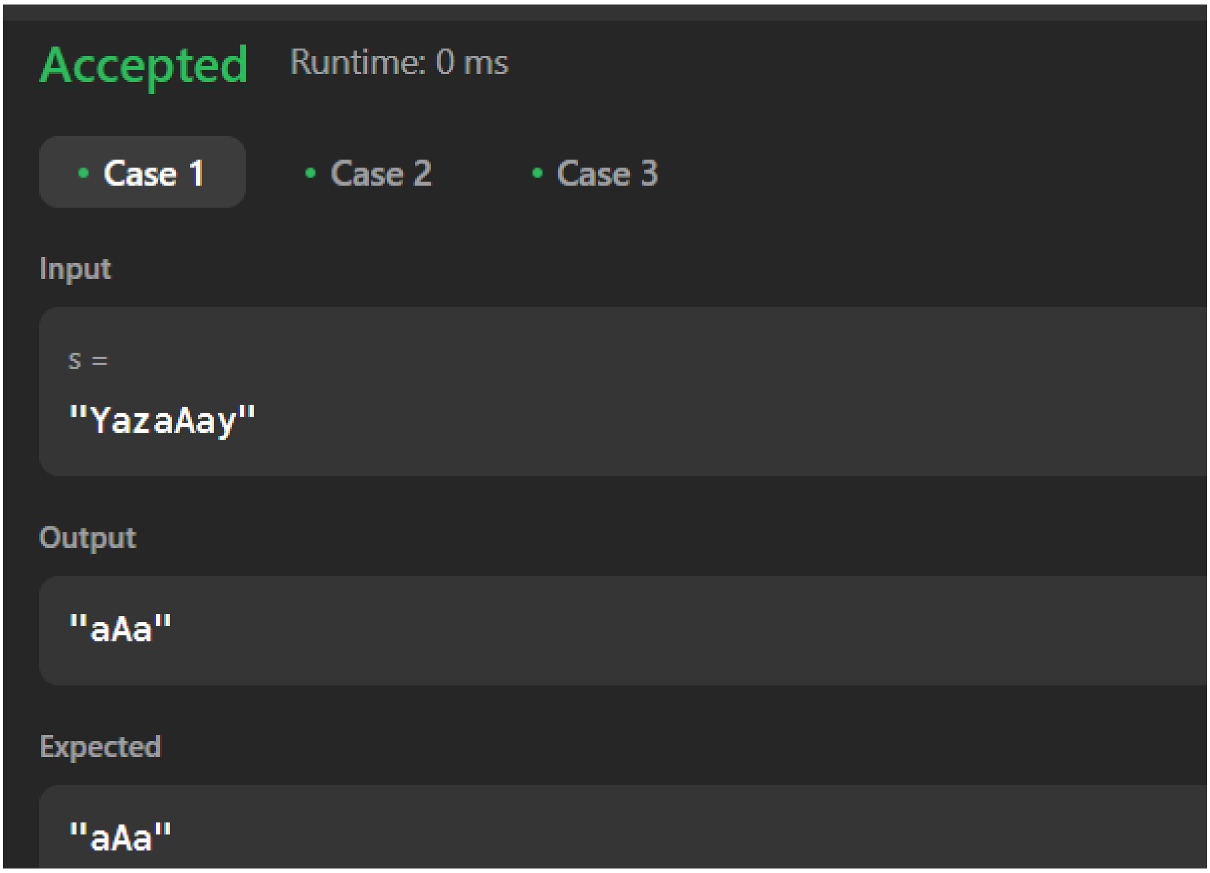
Name- Akul UID-22bcs10330 Section- 605-B

1763. Longest Nice Substring

Code:

```
class Solution
{
public:
string longestNiceSubstring(string s) {
        if (s.size() < 2) return "";
        unordered_set<char> st(begin(s), end(s));
        for (int i = 0; i < s.size(); i++) {
             if (st.find((char) toupper(s[i])) == end(st) || st.find((char)
tolower(s[i])) == end(st)) {
                string s1 = longestNiceSubstring(s.substr(0, i));
                string s2 = longestNiceSubstring(s.substr(i + 1));
                return s1.size() >= s2.size() ? s1 : s2;
            }
        }
        return s;
    }
};
```

Output:



Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

s =

"YazaAay"

Output

"aAa"

Expected

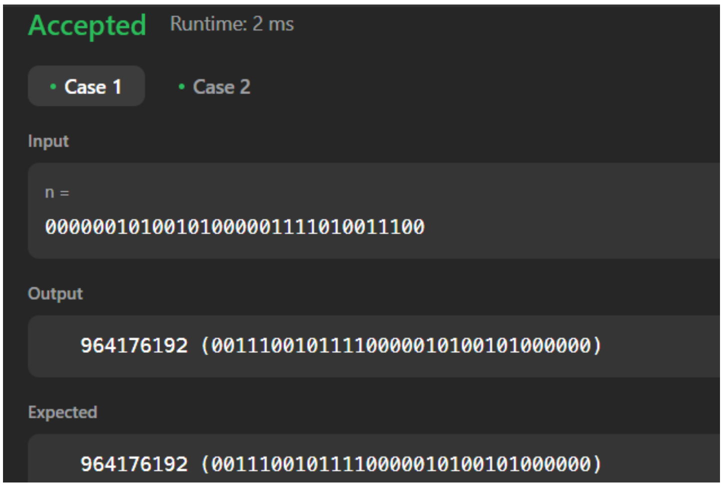
"aAa"

190. Reverse Bits

Code:

```
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;
    }
};
```

Output:



Input

```
n =
00000010100101000001111010011100
```

Output

```
964176192 (00111001011110000010100101000000)
```

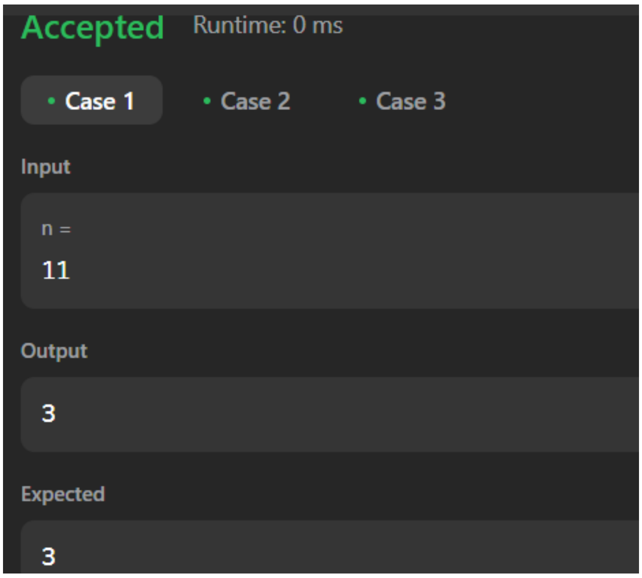
Expected

```
964176192 (00111001011110000010100101000000)
```

191. Number of 1 Bits Code:

```
class Solution {
public:
    int hammingWeight(int n) {
        return __builtin_popcount(n);
    }
};
```

OUTPUT:



Input

```
n =
11
```

Output

```
3
```

Expected

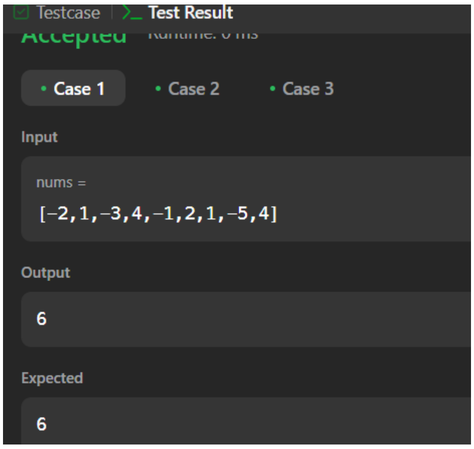
```
3
```

53. Maximum Subarray

Code:

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int n = size(nums), ans = INT_MIN;
        for(int i = 0; i < n; i++)
            for(int j = i, curSum = 0; j < n ; j++)
                curSum += nums[j],
                ans = max(ans, curSum);
        return ans;
    }
};
```

OUTPUT:



Accepted Runtime: 0 ms

Input

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

Output

```
6
```

Expected

```
6
```

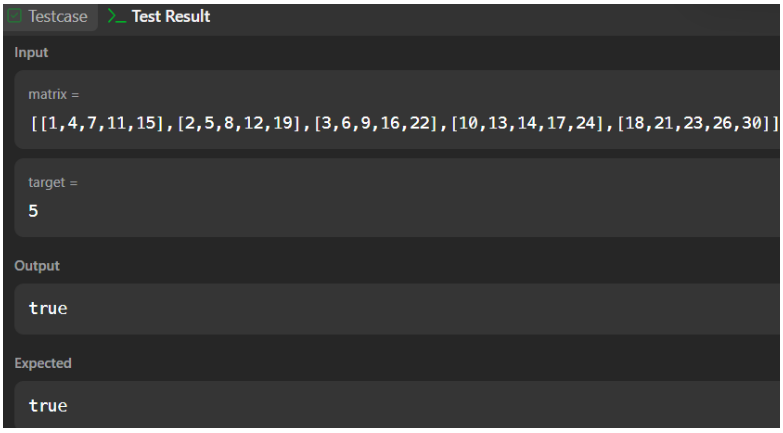
240. Search a 2D Matrix II

Code:

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int row = matrix.size();
        int col = matrix[0].size();
        //top-right conrer
        int i = 0;
        int j = col-1;

        while(i <= row-1 && j >= 0){
            if(matrix[i][j] == target)  return true;
            else if(matrix[i][j] < target) i++;
            else j--;
        }
        return false;
    }
};
```

Output:



Input

matrix =

```
[[1,4,7,11,15],[2,5,8,12,19],[3,6,9,16,22],[10,13,14,17,24],[18,21,23,26,30]]
```

target =

```
5
```

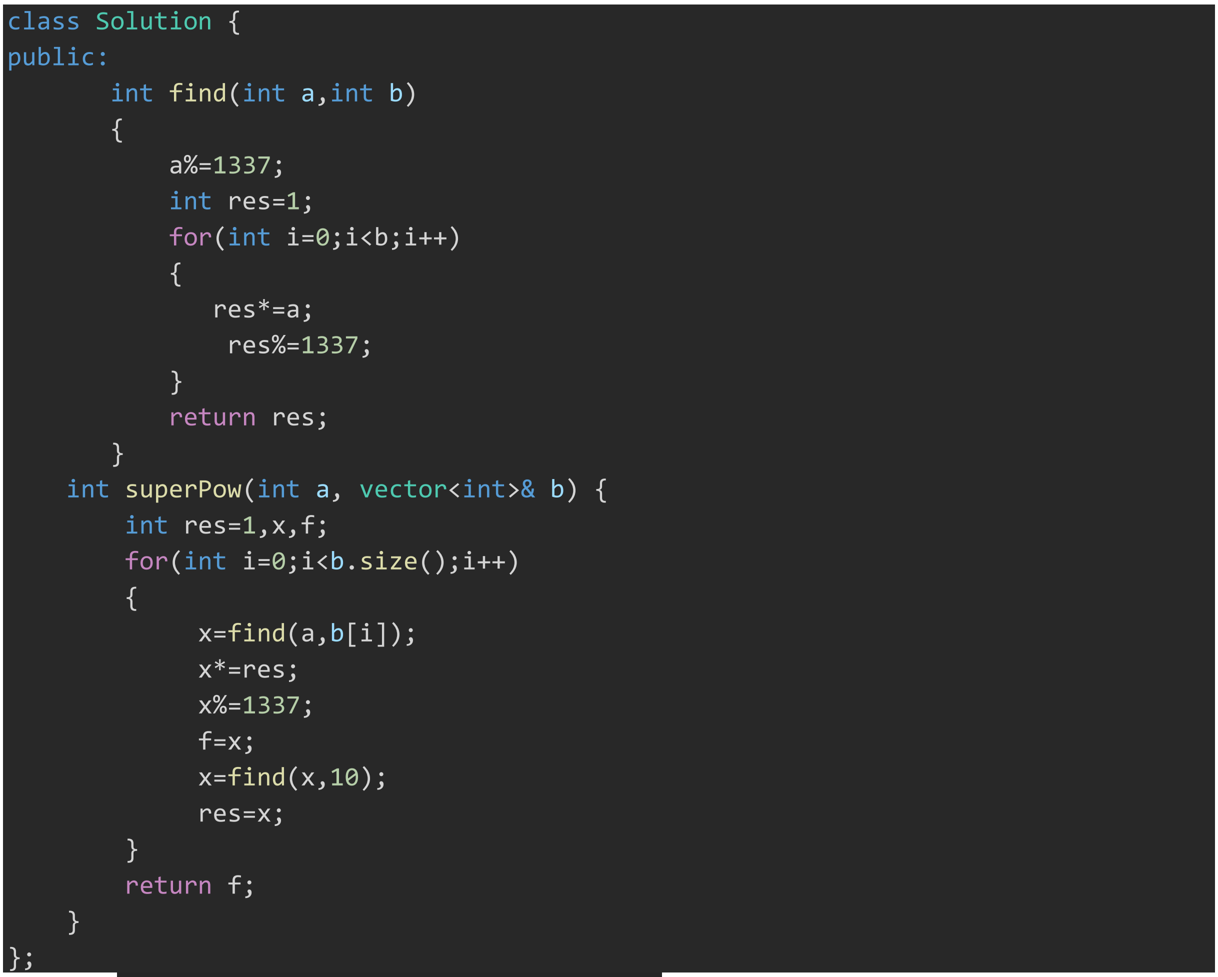
Output

```
true
```

Expected

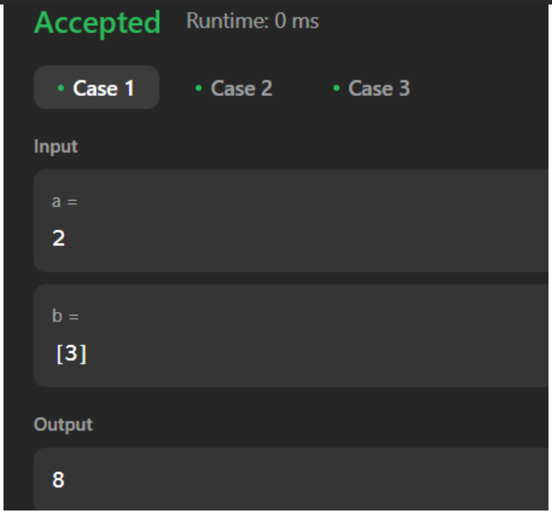
```
true
```

372. Super Pow Code:



```
class Solution {
public:
       int find(int a,int b)
       {
           a%=1337;
           int res=1;
           for(int i=0;i<b;i++)
           {
              res*=a;
               res%=1337;
           }
           return res;
       }
    int superPow(int a, vector<int>& b) {
        int res=1,x,f;
        for(int i=0;i<b.size();i++)
        {
             x=find(a,b[i]);
             x*=res;
             x%=1337;
             f=x;
             x=find(x,10);
             res=x;
        }
        return f;
    }
};
```

Output:



Accepted Runtime: 0 ms

Input

a =

```
2
```

b =

```
[3]
```

Output

```
8
```