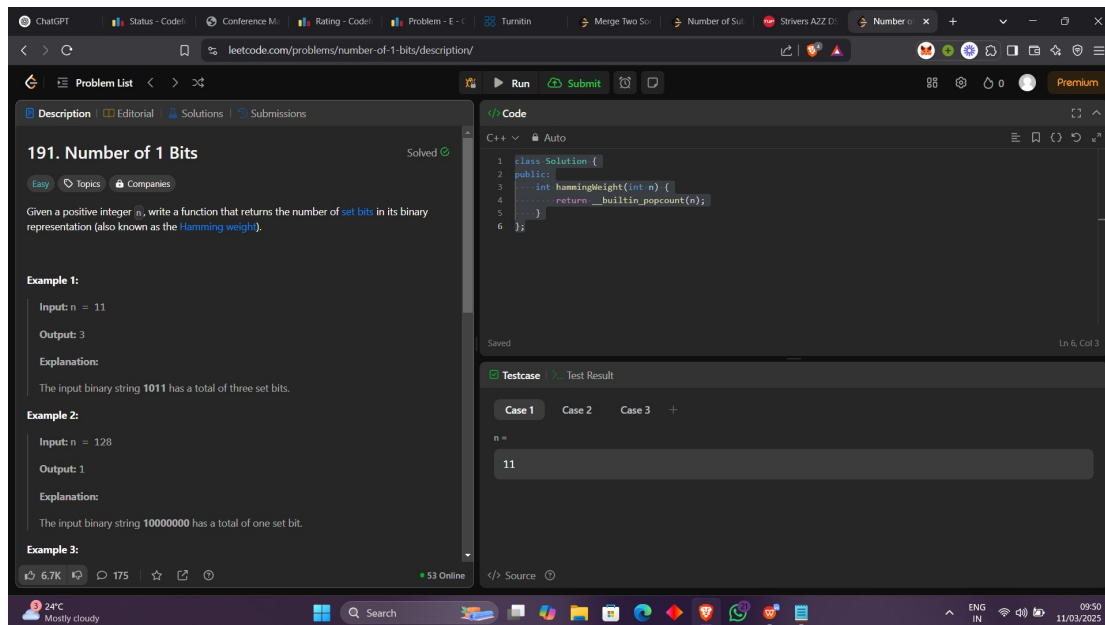


1763. Longest Nice Substring

```
class Solution {
public:
    string longestNiceSubstring(string s) {
        string ans="";
        for(int i=0;i<s.length();i++){
            int count=0;
            string temp="";
            vector<bool> a(26,0),b(26,0);
            for(int j=i;j<s.length();j++){
                temp.push_back(s[j]);
                if(s[j]>='A' && s[j]<='Z'){
                    if(a[s[j]-'A']==0 && b[s[j]-'A']==0)
                        count++;
                    else if(a[s[j]-'A'] && !b[s[j]-'A'])
                        count--;
                    b[s[j]-'A']=1;
                }
                else{
                    if(b[s[j]-'a']==0 && a[s[j]-'a']==0)
                        count++;
                    else if(b[s[j]-'a'] && !a[s[j]-'a'])
                        count--;
                    a[s[j]-'a']=1;
                }
                if(ans.size()<temp.size() && count==0)
                    ans=temp;
            }
        }
        return ans;
    }
};
```

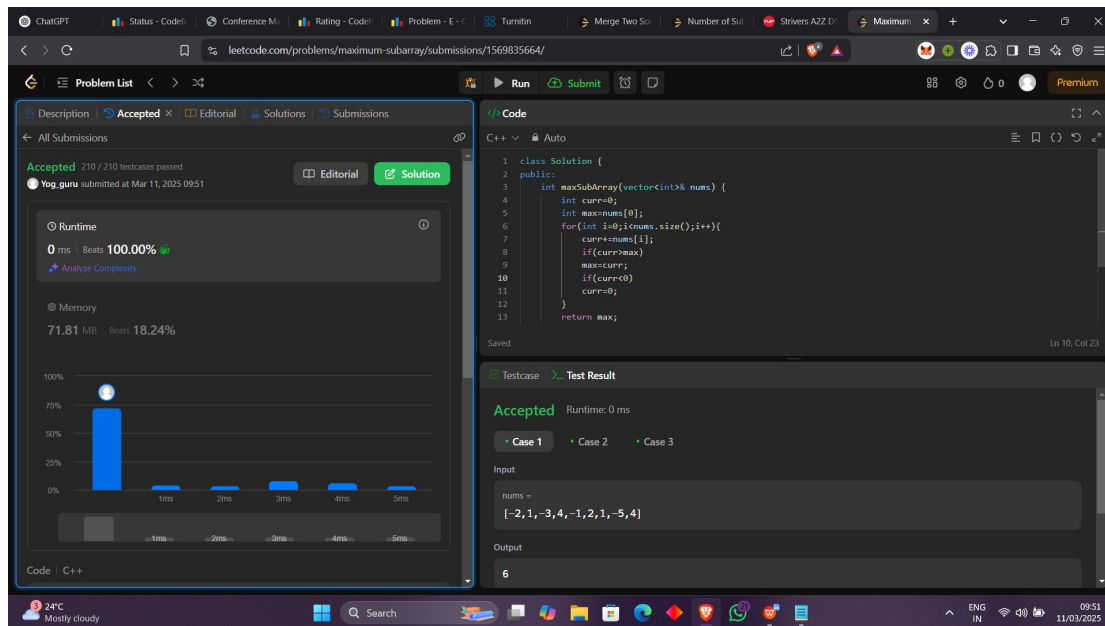

191. Number of 1 Bits

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



53. Maximum Subarray

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



240. Search a 2D Matrix II

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        for(int i=0;i<matrix.size();i++){
            if(matrix[i][0]<=target && matrix[i][matrix[i].size()-1]>=target){
                int start=0;
                int end=matrix[i].size()-1;
                while(start<=end){
                    int mid=start+(end-start)/2;
                    if(matrix[i][mid]==target)
                        return true;
                    if(matrix[i][mid]>target)
                        end=mid-1;
                    else
                        start=mid+1;
                }
            }
        }
        return false;
    }
};
```

The screenshot displays a LeetCode submission for the problem "Search a 2D Matrix II". The submission is by user "Yog_guru" and is marked as "Accepted". The solution is written in C++ and uses a binary search approach. The test results show that the solution passed all 130 test cases with a runtime of 60ms and a memory usage of 18.81MB. The code is as follows:

```

class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        for(int i=0; i<matrix.size(); i++){
            if(matrix[i][0]<target && matrix[i][matrix[i].size()-1]>target){
                int start=0;
                int end=matrix[i].size()-1;
                while(start<end){
                    int mid=start+(end-start)/2;
                    if(matrix[i][mid]==target)
                        return true;
                    if(matrix[i][mid]>target)
                        end=mid-1;
                }
            }
        }
        return false;
    }
};

```

372. Super Pow

```

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;
    }
};

```

ChatGPT | Status - Code | Conference Mo | Rating - Code | Problem - E - C | Turnitin | Merge Two Su | Number of Su | Strivers A2Z D | Super Pow | + | - | x |

leetcode.com/problems/super-pow/submissions/1569837449/

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 57 / 57 testcases passed
Yog_guru submitted at Mar 11, 2025 09:53 [Solution](#)

Runtime
0 ms | Beats: 100.00% [Analyze Complexity](#)

Memory
15.32 MB | Beats: 13.97%

75%
50%
25%
0%

1ms 2ms 3ms 4ms 5ms 6ms 7ms

Code C++

```
1 class Solution {
2     const int base = 1337;
3     int powmod(int a, int k) // a^k mod 1337 where 0 <= k <= 10
4     {
5         a %= base;
6         int result = 1;
7         for (int i = 0; i < k; ++i)
8             result = (result * a) % base;
9         return result;
10    }
11    public:
12    int superPow(int a, vector<int>& b) {
13        if (b.empty()) return 1;
```

Testcase Test Result

Case 1 Case 2 Case 3 +

a =
2

b =
[3]

Source

24°C Mostly cloudy | Search | ENG IN | 09:53 11/03/2025