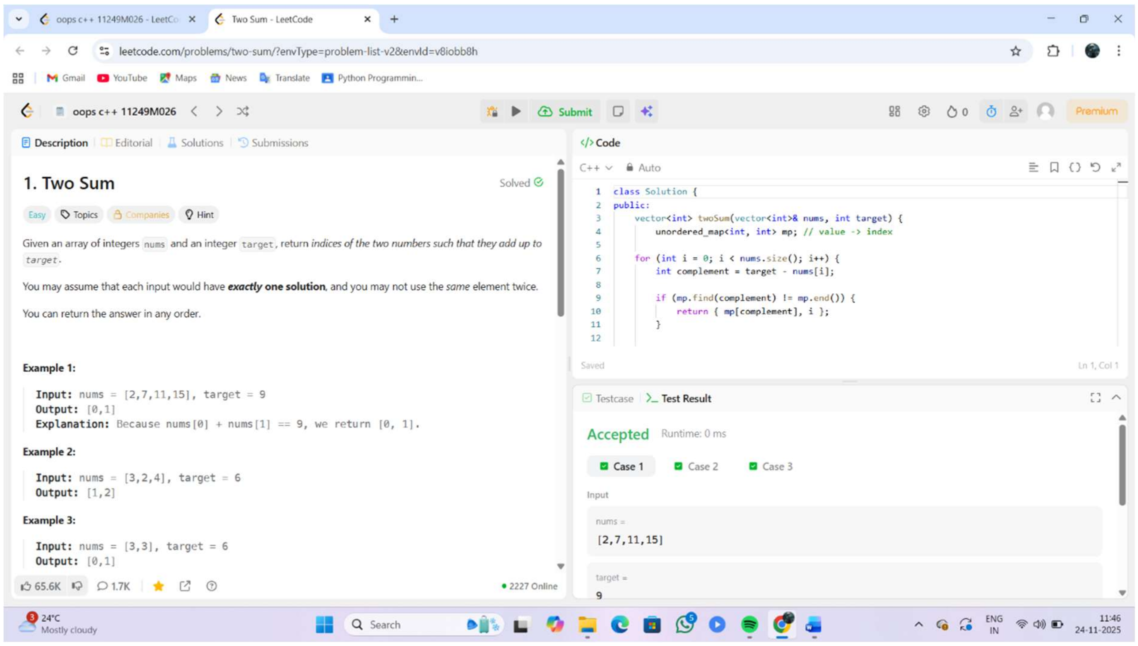
OOPS C++ LEETCODE PROBLEMS

NAME:LITHIKA V

REG.NO:11249M026

1.TWO SUM



leetcode.com/problems/two-sum/?envType=problem-list-v2&envId=v8iobb8h

1. Two Sum

Solved

Easy · Topics · Companies · Hint

Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to `target`*.

You may assume that each input would have ***exactly* one solution**, and you may not use the *same* element twice.

You can return the answer in any order.

Example 1:

```
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].
```

Example 2:

```
Input: nums = [3,2,4], target = 6
Output: [1,2]
```

Example 3:

```
Input: nums = [3,3], target = 6
Output: [0,1]
```

Code — C++

```cpp
class Solution {
public:
    vector<int> twoSum(vector<int>& nums, int target) {
        unordered_map<int, int> mp; // value -> index

        for (int i = 0; i < nums.size(); i++) {
            int complement = target - nums[i];

            if (mp.find(complement) != mp.end()) {
                return { mp[complement], i };
            }

```

Test Result

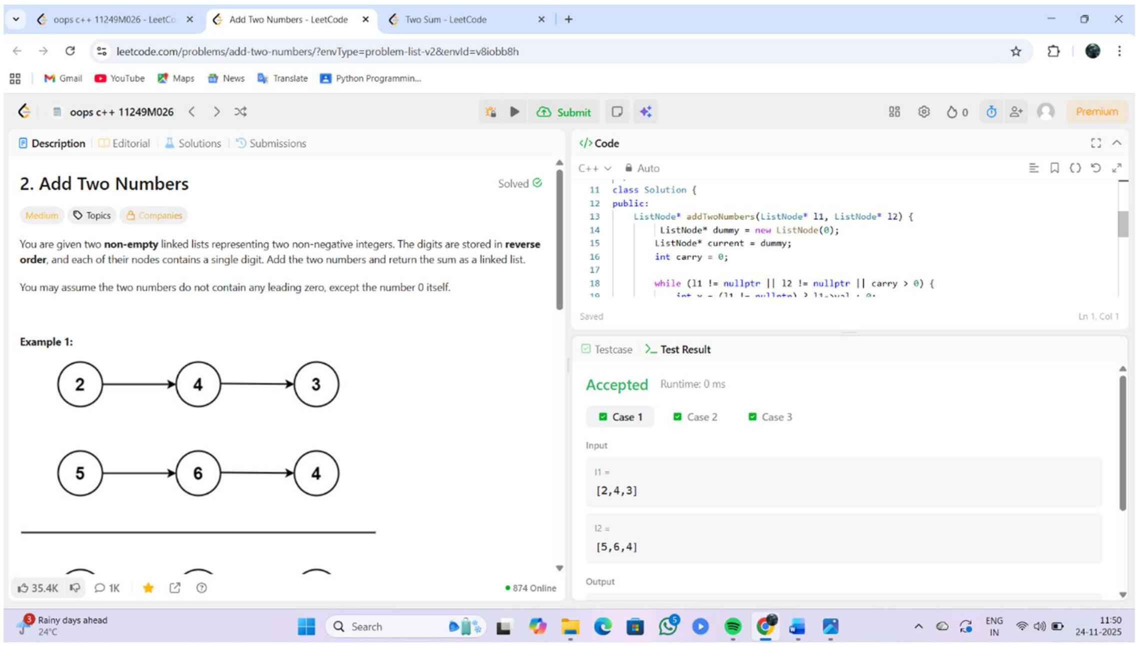
Accepted Runtime: 0 ms

Case 1 · Case 2 · Case 3

Input

```
nums =
[2,7,11,15]
target =
9
```

2.ADD TWO NUMBERS



leetcode.com/problems/add-two-numbers/?envType=problem-list-v2&envId=v8iobb8h

2. Add Two Numbers

Solved

Medium · Topics · Companies

You are given two **non-empty** linked lists representing two non-negative integers. The digits are stored in **reverse order**, and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

Example 1:

Code — C++

```cpp
class Solution {
public:
    ListNode* addTwoNumbers(ListNode* l1, ListNode* l2) {
        ListNode* dummy = new ListNode(0);
        ListNode* current = dummy;
        int carry = 0;

        while (l1 != nullptr || l2 != nullptr || carry > 0) {
```

Test Result

Accepted Runtime: 0 ms

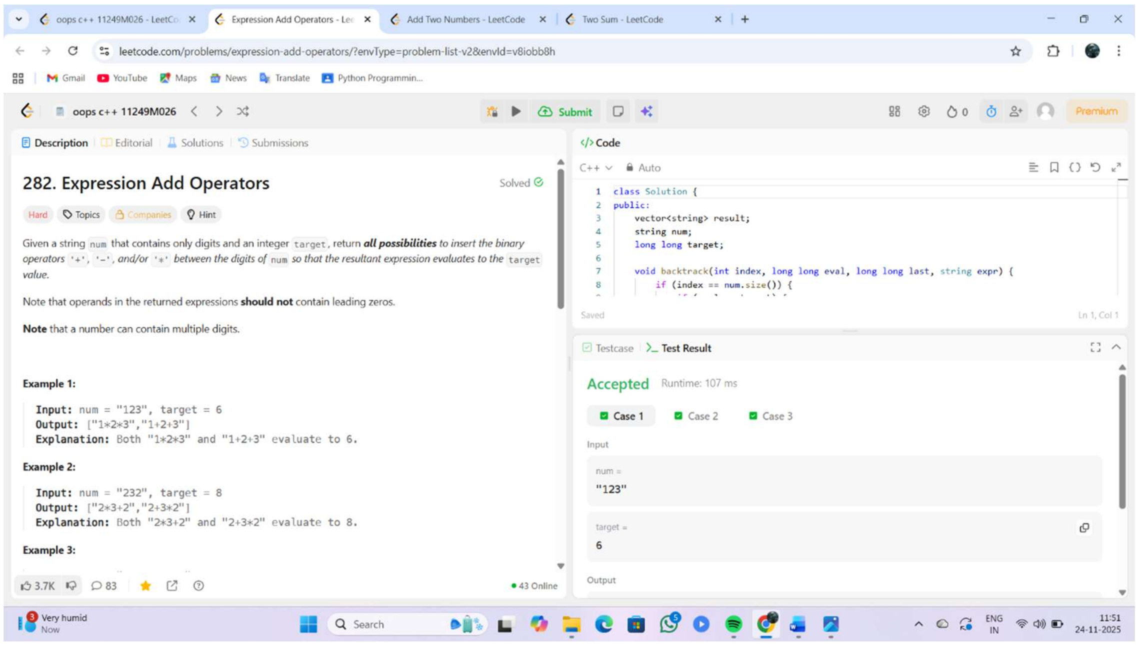
Case 1 · Case 2 · Case 3

Input

```
l1 =
[2,4,3]
l2 =
[5,6,4]
```

Output

3.EXPRESSION ADD OPERATORS



282. Expression Add Operators

Solved

Hard · Topics · Companies · Hint

Given a string `num` that contains only digits and an integer `target`, return ***all possibilities*** *to insert the binary operators* `'+'`, `'-'`, *and/or* `'*'` *between the digits of* `num` *so that the resultant expression evaluates to the* `target` *value.*

Note that operands in the returned expressions **should not** contain leading zeros.

Note that a number can contain multiple digits.

Example 1:

```
Input: num = "123", target = 6
Output: ["1*2*3","1+2+3"]
Explanation: Both "1*2*3" and "1+2+3" evaluate to 6.
```

Example 2:

```
Input: num = "232", target = 8
Output: ["2*3+2","2+3*2"]
Explanation: Both "2*3+2" and "2+3*2" evaluate to 8.
```

Example 3:

3.7K · 83 · 43 Online

```cpp
class Solution {
public:
    vector<string> result;
    string num;
    long long target;

    void backtrack(int index, long long eval, long long last, string expr) {
        if (index == num.size()) {
```

Accepted Runtime: 107 ms

Case 1 · Case 2 · Case 3

Input

num =

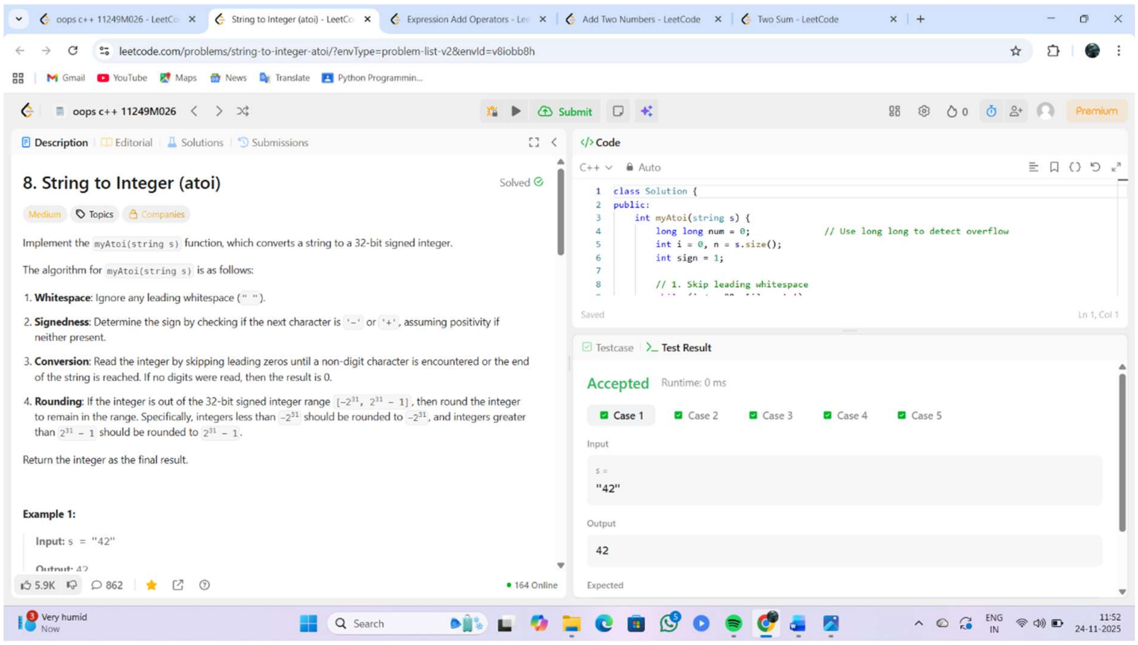
"123"

target =

6

Output

4.STRING TO INTEGER



8. String to Integer (atoi)

Solved

Medium · Topics · Companies

Implement the `myAtoi(string s)` function, which converts a string to a 32-bit signed integer.

The algorithm for `myAtoi(string s)` is as follows:

1. **Whitespace**: Ignore any leading whitespace (`" "`).
2. **Signedness**: Determine the sign by checking if the next character is `'-'` or `'+'`, assuming positivity if neither present.
3. **Conversion**: Read the integer by skipping leading zeros until a non-digit character is encountered or the end of the string is reached. If no digits were read, then the result is 0.
4. **Rounding**: If the integer is out of the 32-bit signed integer range $[-2^{31}, 2^{31} - 1]$, then round the integer to remain in the range. Specifically, integers less than -2^{31} should be rounded to -2^{31}, and integers greater than $2^{31} - 1$ should be rounded to $2^{31} - 1$.

Return the integer as the final result.

Example 1:

Input: s = "42"

5.9K · 862 · 164 Online

```cpp
class Solution {
public:
    int myAtoi(string s) {
        long long num = 0;           // Use long long to detect overflow
        int i = 0, n = s.size();
        int sign = 1;

        // 1. Skip leading whitespace
```

Accepted Runtime: 0 ms

Case 1 · Case 2 · Case 3 · Case 4 · Case 5

Input

s =

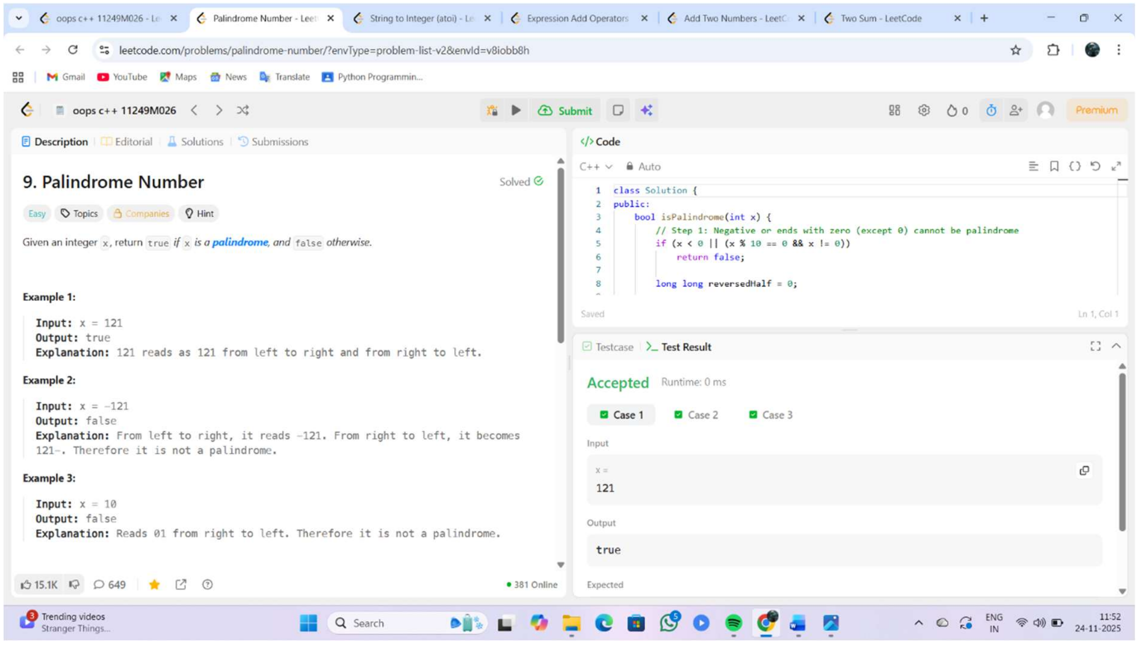
"42"

Output

42

Expected

5.PALINDROME NUMBER



leetcode.com/problems/palindrome-number/?envType=problem-list-v2&envId=v8iobb8h

oops c++ 11249M026

Description | Editorial | Solutions | Submissions

9. Palindrome Number

Solved

Easy · Topics · Companies · Hint

Given an integer `x`, return `true` *if* `x` *is a* ***palindrome****, and* `false` *otherwise.*

Example 1:

```
Input: x = 121
Output: true
Explanation: 121 reads as 121 from left to right and from right to left.
```

Example 2:

```
Input: x = -121
Output: false
Explanation: From left to right, it reads -121. From right to left, it becomes
121-. Therefore it is not a palindrome.
```

Example 3:

```
Input: x = 10
Output: false
Explanation: Reads 01 from right to left. Therefore it is not a palindrome.
```

15.1K · 649 · 381 Online

Code — C++ · Auto

```cpp
class Solution {
public:
    bool isPalindrome(int x) {
        // Step 1: Negative or ends with zero (except 0) cannot be palindrome
        if (x < 0 || (x % 10 == 0 && x != 0))
            return false;

        long long reversedHalf = 0;
```

Saved · Ln 1, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 · Case 2 · Case 3

Input

x =
121

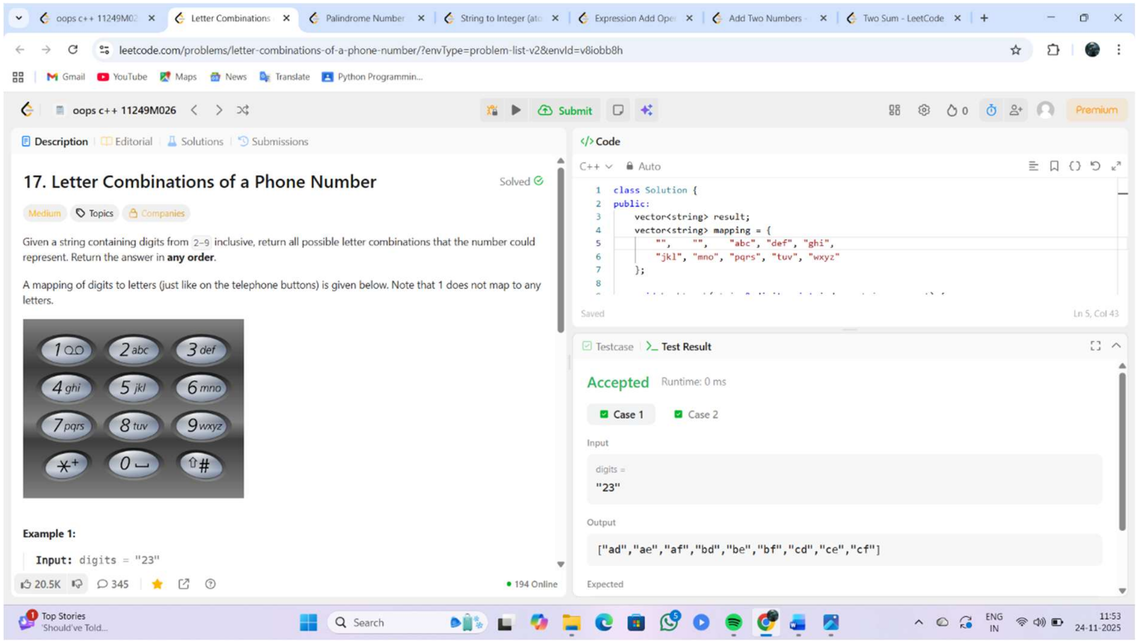
Output

true

Expected

11:52 24-11-2025

6.LETTER COMBINATIONS OF A PHONE NUMBER



leetcode.com/problems/letter-combinations-of-a-phone-number/?envType=problem-list-v2&envId=v8iobb8h

oops c++ 11249M026

Description | Editorial | Solutions | Submissions

17. Letter Combinations of a Phone Number

Solved

Medium · Topics · Companies

Given a string containing digits from `2-9` inclusive, return all possible letter combinations that the number could represent. Return the answer in **any order**.

A mapping of digits to letters (just like on the telephone buttons) is given below. Note that 1 does not map to any letters.

Example 1:

```
Input: digits = "23"
```

20.5K · 345 · 194 Online

Code — C++ · Auto

```cpp
class Solution {
public:
    vector<string> result;
    vector<string> mapping = {
        "",    "",    "abc", "def", "ghi",
        "jkl", "mno", "pqrs", "tuv", "wxyz"
    };
```

Saved · Ln 5, Col 43

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 · Case 2

Input

digits =
"23"

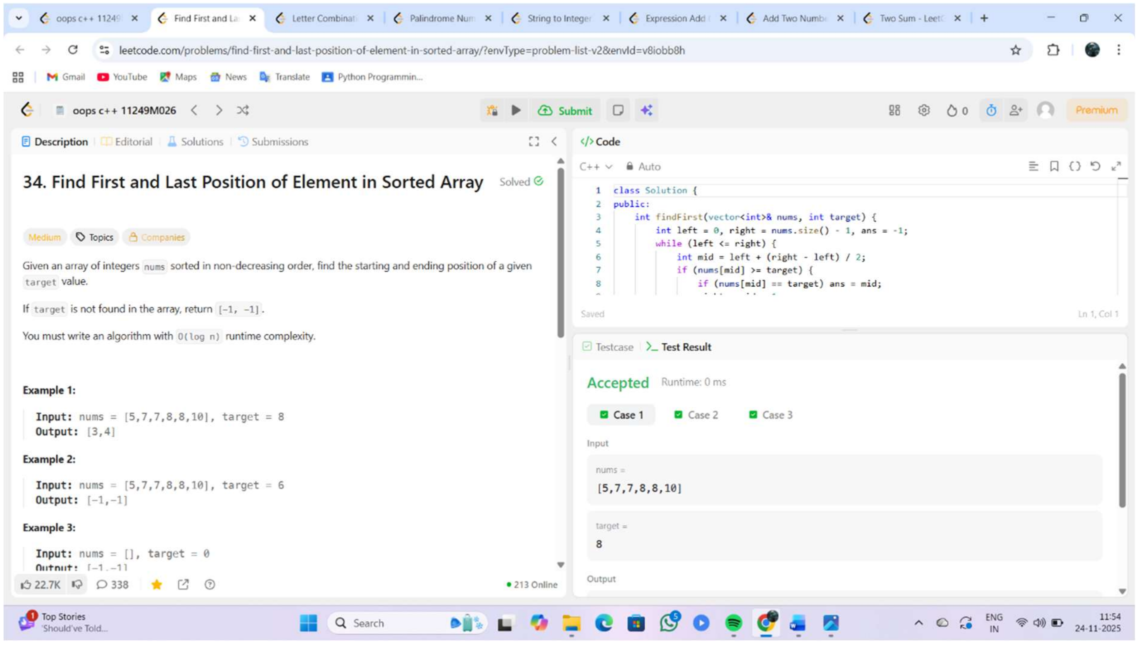
Output

```
["ad","ae","af","bd","be","bf","cd","ce","cf"]
```

Expected

11:53 24-11-2025

7.FINDFIRST AND LAST POSITION OF ELEMENT IN SORTED ARRAY



leetcode.com/problems/find-first-and-last-position-of-element-in-sorted-array/?envType=problem-list-v2&envId=v8iobb8h

34. Find First and Last Position of Element in Sorted Array Solved

Medium · Topics · Companies

Given an array of integers `nums` sorted in non-decreasing order, find the starting and ending position of a given `target` value.

If `target` is not found in the array, return `[-1, -1]`.

You must write an algorithm with `O(log n)` runtime complexity.

Example 1:

```
Input: nums = [5,7,7,8,8,10], target = 8
Output: [3,4]
```

Example 2:

```
Input: nums = [5,7,7,8,8,10], target = 6
Output: [-1,-1]
```

Example 3:

```
Input: nums = [], target = 0
```

```cpp
class Solution {
public:
    int findFirst(vector<int>& nums, int target) {
        int left = 0, right = nums.size() - 1, ans = -1;
        while (left <= right) {
            int mid = left + (right - left) / 2;
            if (nums[mid] >= target) {
                if (nums[mid] == target) ans = mid;
```

Accepted Runtime: 0 ms

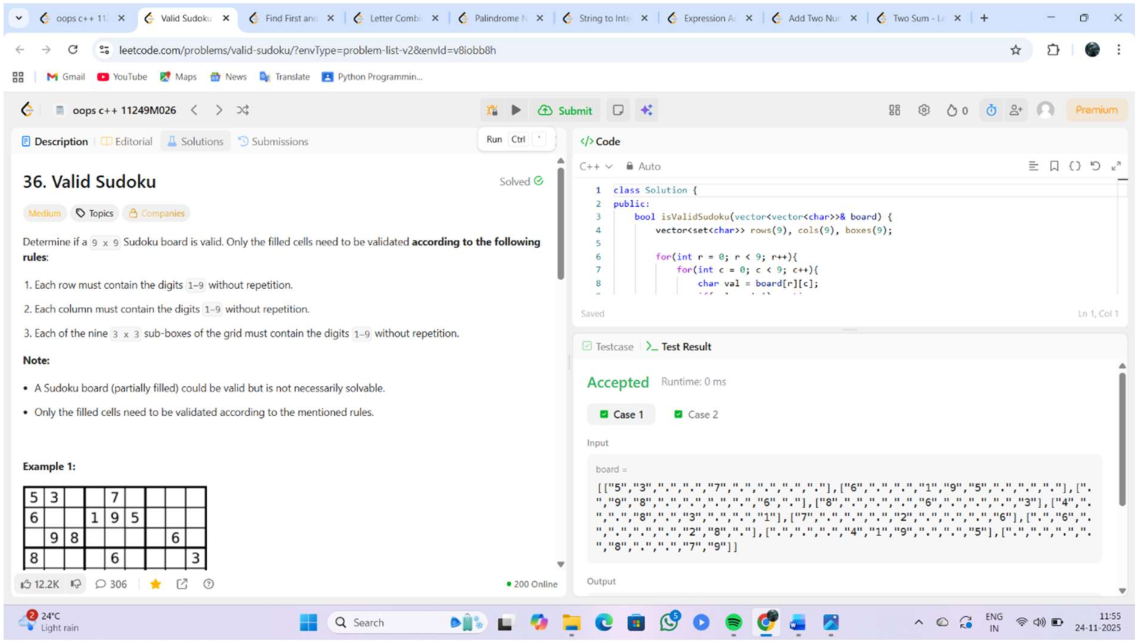
Case 1 · Case 2 · Case 3

Input

```
nums =
[5,7,7,8,8,10]
target =
8
```

Output

8.VALID SUDOKU



leetcode.com/problems/valid-sudoku/?envType=problem-list-v2&envId=v8iobb8h

36. Valid Sudoku Solved

Medium · Topics · Companies

Determine if a `9 x 9` Sudoku board is valid. Only the filled cells need to be validated **according to the following rules**:

1. Each row must contain the digits `1-9` without repetition.
2. Each column must contain the digits `1-9` without repetition.
3. Each of the nine `3 x 3` sub-boxes of the grid must contain the digits `1-9` without repetition.

Note:

- A Sudoku board (partially filled) could be valid but is not necessarily solvable.
- Only the filled cells need to be validated according to the mentioned rules.

Example 1:

```cpp
class Solution {
public:
    bool isValidSudoku(vector<vector<char>>& board) {
        vector<set<char>> rows(9), cols(9), boxes(9);

        for(int r = 0; r < 9; r++){
            for(int c = 0; c < 9; c++){
                char val = board[r][c];
```

Accepted Runtime: 0 ms

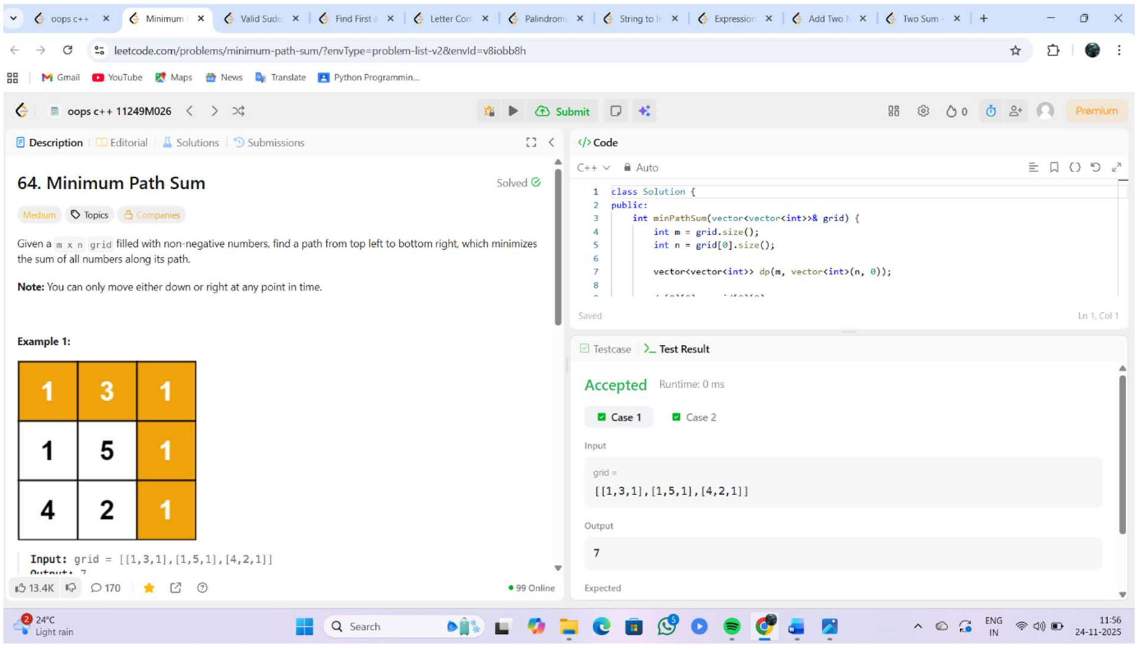
Case 1 · Case 2

Input

```
board =
[["5","3",".",".","7",".",".",".","."],["6",".",".","1","9","5",".",".","."],[".","9","8",".",".",".",".","6","."],["8",".",".",".","6",".",".",".","3"],["4",".",".","8",".","3",".",".","1"],["7",".",".",".","2",".",".",".","6"],[".","6",".",".",".",".","2","8","."],[".",".",".","4","1","9",".",".","5"],[".",".",".",".","8",".",".","7","9"]]
```

Output

9.MINIMUM PATH SUM



leetcode.com/problems/minimum-path-sum/?envType=problem-list-v2&envId=v8iobb8h

oops c++ 11249M026

64. Minimum Path Sum

Solved

Medium · Topics · Companies

Given a `m x n` `grid` filled with non-negative numbers, find a path from top left to bottom right, which minimizes the sum of all numbers along its path.

Note: You can only move either down or right at any point in time.

Example 1:

```
Input: grid = [[1,3,1],[1,5,1],[4,2,1]]
```

```
class Solution {
public:
    int minPathSum(vector<vector<int>>& grid) {
        int m = grid.size();
        int n = grid[0].size();

        vector<vector<int>> dp(m, vector<int>(n, 0));
```

Saved · Ln 1, Col 1

Test Result

Accepted Runtime: 0 ms

Case 1 · Case 2

Input

grid =

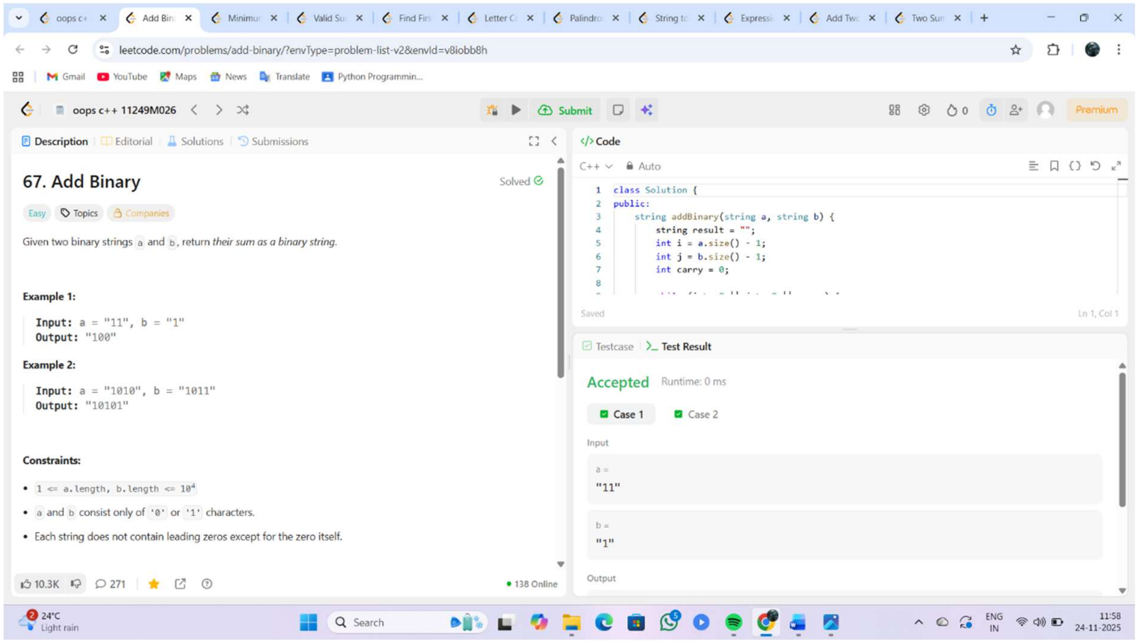
```
[[1,3,1],[1,5,1],[4,2,1]]
```

Output

```
7
```

Expected

10.ADD BINARY



leetcode.com/problems/add-binary/?envType=problem-list-v2&envId=v8iobb8h

oops c++ 11249M026

67. Add Binary

Solved

Easy · Topics · Companies

Given two binary strings `a` and `b`, return *their sum as a binary string*.

Example 1:

```
Input: a = "11", b = "1"
Output: "100"
```

Example 2:

```
Input: a = "1010", b = "1011"
Output: "10101"
```

Constraints:

- `1 <= a.length, b.length <= 10^4`
- `a` and `b` consist only of `'0'` or `'1'` characters.
- Each string does not contain leading zeros except for the zero itself.

```
class Solution {
public:
    string addBinary(string a, string b) {
        string result = "";
        int i = a.size() - 1;
        int j = b.size() - 1;
        int carry = 0;
```

Saved · Ln 1, Col 1

Test Result

Accepted Runtime: 0 ms

Case 1 · Case 2

Input

a =

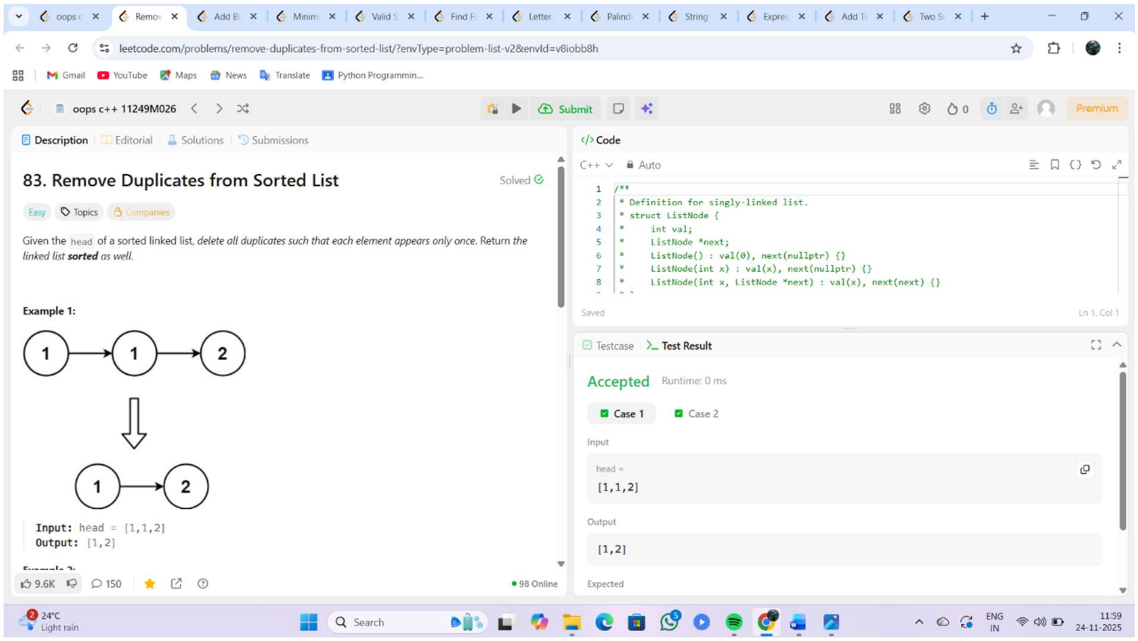
```
"11"
```

b =

```
"1"
```

Output

11.REMOVE DUPLICATES FROM SORTED LIST



leetcode.com/problems/remove-duplicates-from-sorted-list/?envType=problem-list-v2&envId=v8iobb8h

oops c++ 11249M026

83. Remove Duplicates from Sorted List

Solved

Easy · Topics · Companies

Given the `head` of a sorted linked list, *delete all duplicates such that each element appears only once*. Return *the linked list **sorted** as well.*

Example 1:

```
Input: head = [1,1,2]
Output: [1,2]
```

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
```

Accepted Runtime: 0 ms

Case 1 · Case 2

Input

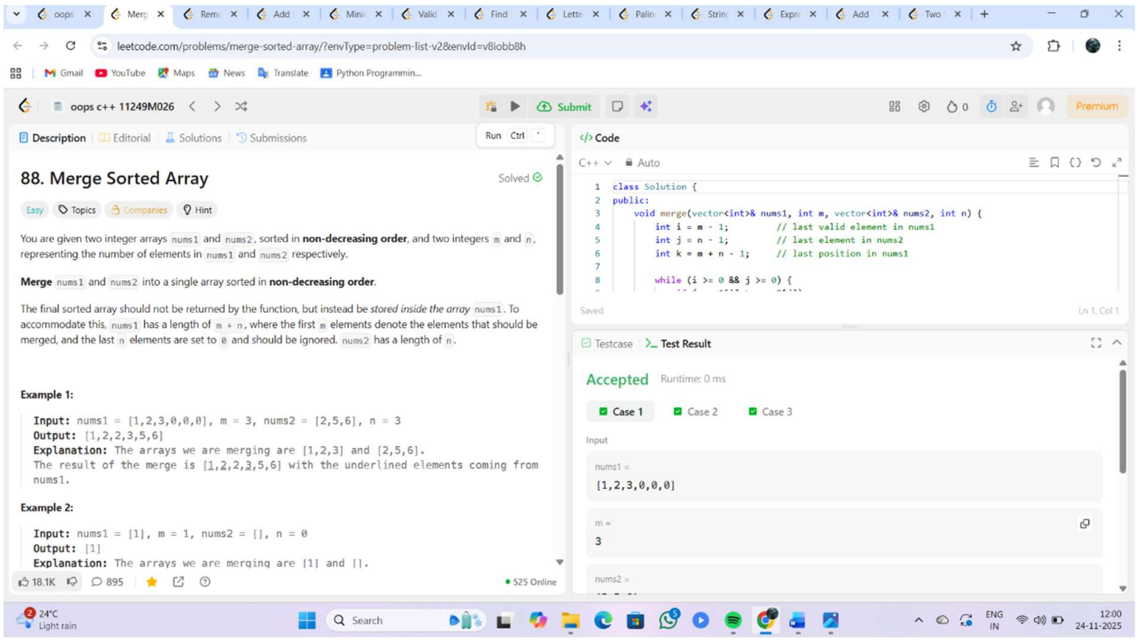
```
head =
[1,1,2]
```

Output

```
[1,2]
```

Expected

12.MERGE SORTED ARRAY



leetcode.com/problems/merge-sorted-array/?envType=problem-list-v2&envId=v8iobb8h

oops c++ 11249M026

88. Merge Sorted Array

Solved

Easy · Topics · Companies · Hint

You are given two integer arrays `nums1` and `nums2`, sorted in **non-decreasing order**, and two integers `m` and `n`, representing the number of elements in `nums1` and `nums2` respectively.

Merge `nums1` and `nums2` into a single array sorted in **non-decreasing order**.

The final sorted array should not be returned by the function, but instead be *stored inside the array* `nums1`. To accommodate this, `nums1` has a length of `m + n`, where the first `m` elements denote the elements that should be merged, and the last `n` elements are set to `0` and should be ignored. `nums2` has a length of `n`.

Example 1:

```
Input: nums1 = [1,2,3,0,0,0], m = 3, nums2 = [2,5,6], n = 3
Output: [1,2,2,3,5,6]
Explanation: The arrays we are merging are [1,2,3] and [2,5,6].
The result of the merge is [1,2,2,3,5,6] with the underlined elements coming from
nums1.
```

Example 2:

```
Input: nums1 = [1], m = 1, nums2 = [], n = 0
Output: [1]
Explanation: The arrays we are merging are [1] and [].
```

```
class Solution {
public:
    void merge(vector<int>& nums1, int m, vector<int>& nums2, int n) {
        int i = m - 1;          // last valid element in nums1
        int j = n - 1;          // last element in nums2
        int k = m + n - 1;      // last position in nums1

        while (i >= 0 && j >= 0) {
```

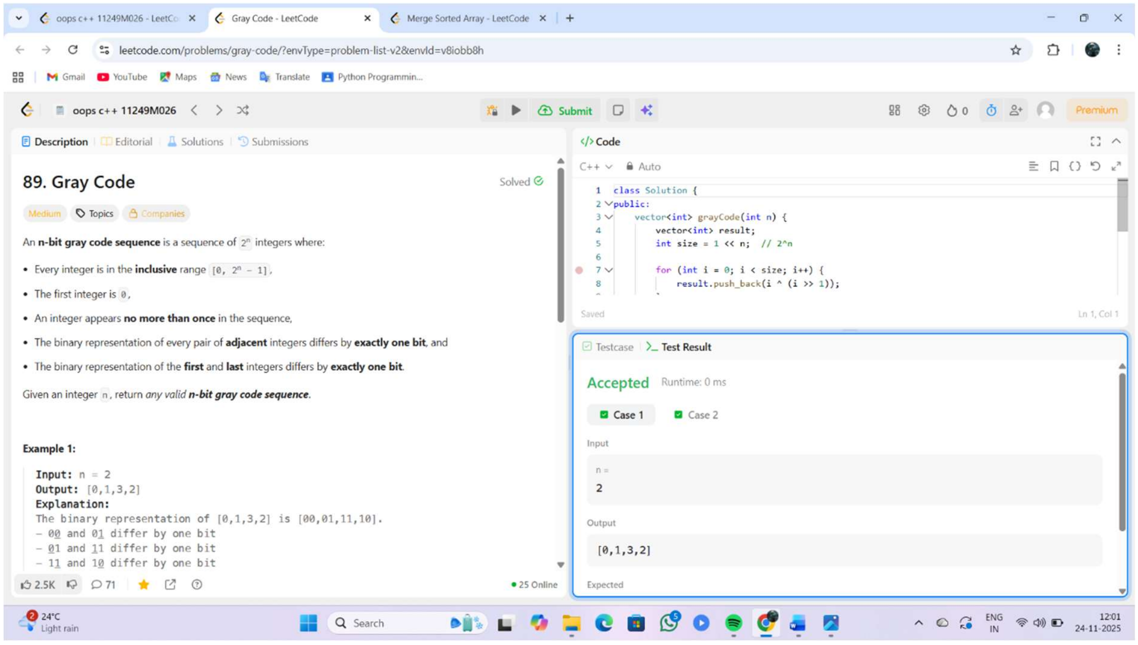
Accepted Runtime: 0 ms

Case 1 · Case 2 · Case 3

Input

```
nums1 =
[1,2,3,0,0,0]
m =
3
nums2 =
```

13.GRAY CODE



89. Gray Code

Solved

An **n-bit gray code sequence** is a sequence of 2^n integers where:

- Every integer is in the **inclusive** range `[0, 2^n - 1]`,
- The first integer is `0`,
- An integer appears **no more than once** in the sequence,
- The binary representation of every pair of **adjacent** integers differs by **exactly one bit**, and
- The binary representation of the **first** and **last** integers differs by **exactly one bit**.

Given an integer `n`, return *any valid* ***n-bit gray code sequence***.

Example 1:

```
Input: n = 2
Output: [0,1,3,2]
Explanation:
The binary representation of [0,1,3,2] is [00,01,11,10].
- 00 and 01 differ by one bit
- 01 and 11 differ by one bit
- 11 and 10 differ by one bit
```

```cpp
class Solution {
public:
    vector<int> grayCode(int n) {
        vector<int> result;
        int size = 1 << n;  // 2^n

        for (int i = 0; i < size; i++) {
            result.push_back(i ^ (i >> 1));
```

Accepted Runtime: 0 ms

Case 1 Case 2

Input

n =

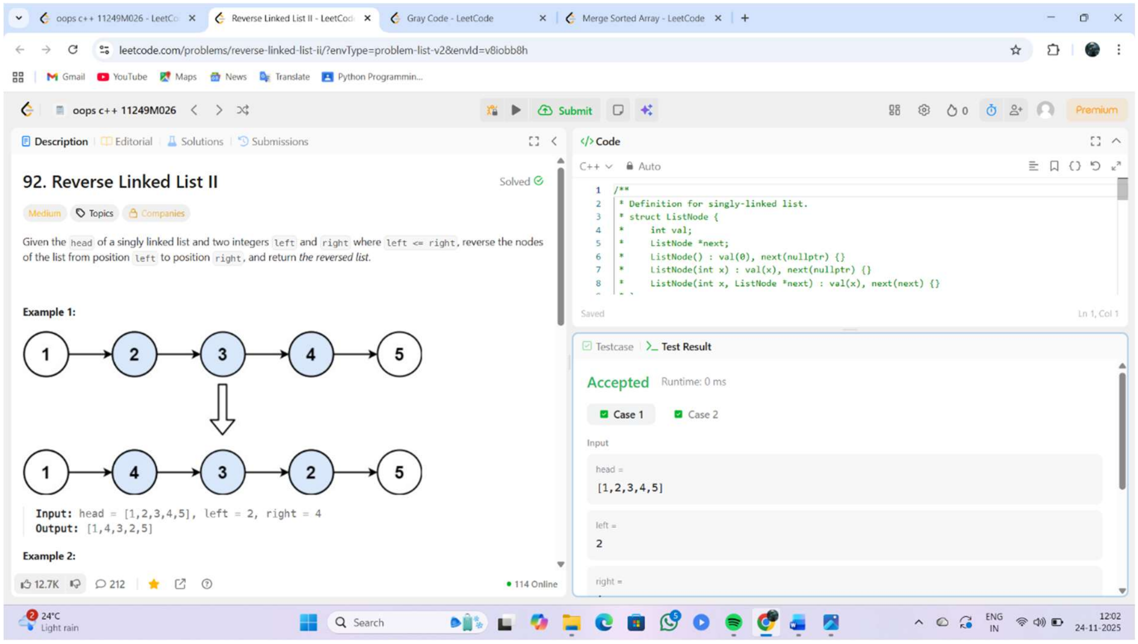
2

Output

```
[0,1,3,2]
```

Expected

14.REVERSE LINKED LIST



92. Reverse Linked List II

Solved

Given the `head` of a singly linked list and two integers `left` and `right` where `left <= right`, reverse the nodes of the list from position `left` to position `right`, and return *the reversed list*.

Example 1:

```
Input: head = [1,2,3,4,5], left = 2, right = 4
Output: [1,4,3,2,5]
```

Example 2:

```cpp
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
```

Accepted Runtime: 0 ms

Case 1 Case 2

Input

head =

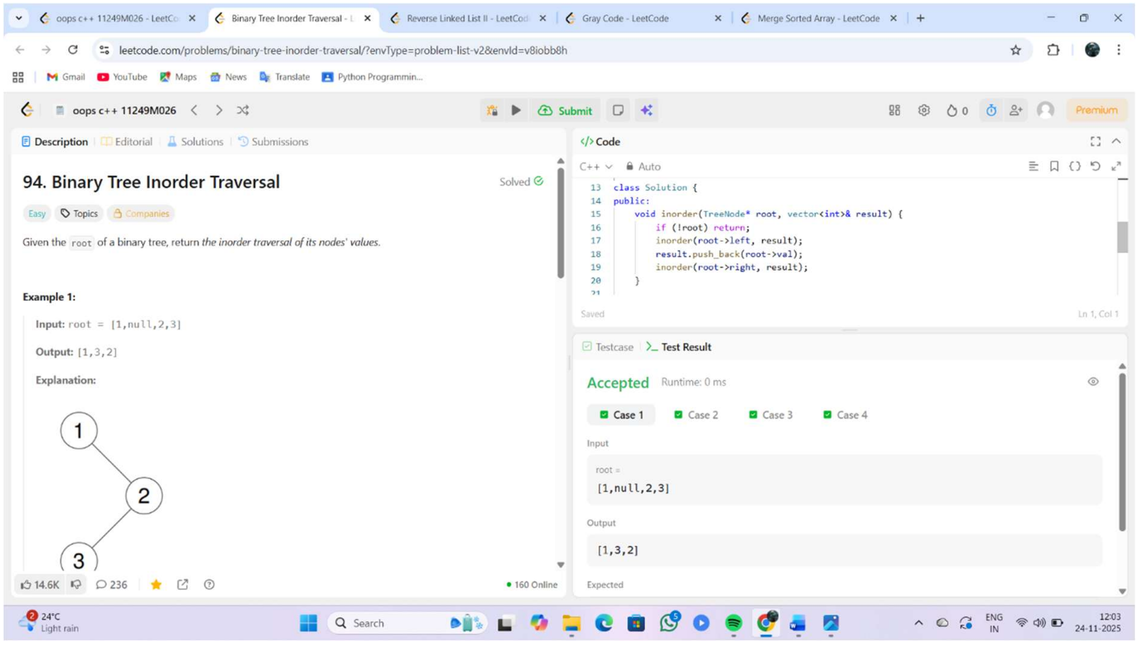
```
[1,2,3,4,5]
```

left =

2

right =

15.BINARY TREE INORDER TRAVERSAL



94. Binary Tree Inorder Traversal

Solved

Easy · Topics · Companies

Given the `root` of a binary tree, return *the inorder traversal of its nodes' values*.

Example 1:

Input: root = [1,null,2,3]

Output: [1,3,2]

Explanation:

```cpp
class Solution {
public:
    void inorder(TreeNode* root, vector<int>& result) {
        if (!root) return;
        inorder(root->left, result);
        result.push_back(root->val);
        inorder(root->right, result);
    }
```

Accepted Runtime: 0 ms

Case 1 · Case 2 · Case 3 · Case 4

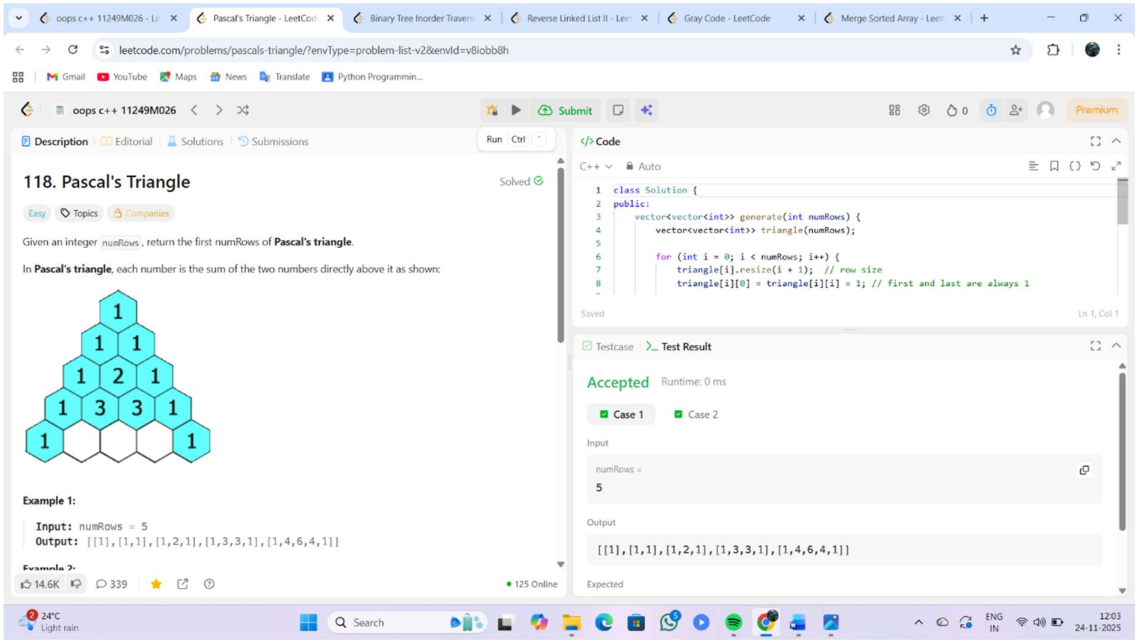
Input

root =
```
[1,null,2,3]
```

Output
```
[1,3,2]
```

Expected

16.PASCAL'S TRIANGLE



118. Pascal's Triangle

Solved

Easy · Topics · Companies

Given an integer `numRows`, return the first numRows of **Pascal's triangle**.

In **Pascal's triangle**, each number is the sum of the two numbers directly above it as shown:

Example 1:

```
Input: numRows = 5
Output: [[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1]]
```

```cpp
class Solution {
public:
    vector<vector<int>> generate(int numRows) {
        vector<vector<int>> triangle(numRows);

        for (int i = 0; i < numRows; i++) {
            triangle[i].resize(i + 1);  // row size
            triangle[i][0] = triangle[i][i] = 1; // first and last are always 1
```

Accepted Runtime: 0 ms

Case 1 · Case 2

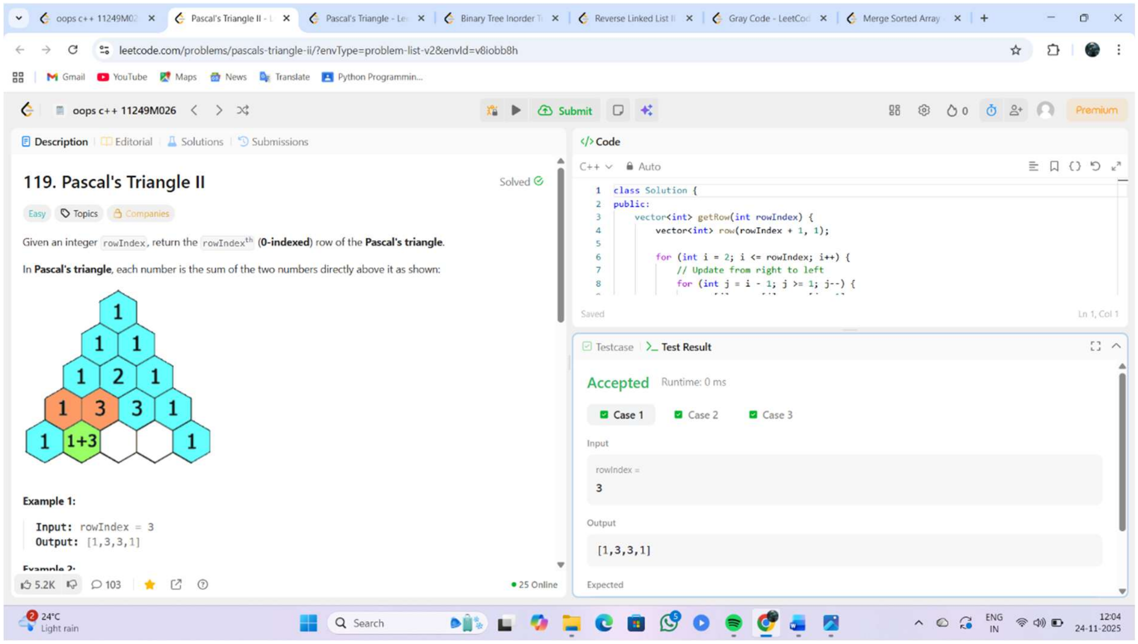
Input

numRows =
```
5
```

Output
```
[[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1]]
```

Expected

17.PASCAL'S TRIANGLE 2



leetcode.com/problems/pascals-triangle-ii/?envType=problem-list-v2&envId=v8iobb8h

119. Pascal's Triangle II

Given an integer `rowIndex`, return the `rowIndex`th (**0-indexed**) row of the **Pascal's triangle**.

In **Pascal's triangle**, each number is the sum of the two numbers directly above it as shown:

Example 1:

```
Input: rowIndex = 3
Output: [1,3,3,1]
```

```cpp
class Solution {
public:
    vector<int> getRow(int rowIndex) {
        vector<int> row(rowIndex + 1, 1);

        for (int i = 2; i <= rowIndex; i++) {
            // Update from right to left
            for (int j = i - 1; j >= 1; j--) {
```

Accepted Runtime: 0 ms

Input

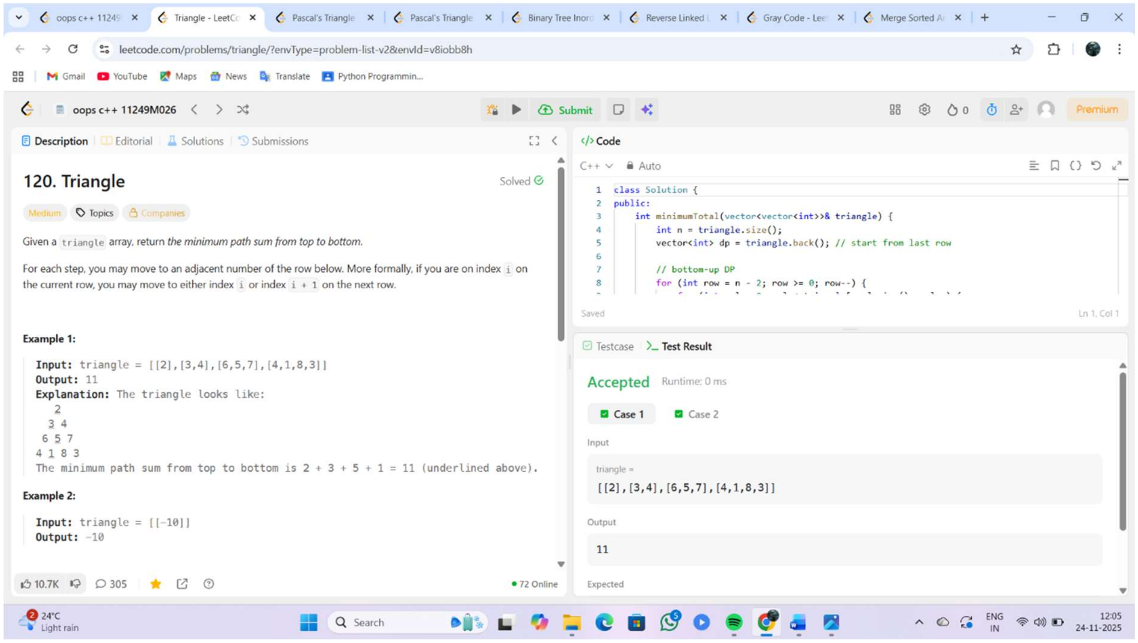
rowIndex =

```
3
```

Output

```
[1,3,3,1]
```

18.TRIANGLE



leetcode.com/problems/triangle/?envType=problem-list-v2&envId=v8iobb8h

120. Triangle

Given a `triangle` array, return *the minimum path sum from top to bottom*.

For each step, you may move to an adjacent number of the row below. More formally, if you are on index `i` on the current row, you may move to either index `i` or index `i + 1` on the next row.

Example 1:

```
Input: triangle = [[2],[3,4],[6,5,7],[4,1,8,3]]
Output: 11
Explanation: The triangle looks like:
   2
  3 4
 6 5 7
4 1 8 3
The minimum path sum from top to bottom is 2 + 3 + 5 + 1 = 11 (underlined above).
```

Example 2:

```
Input: triangle = [[-10]]
Output: -10
```

```cpp
class Solution {
public:
    int minimumTotal(vector<vector<int>>& triangle) {
        int n = triangle.size();
        vector<int> dp = triangle.back(); // start from last row

        // bottom-up DP
        for (int row = n - 2; row >= 0; row--) {
```

Accepted Runtime: 0 ms

Input

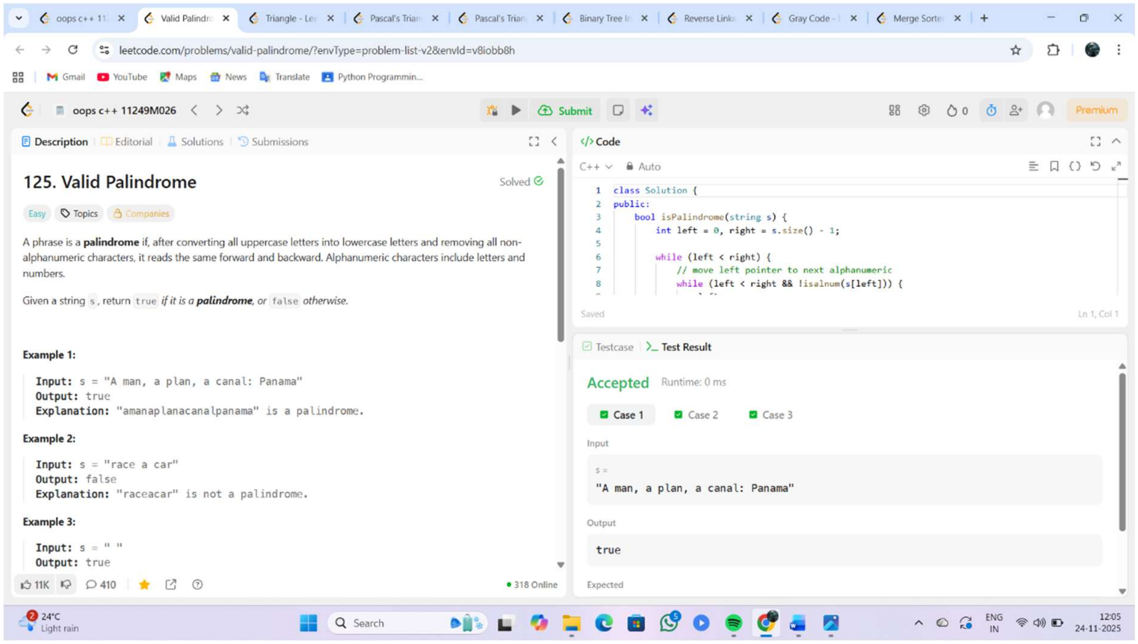
triangle =

```
[[2],[3,4],[6,5,7],[4,1,8,3]]
```

Output

```
11
```

19.VALID PALINDROME



leetcode.com/problems/valid-palindrome/?envType=problem-list-v2&envId=v8iobb8h

oops c++ 11249M026

125. Valid Palindrome

Solved

Easy · Topics · Companies

A phrase is a **palindrome** if, after converting all uppercase letters into lowercase letters and removing all non-alphanumeric characters, it reads the same forward and backward. Alphanumeric characters include letters and numbers.

Given a string `s`, return `true` *if it is a **palindrome**, or* `false` *otherwise.*

Example 1:

```
Input: s = "A man, a plan, a canal: Panama"
Output: true
Explanation: "amanaplanacanalpanama" is a palindrome.
```

Example 2:

```
Input: s = "race a car"
Output: false
Explanation: "raceacar" is not a palindrome.
```

Example 3:

```
Input: s = " "
Output: true
```

11K · 410 · 318 Online

Code — C++ · Auto

```cpp
class Solution {
public:
    bool isPalindrome(string s) {
        int left = 0, right = s.size() - 1;

        while (left < right) {
            // move left pointer to next alphanumeric
            while (left < right && !isalnum(s[left])) {
```

Saved · Ln 1, Col 1

Testcase · Test Result

Accepted Runtime: 0 ms

Case 1 · Case 2 · Case 3

Input

s =

```
"A man, a plan, a canal: Panama"
```

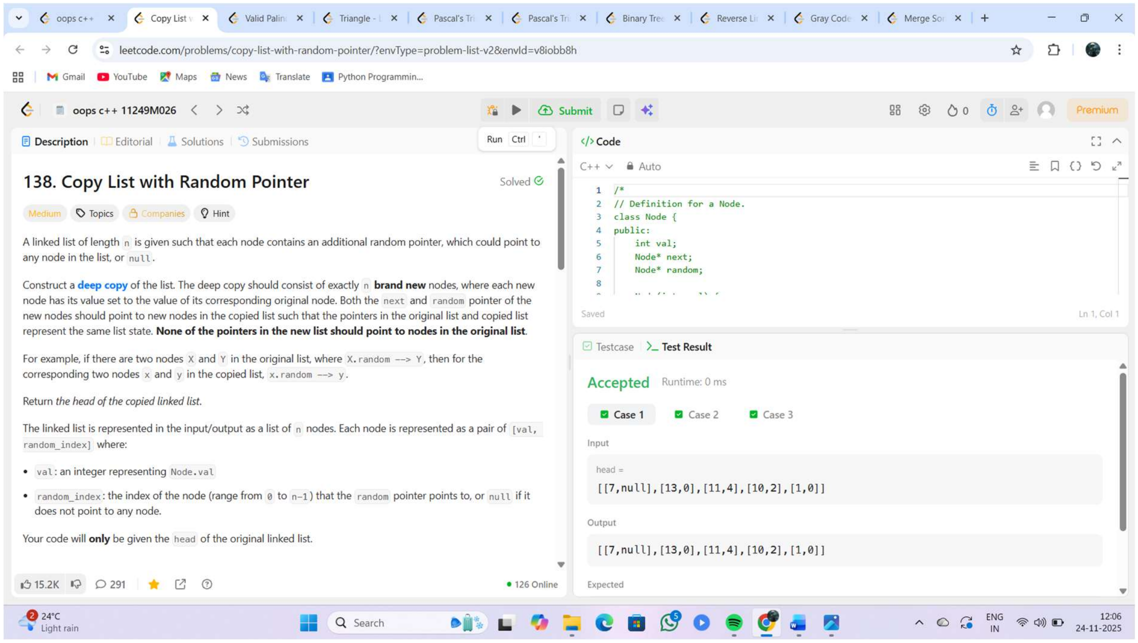
Output

```
true
```

Expected

24°C Light rain · 12:05 24-11-2025

20.COPY LIST WITH RANDOM POINTER



leetcode.com/problems/copy-list-with-random-pointer/?envType=problem-list-v2&envId=v8iobb8h

oops c++ 11249M026

138. Copy List with Random Pointer

Solved

Medium · Topics · Companies · Hint

A linked list of length `n` is given such that each node contains an additional random pointer, which could point to any node in the list, or `null`.

Construct a **deep copy** of the list. The deep copy should consist of exactly `n` **brand new** nodes, where each new node has its value set to the value of its corresponding original node. Both the `next` and `random` pointer of the new nodes should point to new nodes in the copied list such that the pointers in the original list and copied list represent the same list state. **None of the pointers in the new list should point to nodes in the original list.**

For example, if there are two nodes `X` and `Y` in the original list, where `X.random --> Y`, then for the corresponding two nodes `x` and `y` in the copied list, `x.random --> y`.

Return *the head of the copied linked list.*

The linked list is represented in the input/output as a list of `n` nodes. Each node is represented as a pair of `[val, random_index]` where:

- `val`: an integer representing `Node.val`
- `random_index`: the index of the node (range from `0` to `n-1`) that the `random` pointer points to, or `null` if it does not point to any node.

Your code will **only** be given the `head` of the original linked list.

15.2K · 291 · 126 Online

Code — C++ · Auto

```cpp
/*
// Definition for a Node.
class Node {
public:
    int val;
    Node* next;
    Node* random;

```

Saved · Ln 1, Col 1

Testcase · Test Result

Accepted Runtime: 0 ms

Case 1 · Case 2 · Case 3

Input

head =

```
[[7,null],[13,0],[11,4],[10,2],[1,0]]
```

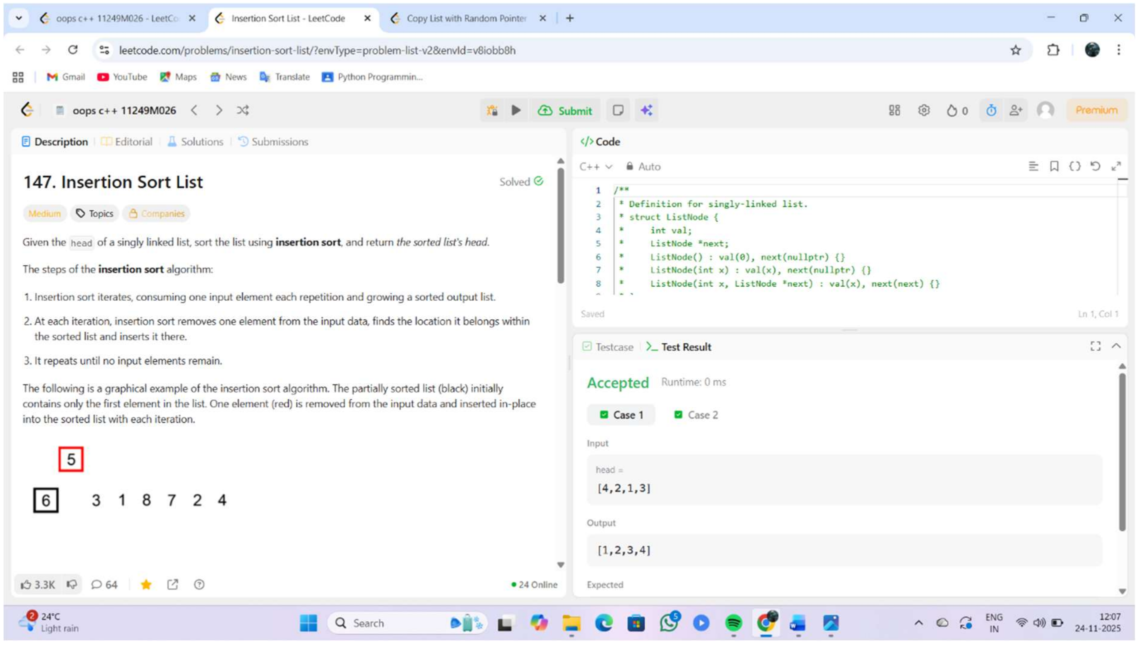
Output

```
[[7,null],[13,0],[11,4],[10,2],[1,0]]
```

Expected

24°C Light rain · 12:06 24-11-2025

21.INSERTION SORT LIST



leetcode.com/problems/insertion-sort-list/?envType=problem-list-v2&envId=v8iobb8h

oops c++ 11249M026

147. Insertion Sort List

Solved ✓

Medium · Topics · Companies

Given the `head` of a singly linked list, sort the list using **insertion sort**, and return *the sorted list's head*.

The steps of the **insertion sort** algorithm:

1. Insertion sort iterates, consuming one input element each repetition and growing a sorted output list.
2. At each iteration, insertion sort removes one element from the input data, finds the location it belongs within the sorted list and inserts it there.
3. It repeats until no input elements remain.

The following is a graphical example of the insertion sort algorithm. The partially sorted list (black) initially contains only the first element in the list. One element (red) is removed from the input data and inserted in-place into the sorted list with each iteration.

5

6 3 1 8 7 2 4

3.3K · 64 · 24 Online

Code

C++ · Auto

```cpp
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
```

Saved · Ln 1, Col 1

Testcase · Test Result

Accepted Runtime: 0 ms

✓ Case 1 ✓ Case 2

Input

head =

```
[4,2,1,3]
```

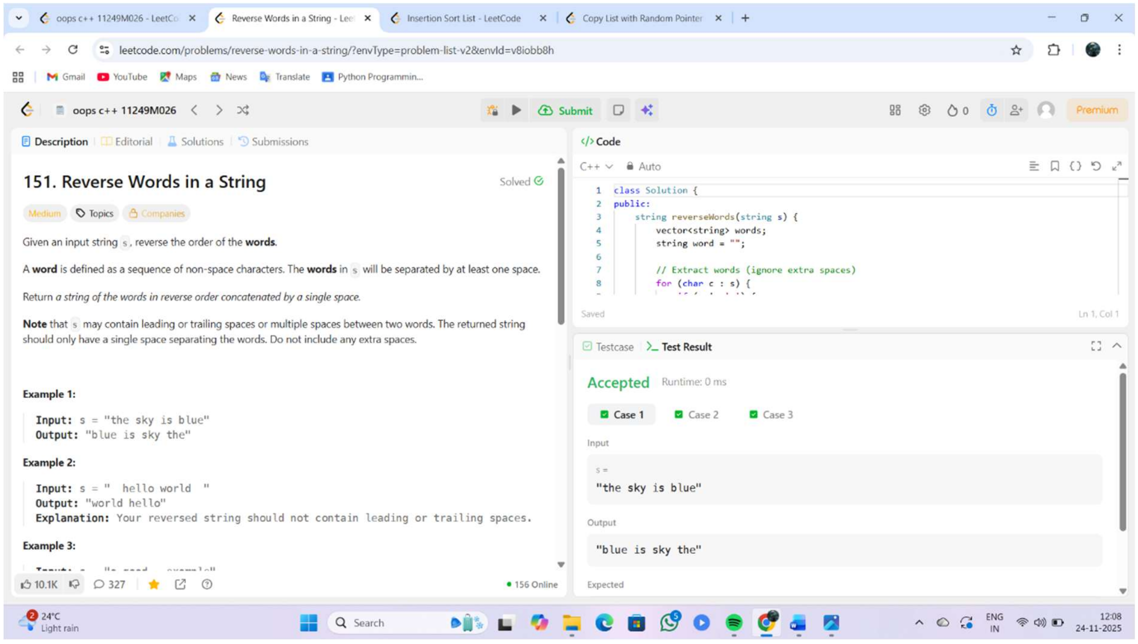
Output

```
[1,2,3,4]
```

Expected

24°C Light rain · 12:07 24-11-2025

22.REVERSE WORDS IN A STRING



leetcode.com/problems/reverse-words-in-a-string/?envType=problem-list-v2&envId=v8iobb8h

oops c++ 11249M026

151. Reverse Words in a String

Solved ✓

Medium · Topics · Companies

Given an input string `s`, reverse the order of the **words**.

A **word** is defined as a sequence of non-space characters. The **words** in `s` will be separated by at least one space.

Return *a string of the words in reverse order concatenated by a single space.*

Note that `s` may contain leading or trailing spaces or multiple spaces between two words. The returned string should only have a single space separating the words. Do not include any extra spaces.

Example 1:

```
Input: s = "the sky is blue"
Output: "blue is sky the"
```

Example 2:

```
Input: s = "  hello world  "
Output: "world hello"
Explanation: Your reversed string should not contain leading or trailing spaces.
```

Example 3:

10.1K · 327 · 156 Online

Code

C++ · Auto

```cpp
class Solution {
public:
    string reverseWords(string s) {
        vector<string> words;
        string word = "";

        // Extract words (ignore extra spaces)
        for (char c : s) {
```

Saved · Ln 1, Col 1

Testcase · Test Result

Accepted Runtime: 0 ms

✓ Case 1 ✓ Case 2 ✓ Case 3

Input

s =

```
"the sky is blue"
```

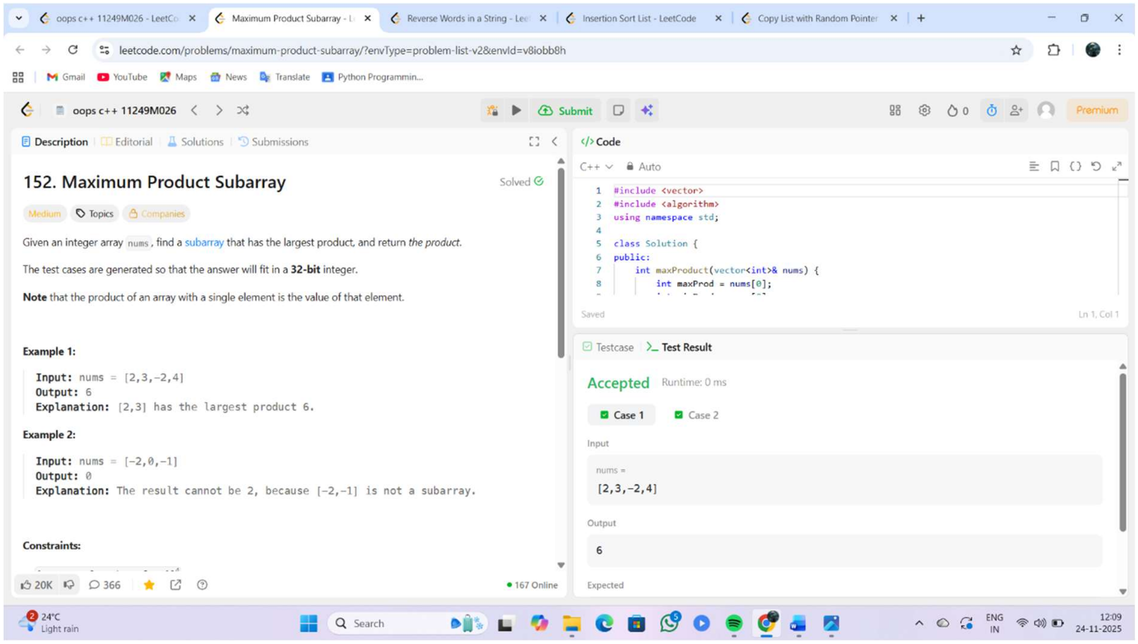
Output

```
"blue is sky the"
```

Expected

24°C Light rain · 12:08 24-11-2025

23.MAXIMUM PRODUCT SUBARRAY



leetcode.com/problems/maximum-product-subarray/?envType=problem-list-v2&envId=v8iobb8h

152. Maximum Product Subarray

Solved

Medium · Topics · Companies

Given an integer array `nums`, find a subarray that has the largest product, and return *the product*.

The test cases are generated so that the answer will fit in a **32-bit** integer.

Note that the product of an array with a single element is the value of that element.

Example 1:

```
Input: nums = [2,3,-2,4]
Output: 6
Explanation: [2,3] has the largest product 6.
```

Example 2:

```
Input: nums = [-2,0,-1]
Output: 0
Explanation: The result cannot be 2, because [-2,-1] is not a subarray.
```

Constraints:

```cpp
#include <vector>
#include <algorithm>
using namespace std;

class Solution {
public:
    int maxProduct(vector<int>& nums) {
        int maxProd = nums[0];
```

Test Result

Accepted Runtime: 0 ms

Case 1 · Case 2

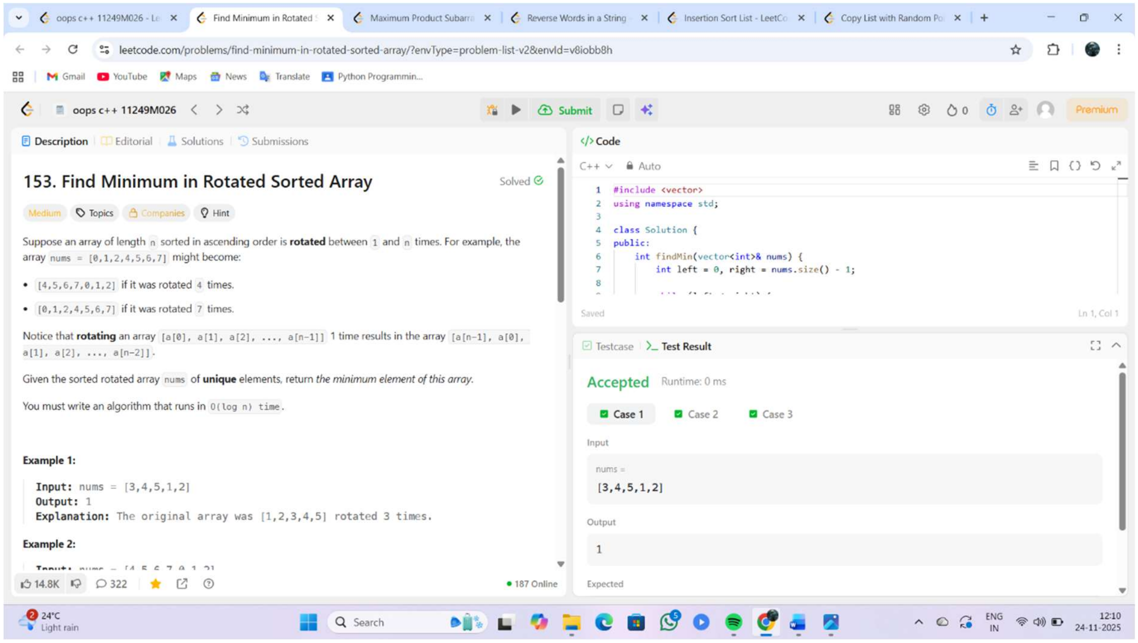
Input

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

Output
```
6
```

Expected

24.FIND MINIMUM IN ROTATED SORTED ARRAY



leetcode.com/problems/find-minimum-in-rotated-sorted-array/?envType=problem-list-v2&envId=v8iobb8h

153. Find Minimum in Rotated Sorted Array

Solved

Medium · Topics · Companies · Hint

Suppose an array of length `n` sorted in ascending order is **rotated** between `1` and `n` times. For example, the array `nums = [0,1,2,4,5,6,7]` might become:

- `[4,5,6,7,0,1,2]` if it was rotated `4` times.
- `[0,1,2,4,5,6,7]` if it was rotated `7` times.

Notice that **rotating** an array `[a[0], a[1], a[2], ..., a[n-1]]` 1 time results in the array `[a[n-1], a[0], a[1], a[2], ..., a[n-2]]`.

Given the sorted rotated array `nums` of **unique** elements, return *the minimum element of this array*.

You must write an algorithm that runs in `O(log n) time`.

Example 1:

```
Input: nums = [3,4,5,1,2]
Output: 1
Explanation: The original array was [1,2,3,4,5] rotated 3 times.
```

Example 2:

```cpp
#include <vector>
using namespace std;

class Solution {
public:
    int findMin(vector<int>& nums) {
        int left = 0, right = nums.size() - 1;
```

Test Result

Accepted Runtime: 0 ms

Case 1 · Case 2 · Case 3

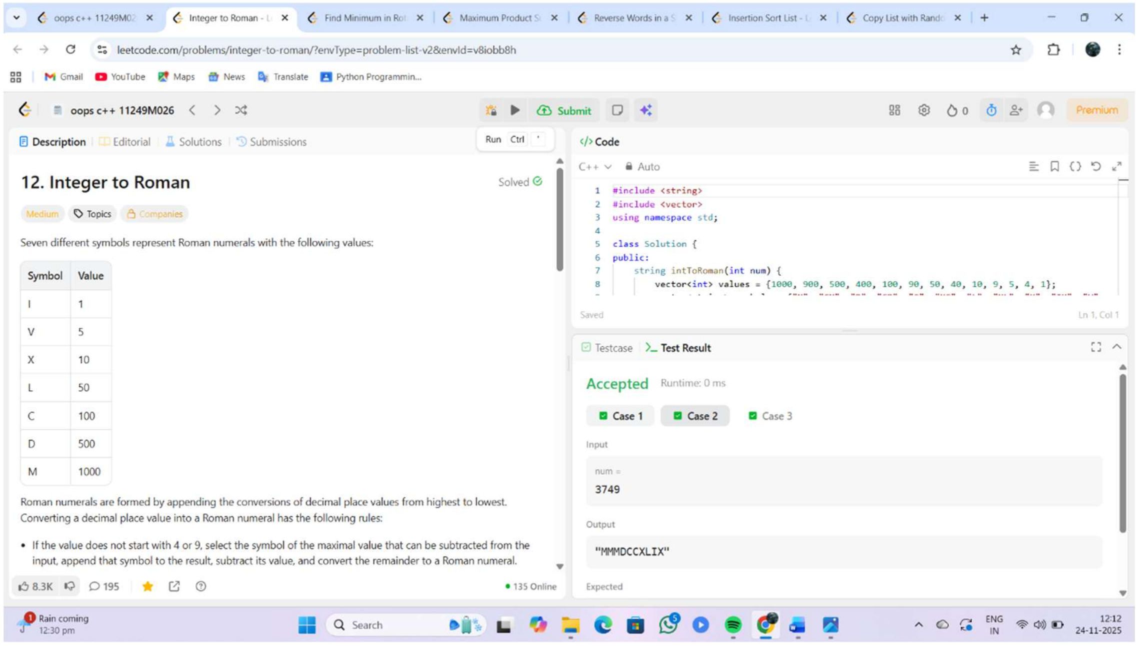
Input

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

Output
```
1
```

Expected

25.INTEGER TO ROMAN



leetcode.com/problems/integer-to-roman/?envType=problem-list-v2&envId=v8iobb8h

12. Integer to Roman

Solved

Medium · Topics · Companies

Seven different symbols represent Roman numerals with the following values:

Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

Roman numerals are formed by appending the conversions of decimal place values from highest to lowest. Converting a decimal place value into a Roman numeral has the following rules:

- If the value does not start with 4 or 9, select the symbol of the maximal value that can be subtracted from the input, append that symbol to the result, subtract its value, and convert the remainder to a Roman numeral.

```cpp
#include <string>
#include <vector>
using namespace std;

class Solution {
public:
    string intToRoman(int num) {
        vector<int> values = {1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1};
```

Accepted Runtime: 0 ms

Case 1 · Case 2 · Case 3

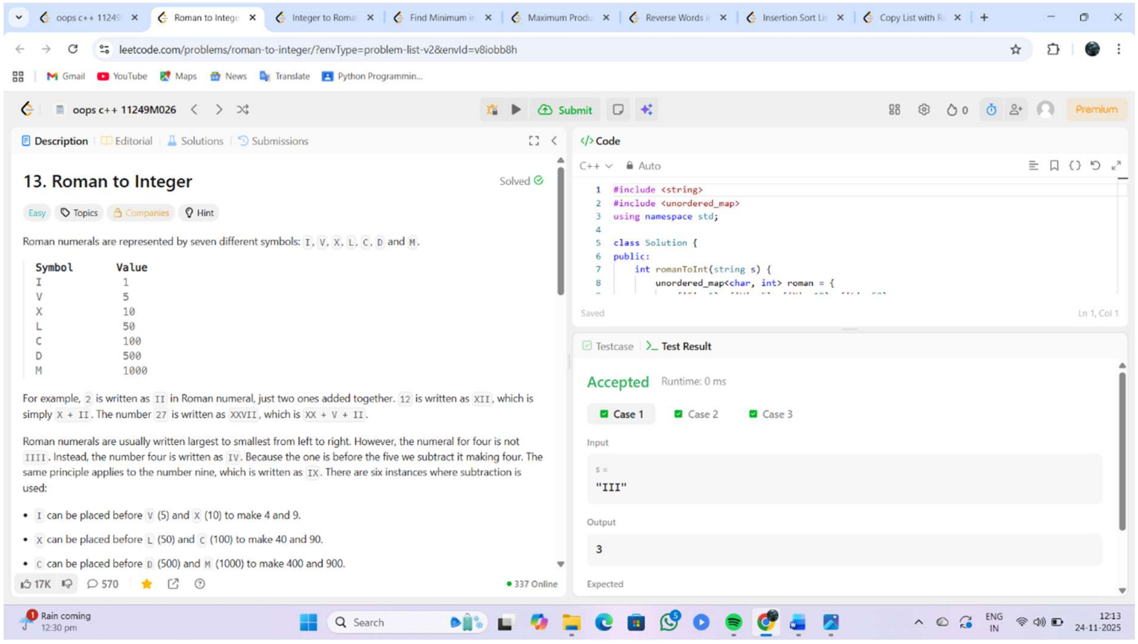
Input

num =
```
3749
```

Output
```
"MMMDCCXLIX"
```

Expected

26.ROMAN TO INTEGER



leetcode.com/problems/roman-to-integer/?envType=problem-list-v2&envId=v8iobb8h

13. Roman to Integer

Solved

Easy · Topics · Companies · Hint

Roman numerals are represented by seven different symbols: `I`, `V`, `X`, `L`, `C`, `D` and `M`.

```
Symbol       Value
I             1
V             5
X             10
L             50
C             100
D             500
M             1000
```

For example, `2` is written as `II` in Roman numeral, just two ones added together. `12` is written as `XII`, which is simply `X + II`. The number `27` is written as `XXVII`, which is `XX + V + II`.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not `IIII`. Instead, the number four is written as `IV`. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as `IX`. There are six instances where subtraction is used:

- `I` can be placed before `V` (5) and `X` (10) to make 4 and 9.
- `X` can be placed before `L` (50) and `C` (100) to make 40 and 90.
- `C` can be placed before `D` (500) and `M` (1000) to make 400 and 900.

```cpp
#include <string>
#include <unordered_map>
using namespace std;

class Solution {
public:
    int romanToInt(string s) {
        unordered_map<char, int> roman = {
```

Accepted Runtime: 0 ms

Case 1 · Case 2 · Case 3

Input

s =
```
"III"
```

Output
```
3
```

Expected