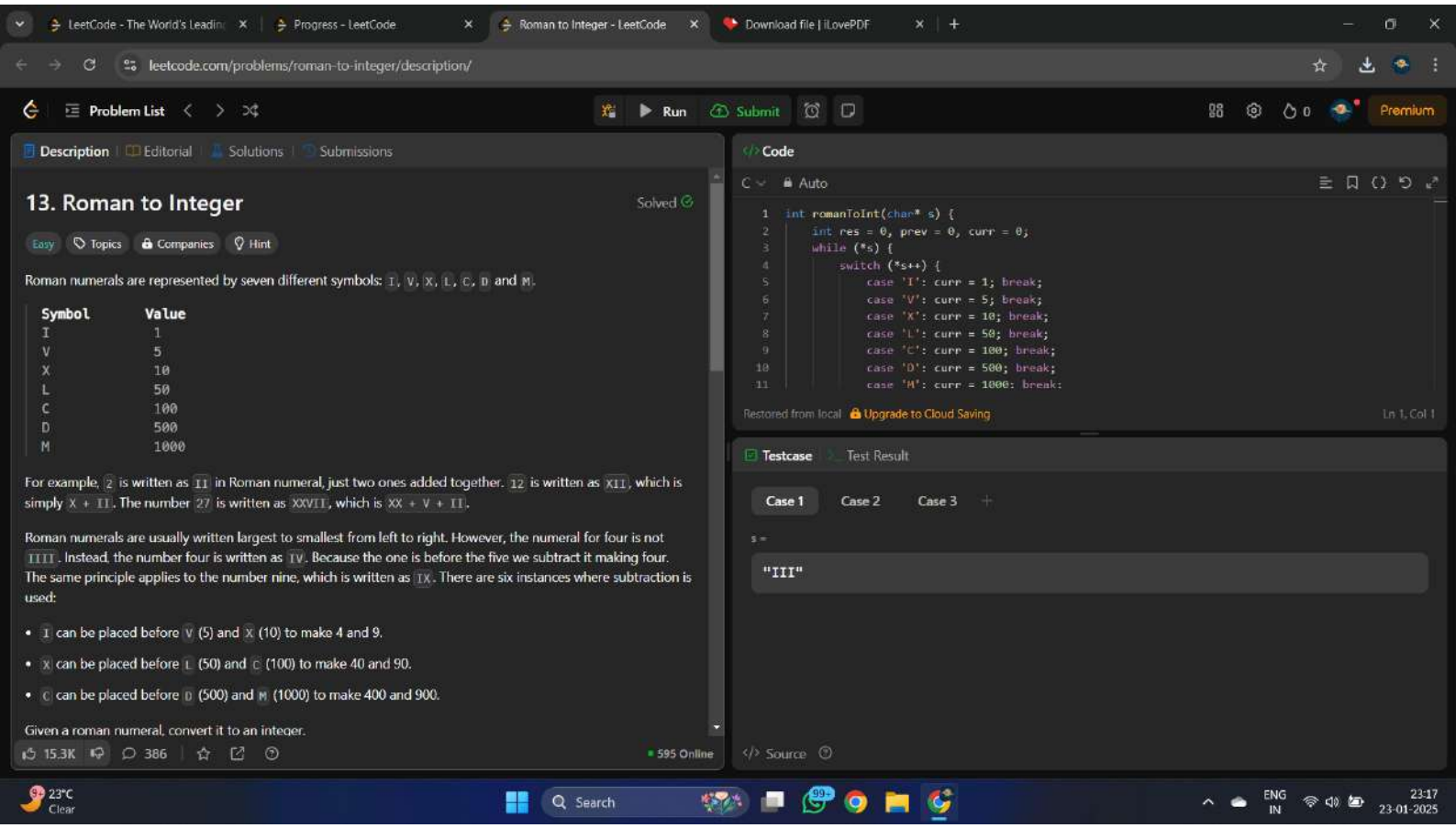

LeetCode - The World's Leadin
Progress - LeetCode
Roman to Integer - LeetCode
Download file | iLovePDF
leetcode.com/problems/roman-to-integer/description/
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
13. Roman to Integer
Solved
Easy
Topics
Companies
Hint
Roman numerals 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.
Given a roman numeral, convert it to an integer.
15.3K
386
595 Online
Code
C
Auto
int romanToInt(char* s) {
int res = 0, prev = 0, curr = 0;
while (*s) {
switch (*s++) {
case 'I': curr = 1; break;
case 'V': curr = 5; break;
case 'X': curr = 10; break;
case 'L': curr = 50; break;
case 'C': curr = 100; break;
case 'D': curr = 500; break;
case 'M': curr = 1000; break;
Restored from local
Upgrade to Cloud Saving
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
Case 3
s =
"III"
Source
23°C
Clear
Search
ENG
IN
23:17
23-01-2025

LeetCode - The World's Leadin | Progress - LeetCode | Check if a Parentheses String C | Download file | iLovePDF

leetcode.com/problems/check-if-a-parentheses-string-can-be-valid/description/

Problem List | Run | Submit | Premium

Description | Editorial | Solutions | Submissions

2116. Check if a Parentheses String Can Be Valid

Solved

Medium | Topics | Companies | Hint

A parentheses string is a **non-empty** string consisting only of `'('` and `')'`. It is valid if **any** of the following conditions is **true**:

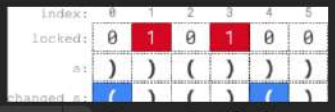
- It is `()`.
- It can be written as `AB` (`A` concatenated with `B`), where `A` and `B` are valid parentheses strings.
- It can be written as `(A)`, where `A` is a valid parentheses string.

You are given a parentheses string `s` and a string `locked`, both of length `n`. `locked` is a binary string consisting only of `'0'`s and `'1'`s. For **each** index `i` of `locked`,

- If `locked[i]` is `'1'`, you **cannot** change `s[i]`.
- But if `locked[i]` is `'0'`, you **can** change `s[i]` to either `'('` or `')'`.

Return `true` *if you can make* `s` *a valid parentheses string.* Otherwise, return `false`.

Example 1:



1.9K | 222 | 150 Online

Code

C | Auto

```
bool canBeValid(char* s, char* locked) {
    int n = strlen(s), open = 0, balance = 0;
    if (n % 2 != 0) return false;

    for (int i = 0; i < n; i++) {
        if (locked[i] == '0' || s[i] == '(') open++;
        else open--;
        if (open < 0) return false;
    }

    open = 0;
```

Restored from local | Upgrade to Cloud Saving | Ln 1, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3

s =

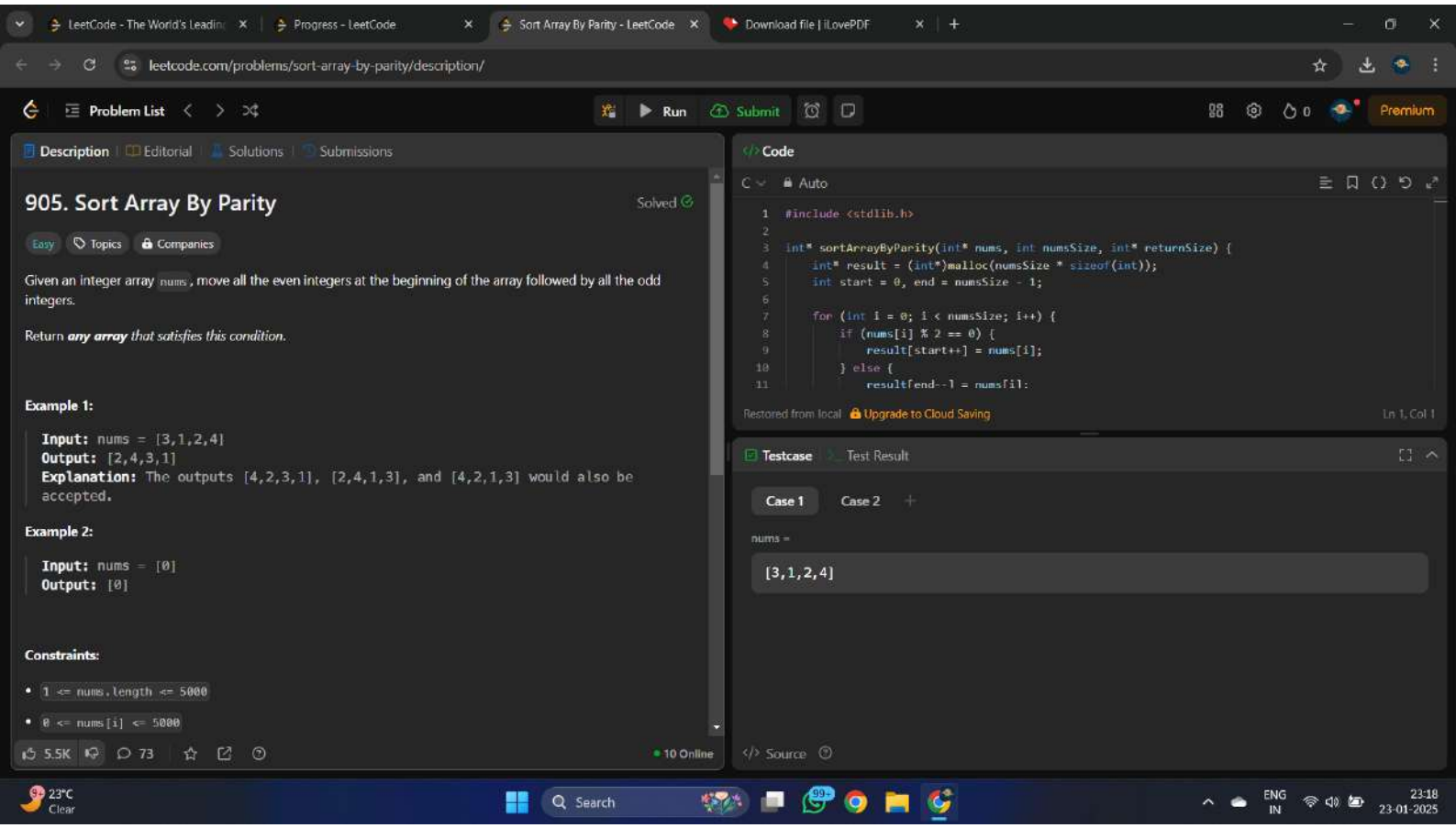
"))()))"

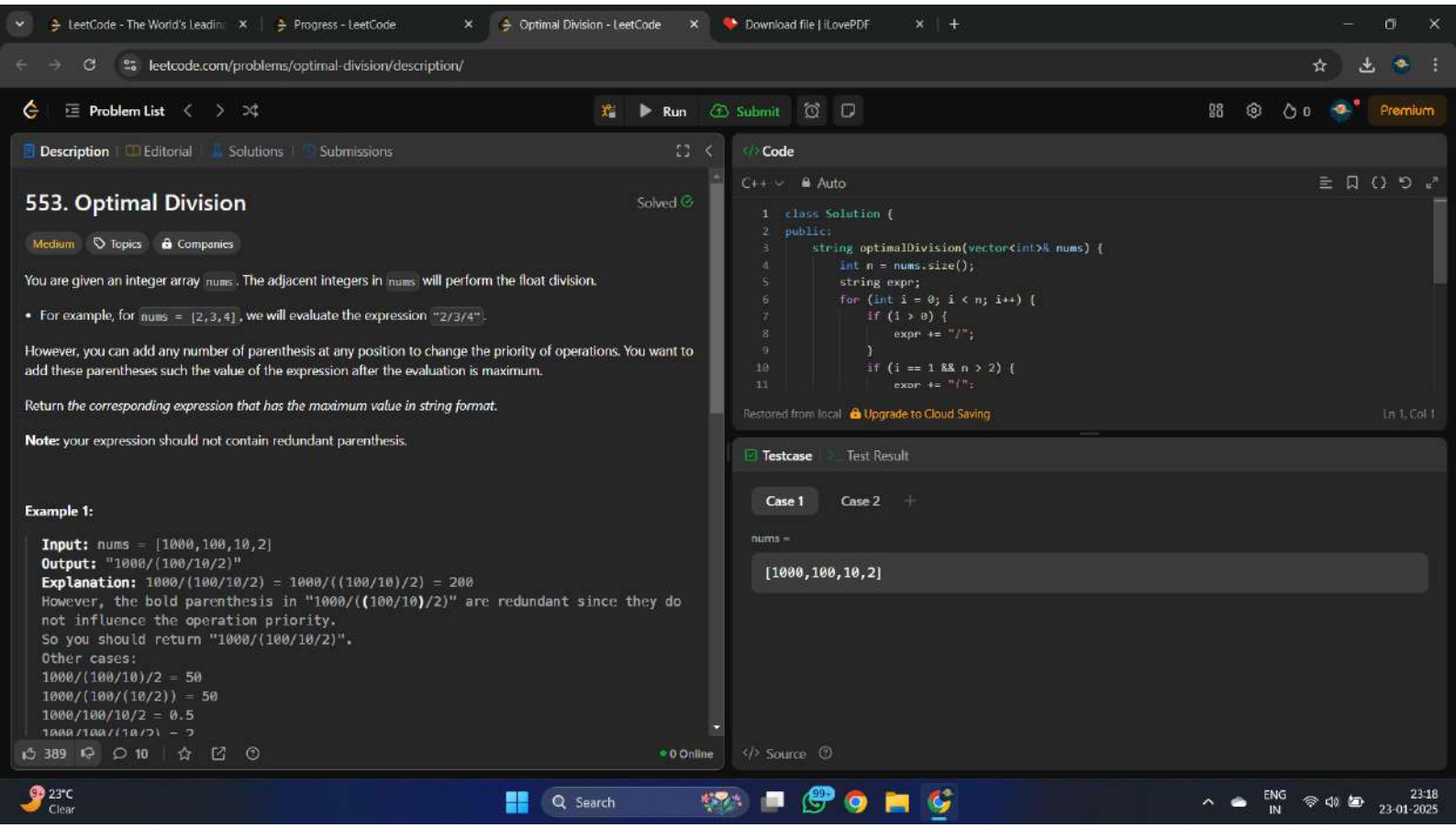
locked =

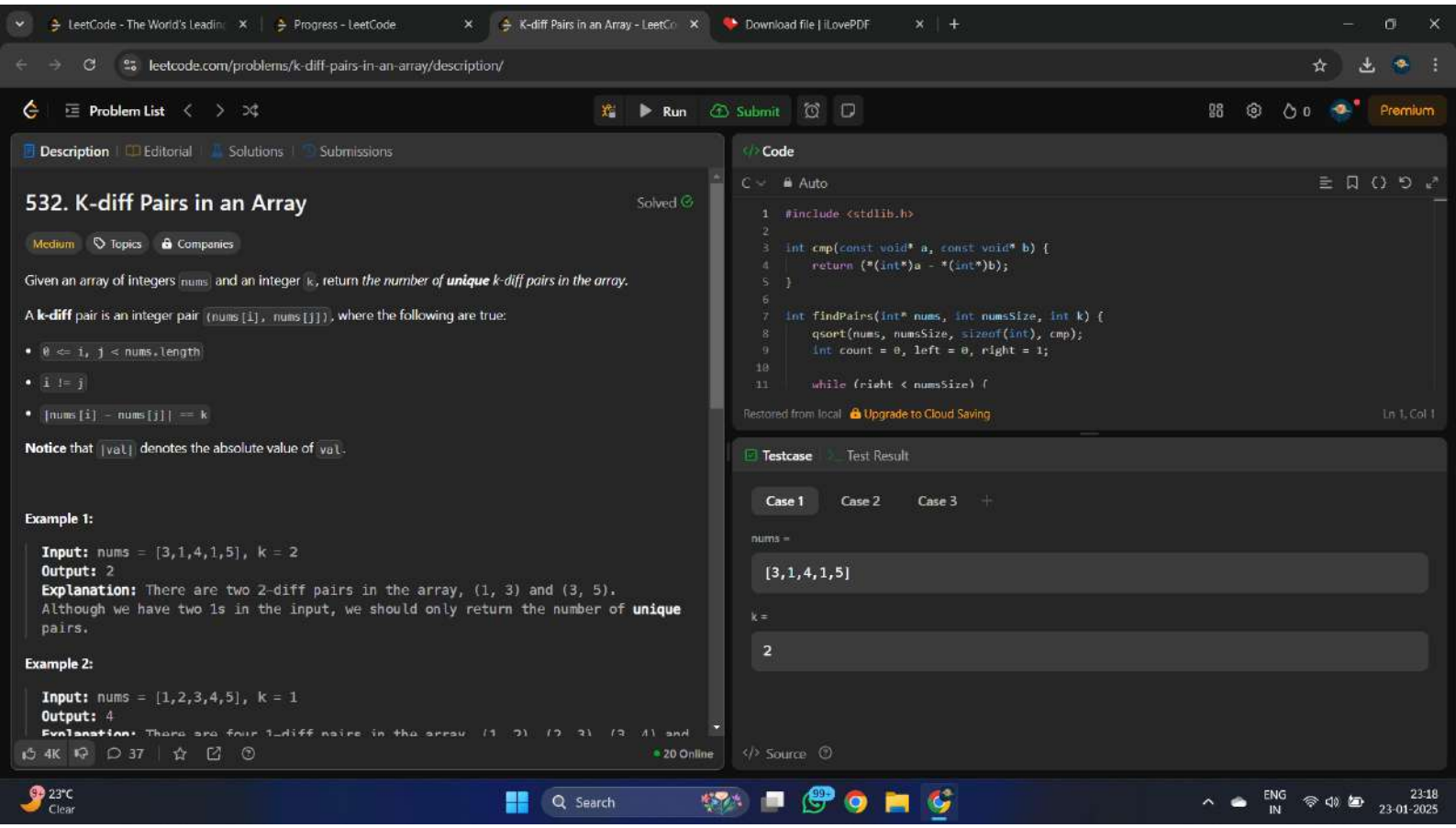
"010100"

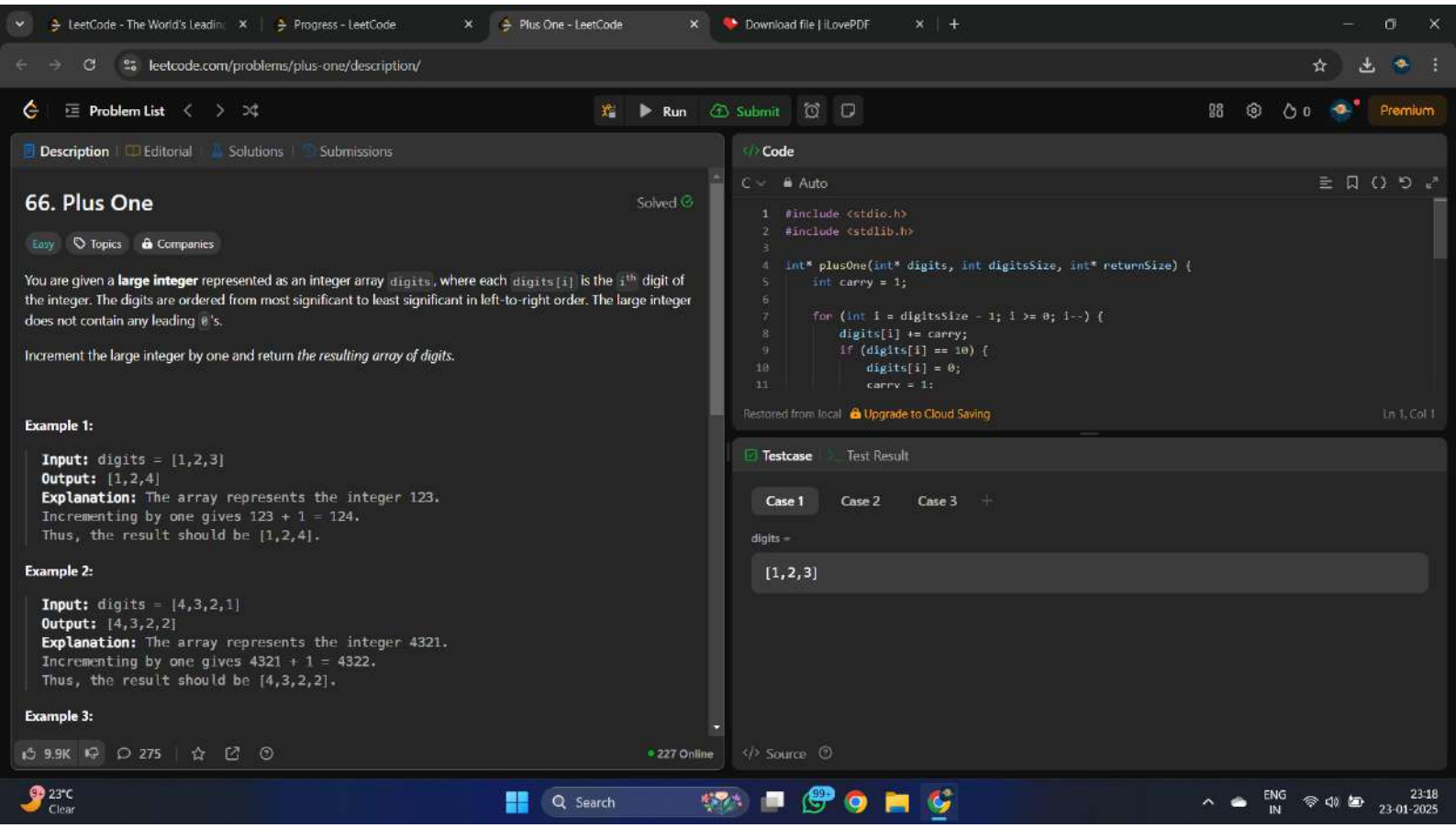
Source

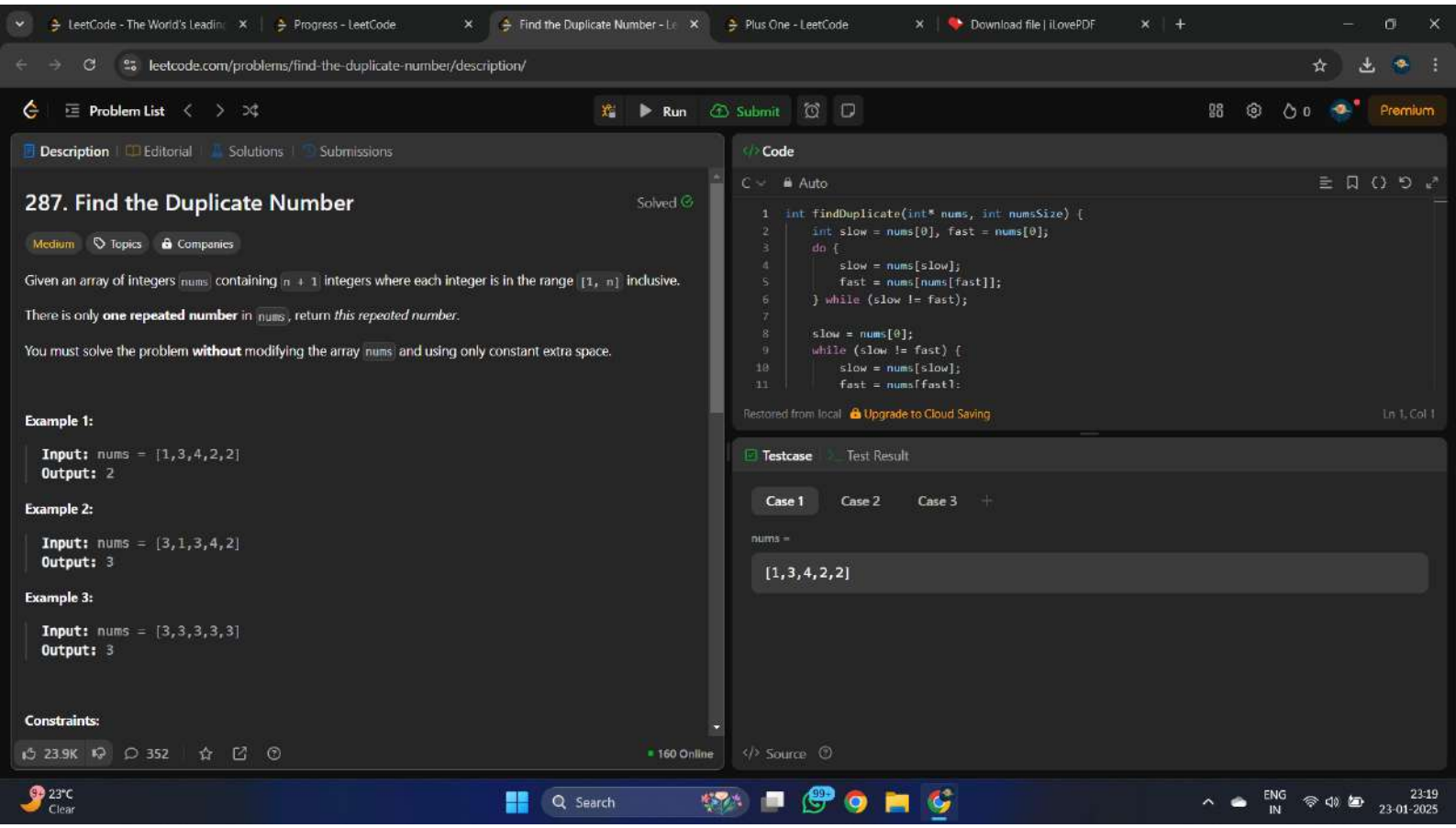
23°C Clear | Search | ENG IN | 23:17 23-01-2025

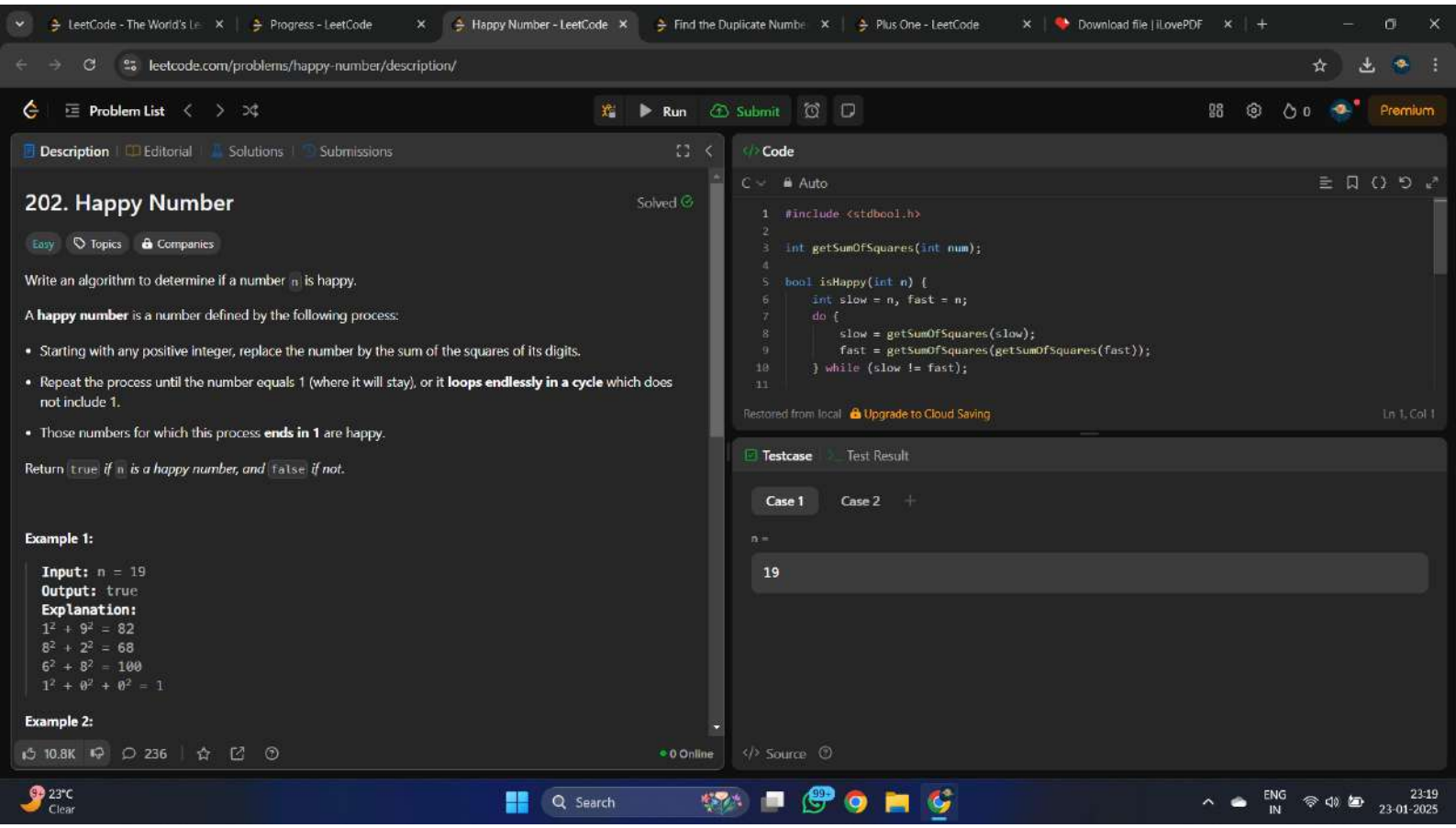

LeetCode - The World's Leadin
Progress - LeetCode
Sort Array By Parity - LeetCode
Download file | iLovePDF
leetcode.com/problems/sort-array-by-parity/description/
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
905. Sort Array By Parity
Solved
Easy
Topics
Companies
Given an integer array nums, move all the even integers at the beginning of the array followed by all the odd integers.
Return any array that satisfies this condition.
Example 1:
Input: nums = [3,1,2,4]
Output: [2,4,3,1]
Explanation: The outputs [4,2,3,1], [2,4,1,3], and [4,2,1,3] would also be accepted.
Example 2:
Input: nums = [0]
Output: [0]
Constraints:
1 <= nums.length <= 5000
0 <= nums[i] <= 5000
5.5K
73
10 Online
Code
C
Auto
#include <stdlib.h>
int* sortArrayByParity(int* nums, int numsSize, int* returnSize) {
int* result = (int*)malloc(numsSize * sizeof(int));
int start = 0, end = numsSize - 1;
for (int i = 0; i < numsSize; i++) {
if (nums[i] % 2 == 0) {
result[start++] = nums[i];
} else {
result[end--] = nums[i];
Restored from local
Upgrade to Cloud Saving
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
nums =
[3,1,2,4]
Source
23°C
Clear
Search
ENG
IN
23:18
23-01-2025


LeetCode - The World's Leadin
Progress - LeetCode
Optimal Division - LeetCode
Download file | iLovePDF
leetcode.com/problems/optimal-division/description/
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
553. Optimal Division
Solved
Medium
Topics
Companies
You are given an integer array nums. The adjacent integers in nums will perform the float division.
For example, for nums = [2,3,4], we will evaluate the expression "2/3/4".
However, you can add any number of parenthesis at any position to change the priority of operations. You want to add these parentheses such the value of the expression after the evaluation is maximum.
Return the corresponding expression that has the maximum value in string format.
Note: your expression should not contain redundant parenthesis.
Example 1:
Input: nums = [1000,100,10,2]
Output: "1000/(100/10/2)"
Explanation: 1000/(100/10/2) = 1000/((100/10)/2) = 200
However, the bold parenthesis in "1000/((100/10)/2)" are redundant since they do not influence the operation priority.
So you should return "1000/(100/10/2)".
Other cases:
1000/(100/10)/2 = 50
1000/(100/(10/2)) = 50
1000/100/10/2 = 0.5
389
10
0 Online
Code
C++
Auto
class Solution {
public:
string optimalDivision(vector<int>& nums) {
int n = nums.size();
string expr;
for (int i = 0; i < n; i++) {
if (i > 0) {
expr += "/";
}
if (i == 1 && n > 2) {
expr += "(";
Restored from local
Upgrade to Cloud Saving
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
nums =
[1000,100,10,2]
Source
23°C
Clear
Search
ENG
IN
23:18
23-01-2025


LeetCode - The World's Leadin
Progress - LeetCode
K-diff Pairs in an Array - LeetCo
Download file | iLovePDF
leetcode.com/problems/k-diff-pairs-in-an-array/description/
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
532. K-diff Pairs in an Array
Solved
Medium
Topics
Companies
Given an array of integers nums and an integer k, return the number of unique k-diff pairs in the array.
A k-diff pair is an integer pair (nums[i], nums[j]), where the following are true:
0 <= i, j < nums.length
i != j
|nums[i] - nums[j]| == k
Notice that |val| denotes the absolute value of val.
Example 1:
Input: nums = [3,1,4,1,5], k = 2
Output: 2
Explanation: There are two 2-diff pairs in the array, (1, 3) and (3, 5).
Although we have two 1s in the input, we should only return the number of unique pairs.
Example 2:
Input: nums = [1,2,3,4,5], k = 1
Output: 4
4K
37
20 Online
Code
C
Auto
#include <stdlib.h>
int cmp(const void* a, const void* b) {
return (*(int*)a - *(int*)b);
}
int findPairs(int* nums, int numsSize, int k) {
qsort(nums, numsSize, sizeof(int), cmp);
int count = 0, left = 0, right = 1;
while (right < numsSize) {
Restored from local
Upgrade to Cloud Saving
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
Case 3
nums =
[3,1,4,1,5]
k =
2
Source
23°C
Clear
Search
ENG
IN
23:18
23-01-2025


LeetCode - The World's Leadin
Progress - LeetCode
Plus One - LeetCode
Download file | iLovePDF
leetcode.com/problems/plus-one/description/
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
66. Plus One
Solved
Easy
Topics
Companies
You are given a large integer represented as an integer array digits, where each digits[i] is the ith digit of the integer. The digits are ordered from most significant to least significant in left-to-right order. The large integer does not contain any leading 0's.
Increment the large integer by one and return the resulting array of digits.
Example 1:
Input: digits = [1,2,3]
Output: [1,2,4]
Explanation: The array represents the integer 123.
Incrementing by one gives 123 + 1 = 124.
Thus, the result should be [1,2,4].
Example 2:
Input: digits = [4,3,2,1]
Output: [4,3,2,2]
Explanation: The array represents the integer 4321.
Incrementing by one gives 4321 + 1 = 4322.
Thus, the result should be [4,3,2,2].
Example 3:
9.9K
275
227 Online
Code
C
Auto
#include <stdio.h>
#include <stdlib.h>
int* plusOne(int* digits, int digitsSize, int* returnSize) {
int carry = 1;
for (int i = digitsSize - 1; i >= 0; i--) {
digits[i] += carry;
if (digits[i] == 10) {
digits[i] = 0;
carry = 1;
Restored from local
Upgrade to Cloud Saving
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
Case 3
digits =
[1,2,3]
Source
23°C
Clear
Search
ENG
IN
23:18
23-01-2025


LeetCode - The World's Leadin
Progress - LeetCode
Find the Duplicate Number - Le
Plus One - LeetCode
Download file | iLovePDF
leetcode.com/problems/find-the-duplicate-number/description/
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
287. Find the Duplicate Number
Solved
Medium
Topics
Companies
Given an array of integers nums containing n + 1 integers where each integer is in the range [1, n] inclusive.
There is only one repeated number in nums, return this repeated number.
You must solve the problem without modifying the array nums and using only constant extra space.
Example 1:
Input: nums = [1,3,4,2,2]
Output: 2
Example 2:
Input: nums = [3,1,3,4,2]
Output: 3
Example 3:
Input: nums = [3,3,3,3,3]
Output: 3
Constraints:
23.9K
352
160 Online
Code
C
Auto
int findDuplicate(int* nums, int numsSize) {
int slow = nums[0], fast = nums[0];
do {
slow = nums[slow];
fast = nums[nums[fast]];
} while (slow != fast);
slow = nums[0];
while (slow != fast) {
slow = nums[slow];
fast = nums[fast];
Restored from local
Upgrade to Cloud Saving
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
Case 3
nums =
[1,3,4,2,2]
Source
23°C
Clear
Search
ENG
IN
23:19
23-01-2025


LeetCode - The World's Le
Progress - LeetCode
Happy Number - LeetCode
Find the Duplicate Numbe
Plus One - LeetCode
Download file | iLovePDF
leetcode.com/problems/happy-number/description/
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
202. Happy Number
Solved
Easy
Topics
Companies
Write an algorithm to determine if a number n is happy.
A happy number is a number defined by the following process:
Starting with any positive integer, replace the number by the sum of the squares of its digits.
Repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1.
Those numbers for which this process ends in 1 are happy.
Return true if n is a happy number, and false if not.
Example 1:
Input: n = 19
Output: true
Explanation:
1² + 9² = 82
8² + 2² = 68
6² + 8² = 100
1² + 0² + 0² = 1
Example 2:
10.8K
236
0 Online
Code
C
Auto
#include <stdbool.h>
int getSumOfSquares(int num);
bool isHappy(int n) {
int slow = n, fast = n;
do {
slow = getSumOfSquares(slow);
fast = getSumOfSquares(getSumOfSquares(fast));
} while (slow != fast);
Restored from local
Upgrade to Cloud Saving
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
n =
19
Source
23°C
Clear
Search
ENG
IN
23:19
23-01-2025

LeetCode - The Wor | Progress - LeetCode | Balanced Binary Tree | Happy Number - Le | Find the Duplicate N | Plus One - LeetCode | Download file | iLov

leetcode.com/problems/balanced-binary-tree/description/

Problem List | Run | Submit | Premium

Description | Editorial | Solutions | Submissions

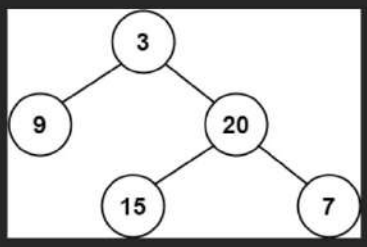
110. Balanced Binary Tree

Solved

Easy | Topics | Companies

Given a binary tree, determine if it is height-balanced.

Example 1:



Input: root = [3,9,20,null,null,15,7]
Output: true

Example 2:

11.1K | 132 | 168 Online

Code

C | Auto

```c
#include <stdbool.h>
#include <stdlib.h>

int height(struct TreeNode* root) {
    if (root == NULL) return 0;

    int leftHeight = height(root->left);
    if (leftHeight == -1) return -1;

    int rightHeight = height(root->right);
    if (rightHeight == -1) return -1;
```

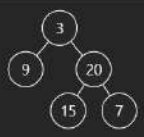
Restored from local | Upgrade to Cloud Saving | Ln 1, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3

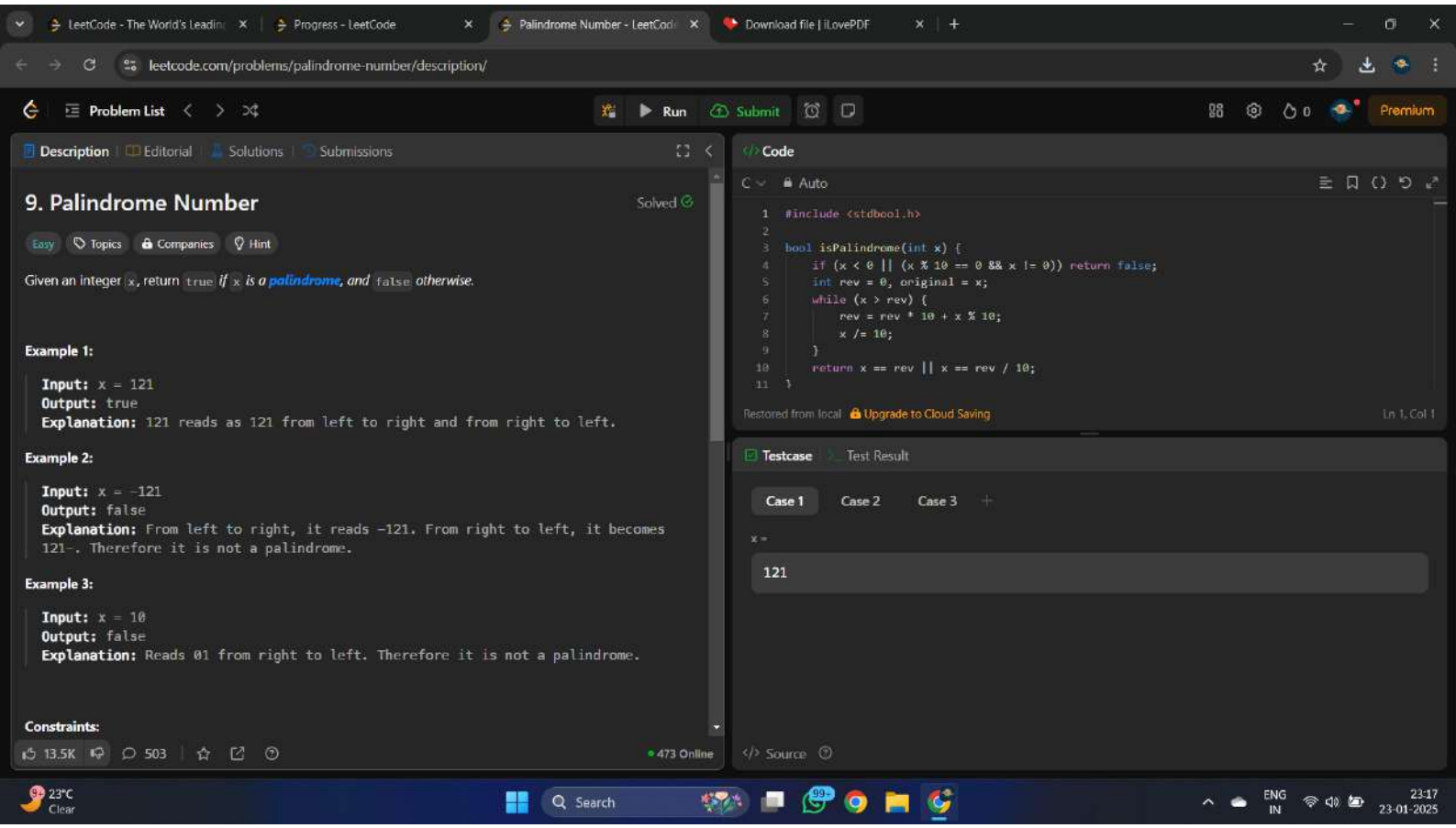
root =

[3,9,20,null,null,15,7]



Source

23°C Clear | Search | ENG IN | 23:19 23-01-2025


LeetCode - The World's Leadin
Progress - LeetCode
Palindrome Number - LeetCod
Download file | iLovePDF
leetcode.com/problems/palindrome-number/description/
Problem List
Run
Submit
Premium
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.
Constraints:
13.5K
503
473 Online
Code
C
Auto
#include <stdbool.h>
bool isPalindrome(int x) {
if (x < 0 || (x % 10 == 0 && x != 0)) return false;
int rev = 0, original = x;
while (x > rev) {
rev = rev * 10 + x % 10;
x /= 10;
}
return x == rev || x == rev / 10;
}
Restored from local
Upgrade to Cloud Saving
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
Case 3
x =
121
Source
23°C
Clear
Search
ENG
IN
23:17
23-01-2025