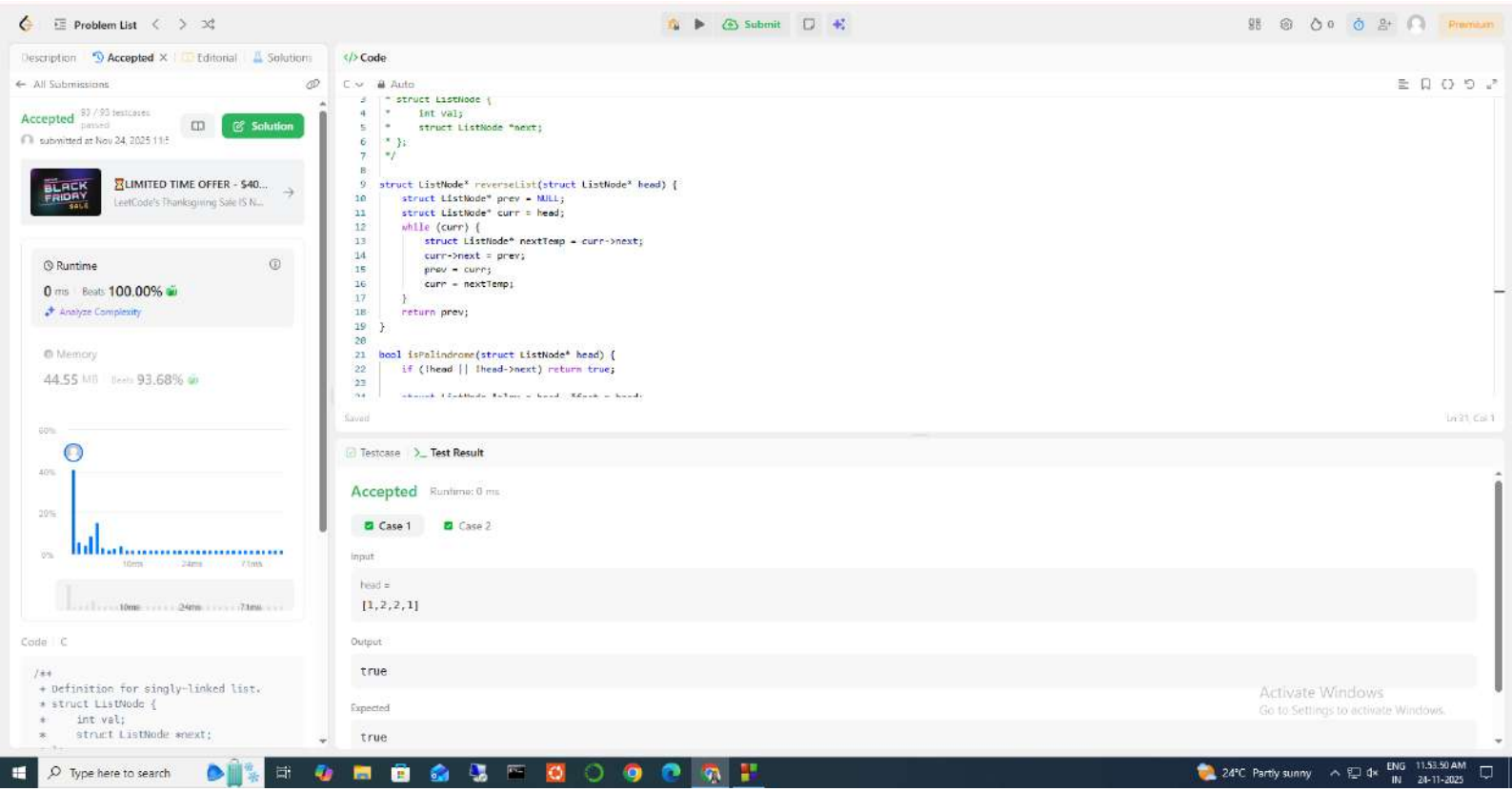
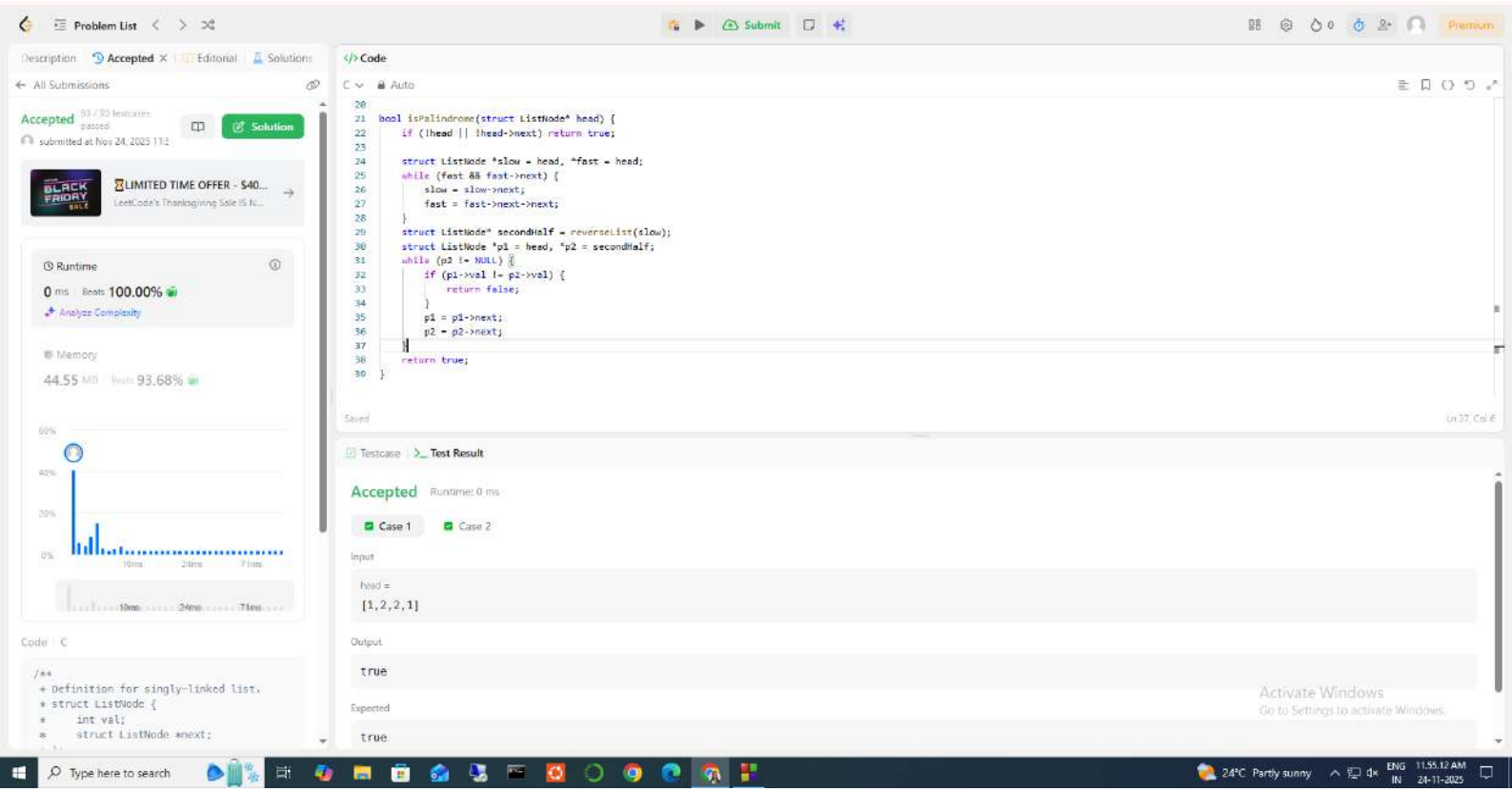

Problem List
Submit
Premium
Description
Accepted
Editorial
Solutions
All Submissions
Accepted
Solution
LIMITED TIME OFFER - $40...
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
44.55 MB | Beats 93.68%
Code | C
/**
* Definition for singly-linked list.
* struct ListNode {
*   int val;
*   struct ListNode *next;
Code
C
Auto
* struct ListNode {
*   int val;
*   struct ListNode *next;
* };
*/
struct ListNode* reverseList(struct ListNode* head) {
struct ListNode* prev = NULL;
struct ListNode* curr = head;
while (curr) {
struct ListNode* nextTemp = curr->next;
curr->next = prev;
prev = curr;
curr = nextTemp;
}
return prev;
}
bool isPalindrome(struct ListNode* head) {
if (!head || !head->next) return true;
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
head =
[1,2,2,1]
Output
true
Expected
true
Activate Windows
Go to Settings to activate Windows.
Type here to search
24°C Partly sunny
ENG
IN
11:53:50 AM
24-11-2025


Problem List
Submit
Premium
Description
Accepted
Editorial
Solutions
All Submissions
Accepted
submitted at Nov 24, 2025 11:1
Solution
LIMITED TIME OFFER - $40...
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
44.55 MB | Beats 93.68%
Code | C
/**
* Definition for singly-linked list.
* struct ListNode {
*   int val;
*   struct ListNode *next;
Code
C
Auto
bool isPalindrome(struct ListNode* head) {
if (!head || !head->next) return true;
struct ListNode *slow = head, *fast = head;
while (fast && fast->next) {
slow = slow->next;
fast = fast->next->next;
}
struct ListNode* secondHalf = reverseList(slow);
struct ListNode *p1 = head, *p2 = secondHalf;
while (p2 != NULL) {
if (p1->val != p2->val) {
return false;
}
p1 = p1->next;
p2 = p2->next;
}
return true;
}
Saved
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
head =
[1,2,2,1]
Output
true
Expected
true
Activate Windows
Go to Settings to activate Windows.
Type here to search
24°C Partly sunny
ENG
IN
11:55:12 AM
24-11-2025