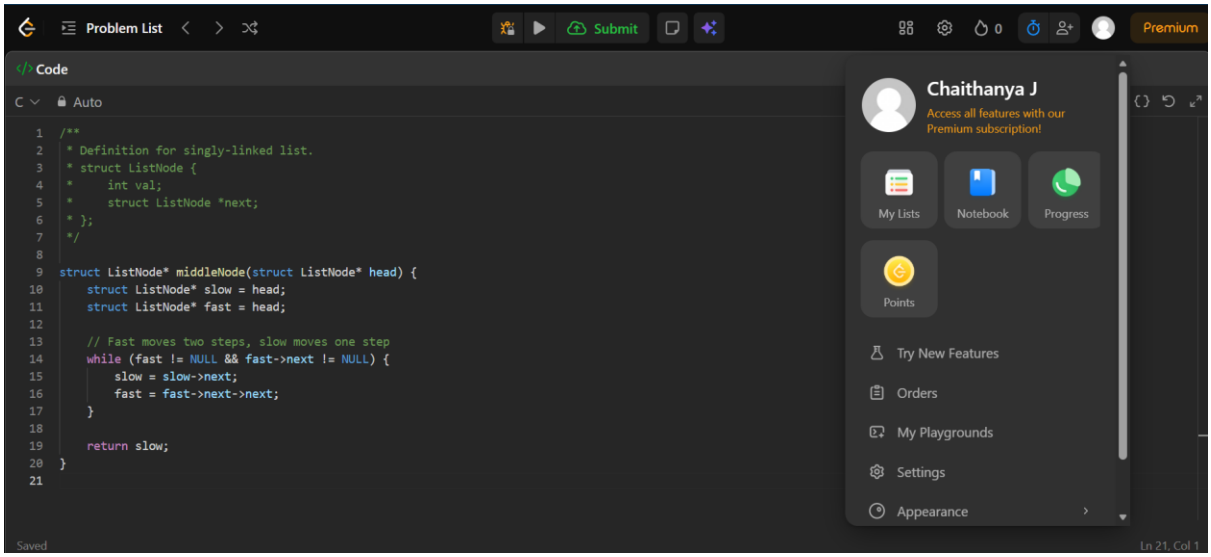
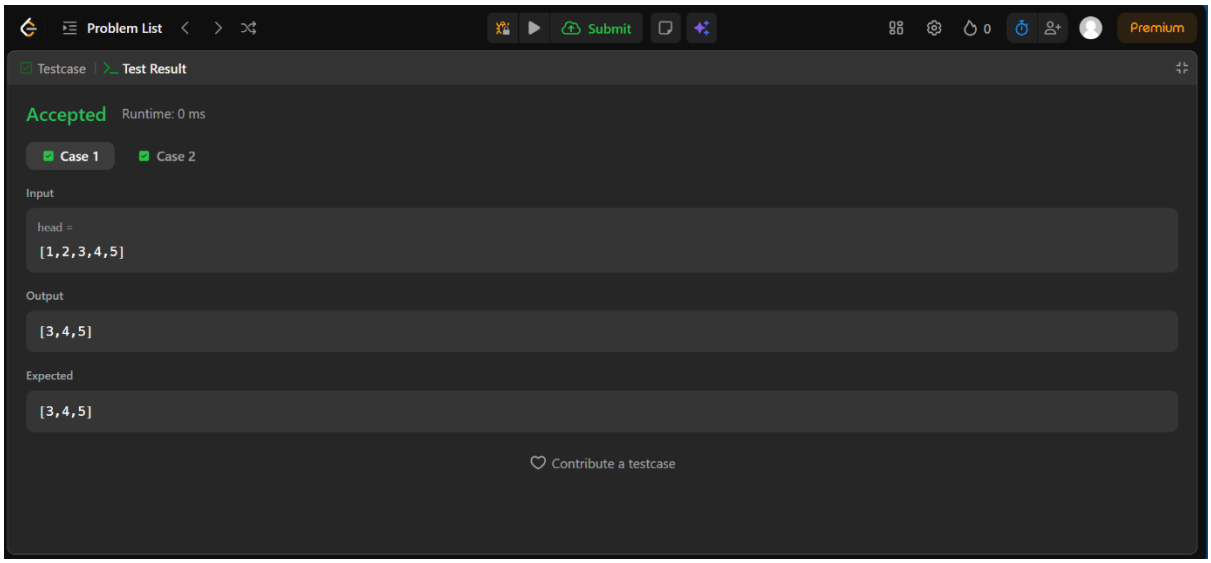
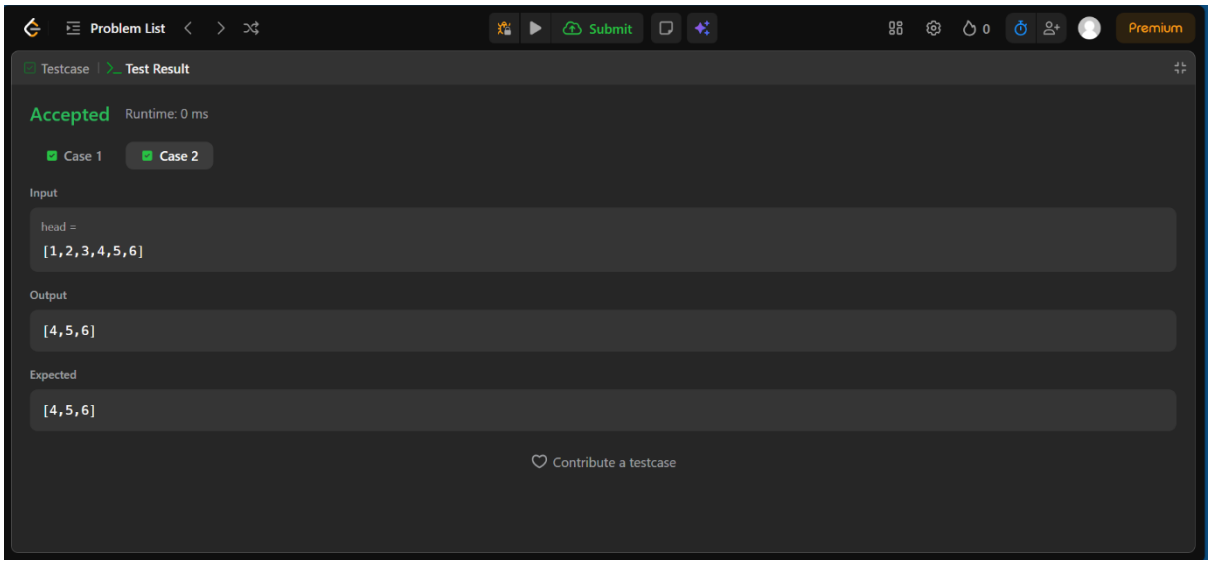
Leetcode:

```
struct ListNode* middleNode(struct ListNode* head) {
    struct ListNode* slow = head;
    struct ListNode* fast = head;

    // Fast moves two steps, slow moves one step
    while (fast != NULL && fast->next != NULL) {
        slow = slow->next;
        fast = fast->next->next;
    }

    return slow;
}
```



OUTPUT:





OBSERVATION:

Page No:
Date.

21/2/25

Prg-46
LeetCode.

Middle of the Linked List :-

Code :-

```
Struct ListNode * middleNode (struct ListNode * head)
{
    struct ListNode * slow = head;
    struct ListNode * fast = head;

    while (fast != NULL && fast -> next != NULL)
    {
        slow = slow -> next;
        fast = fast -> next -> next;
    }

    return slow;
}
```

O/p Case 1 :-

Input => head = [1,2,3,4,5]

Output => [3,4,5]

Case 2 :-

I/p => head = [1,2,3,4,5,6]

O/p => [4,5,6].