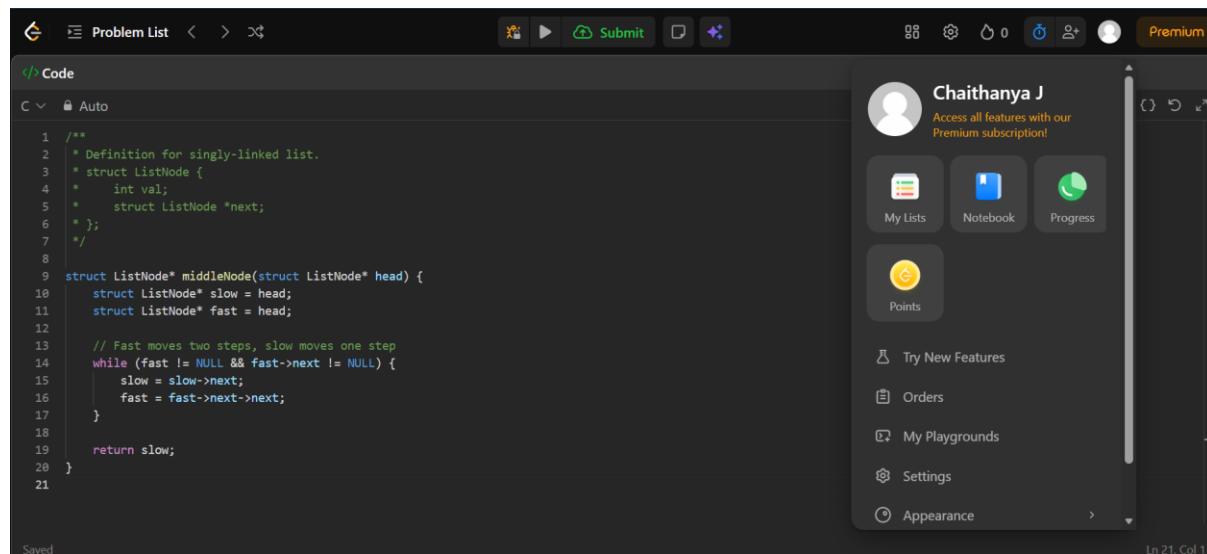


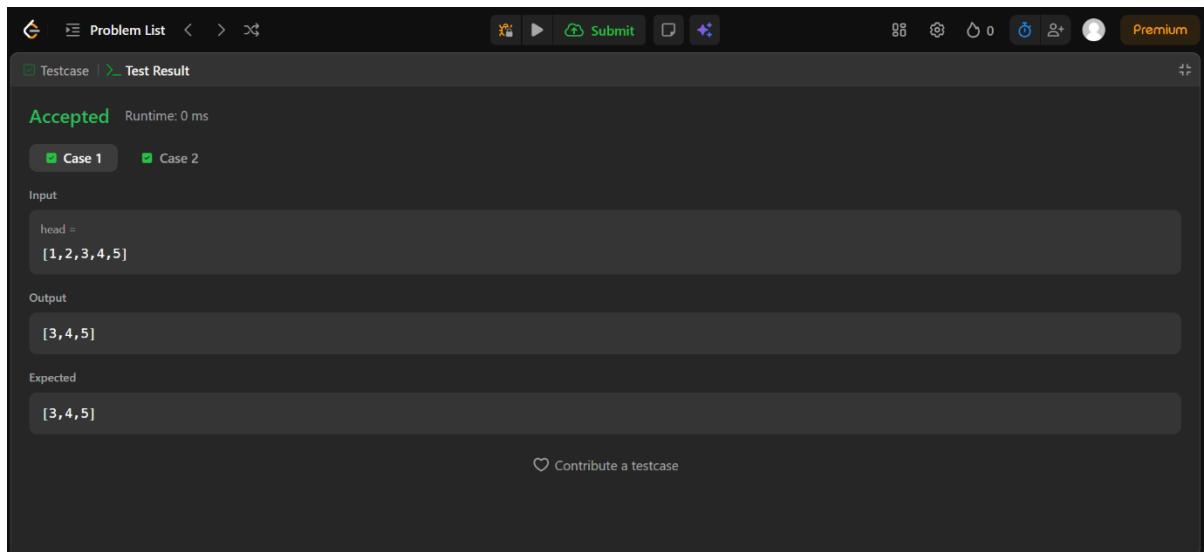
Leetcode:

```
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;  
}
```



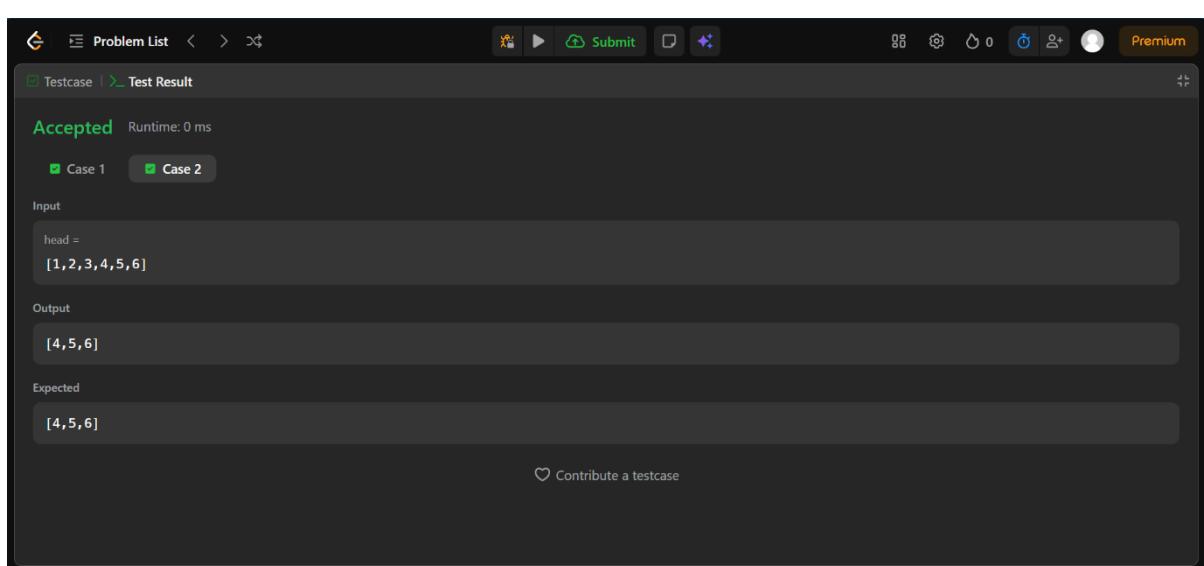
The screenshot shows the Leetcode IDE interface. On the left, there is a code editor window with the C language selected. The code is the same as provided above. On the right, there is a sidebar with the user's profile picture and name, "Chaithanya J". Below the profile, there are several buttons for "My Lists", "Notebook", "Progress", and "Points". At the bottom of the sidebar, there are links for "Try New Features", "Orders", "My Playgrounds", "Settings", and "Appearance". The status bar at the bottom indicates "Saved" and "In 21, Col 1".

OUTPUT:



Screenshot of a programming test result interface. The status is "Accepted" with a runtime of 0 ms. Two test cases are shown: Case 1 and Case 2. Both cases have the same input and output, which is [3,4,5]. The expected output is also [3,4,5]. A "Contribute a testcase" button is visible at the bottom.

Input	Output	Expected
head = [1,2,3,4,5]	[3,4,5]	[3,4,5]



Screenshot of a programming test result interface. The status is "Accepted" with a runtime of 0 ms. Two test cases are shown: Case 1 and Case 2. Both cases have the same input and output, which is [4,5,6]. The expected output is also [4,5,6]. A "Contribute a testcase" button is visible at the bottom.

Input	Output	Expected
head = [1,2,3,4,5,6]	[4,5,6]	[4,5,6]

OBSERVATION:

21/10/2025

Prg-4 b
LeetCode

Page No:
Date:

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 \Rightarrow head $\Rightarrow [1, 2, 3, 4, 5]$

Output $\Rightarrow [3, 4, 5]$

Case 2 :-

I/P \Rightarrow head $\Rightarrow [1, 2, 3, 4, 5, 6]$

O/P $\Rightarrow [4, 5, 6]$

