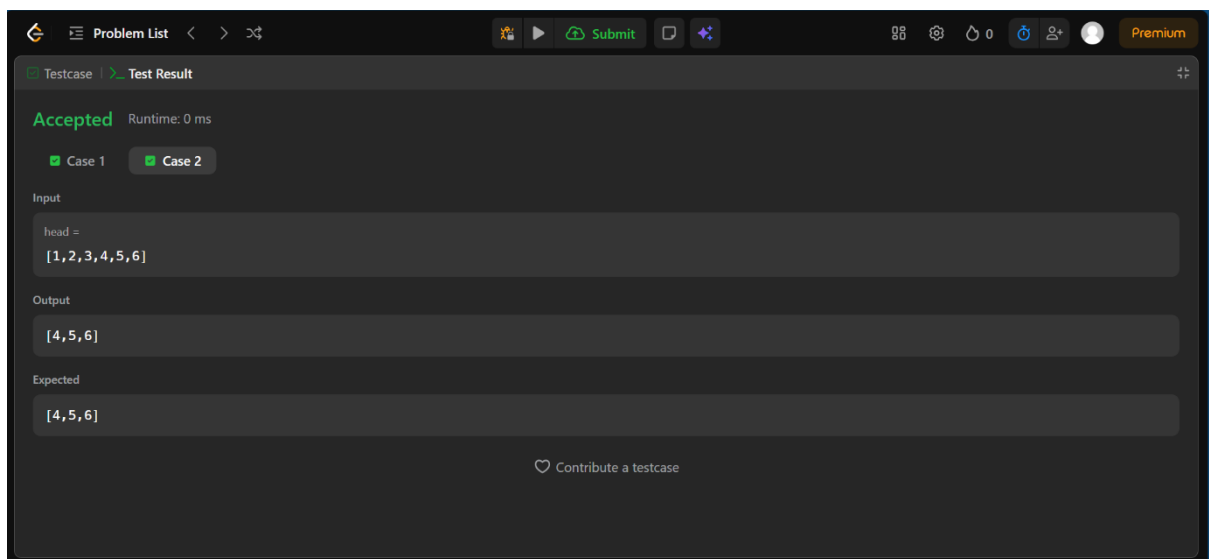
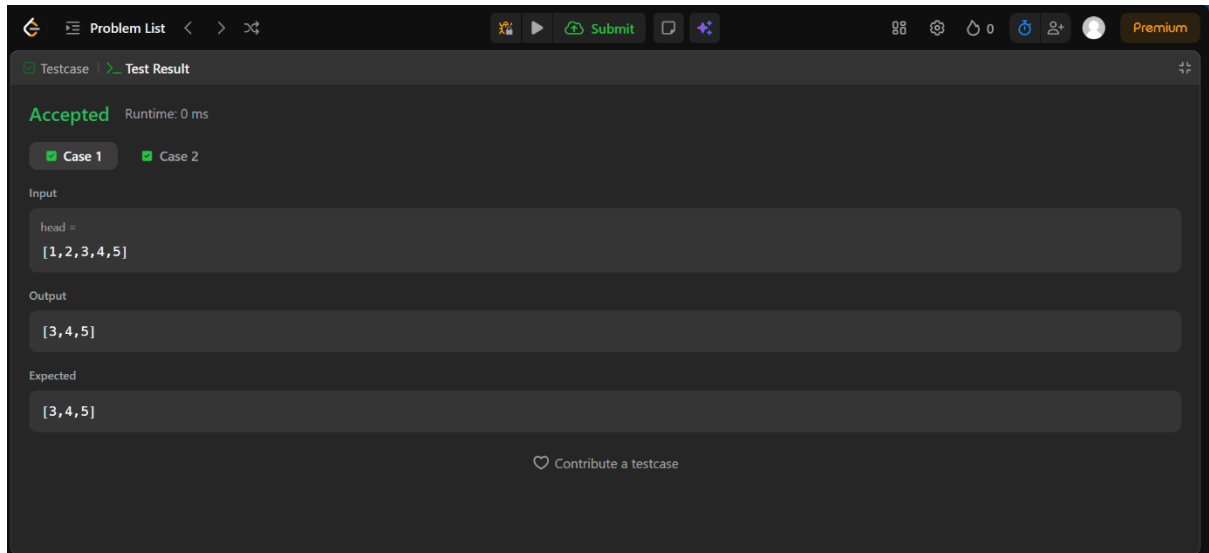


Leetcode:

```
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;  
}
```

```
1  /**  
2   * Definition for singly-linked list.  
3   * struct ListNode {  
4   *     int val;  
5   *     struct ListNode *next;  
6   * };  
7   */  
8  
9  struct ListNode* middleNode(struct ListNode* head) {  
10     struct ListNode* slow = head;  
11     struct ListNode* fast = head;  
12  
13     // Fast moves two steps, slow moves one step  
14     while (fast != NULL && fast->next != NULL) {  
15         slow = slow->next;  
16         fast = fast->next->next;  
17     }  
18  
19     return slow;  
20 }  
21
```

OUTPUT:



OBSERVATION:

21/12/15

Prg-4b
LinkedList

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

Output \Rightarrow [3, 4, 5]

Case 2 :-

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

O/p \Rightarrow [4, 5, 6]

