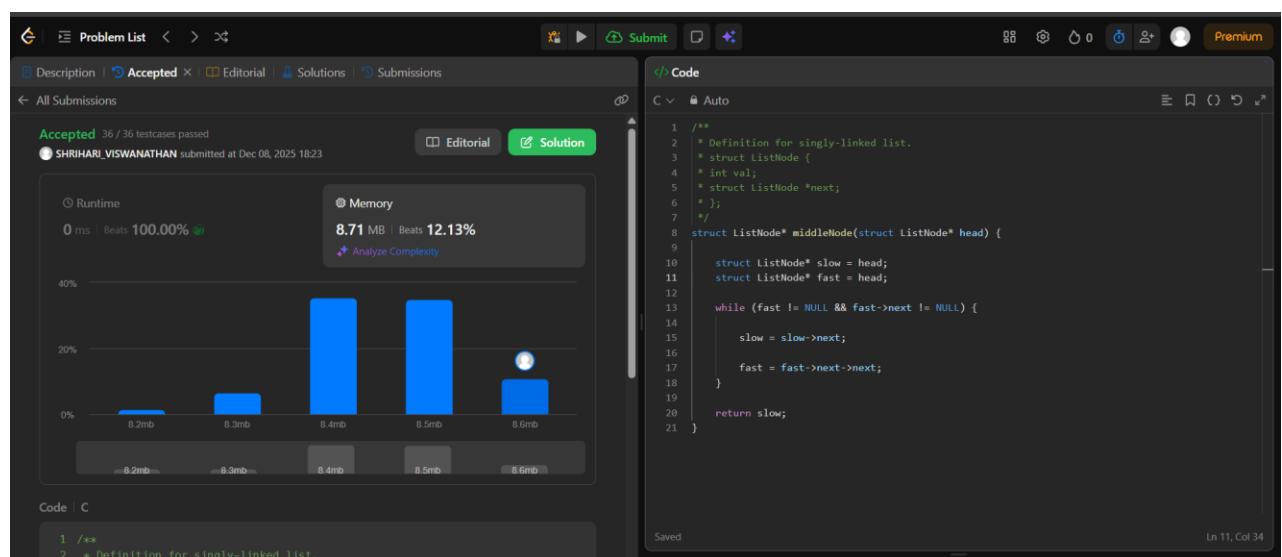


Lab_04B_LEETCODE_ 876. Middle of the Linked List

```
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;  
}
```



The screenshot shows a LeetCode submission page for problem 876. The submission was made by SHRIHARI VISWANATHAN on Dec 08, 2025 at 18:23. It has been accepted and passed all 36 testcases. The runtime is 0 ms, beating 100.00% of submissions. The memory usage is 8.71 MB, which is 12.13% faster than the average. Below the stats, there's a bar chart showing memory usage across different memory ranges: 0.2mb, 0.3mb, 0.4mb, 0.5mb, and 0.6mb. The 0.4mb bin is the largest. The code editor on the right contains the C implementation provided above.

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

