

LeetCode Problem – 876

The screenshot shows the LeetCode problem page for "876. Middle of the Linked List". The problem description asks for the middle node of a singly linked list. If there are two middle nodes, return the second one. Example 1 shows a list [1, 2, 3, 4, 5] with node 3 highlighted. Example 2 shows a list [1, 2, 3, 4, 5, 6] with node 4 highlighted. The code editor contains C++ code for finding the middle node.

```
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     struct ListNode *i=head ,*j=head;
10    while(j !=NULL && j->next !=NULL)
11    {
12        i=i->next;
13        j=j->next->next;
14    } head=i ;
15    return  head;
16 }
```

Solution:

The screenshot shows the LeetCode code editor with the solution for problem 876. The code is a C++ implementation of the two-pointer technique to find the middle node of a singly-linked list.

```
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     struct ListNode *i=head ,*j=head;
10    while(j !=NULL && j->next !=NULL)
11    {
12        i=i->next;
13        j=j->next->next;
14    } head=i ;
15    return  head;
16 }
```