Given a non-empty, singly linked list with head node, return a middle node of linked list. If there are two middle nodes, return the second middle node.

C++:

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* middleNode(ListNode* head) {

    }
};
```

Sample input 1:
[1, 2, 3, 4, 5]

Sample output 2:
Node 3 from this list (Serialization: [3,4,5]) The returned node has value 3. (The judge's serialization of this node is [3,4,5]). Note that we returned a ListNode object ans, such that: ans.val = 3, ans.next.val = 4, ans.next.next.val = 5, and ans.next.next.next = NULL.

Sample input 1:
[1, 2, 3, 4, 5, 6]

Sample output 2:
Node 4 from this list (Serialization: [4,5,6]) Since the list has two middle nodes with values 3 and 4, we return the second one.