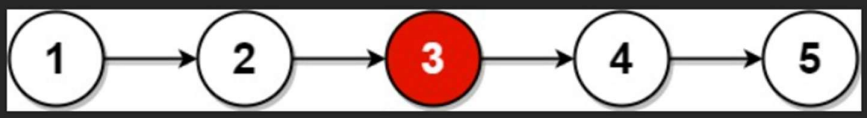
Given the `head` of a singly linked list, return *the middle node of the linked list.*

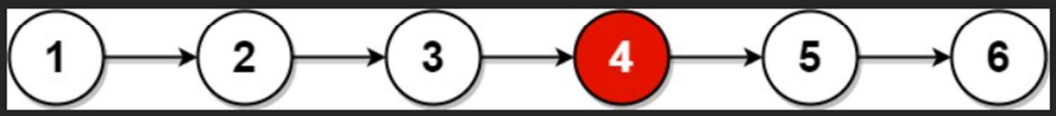
If there are two middle nodes, return **the second middle** node.

Example 1:



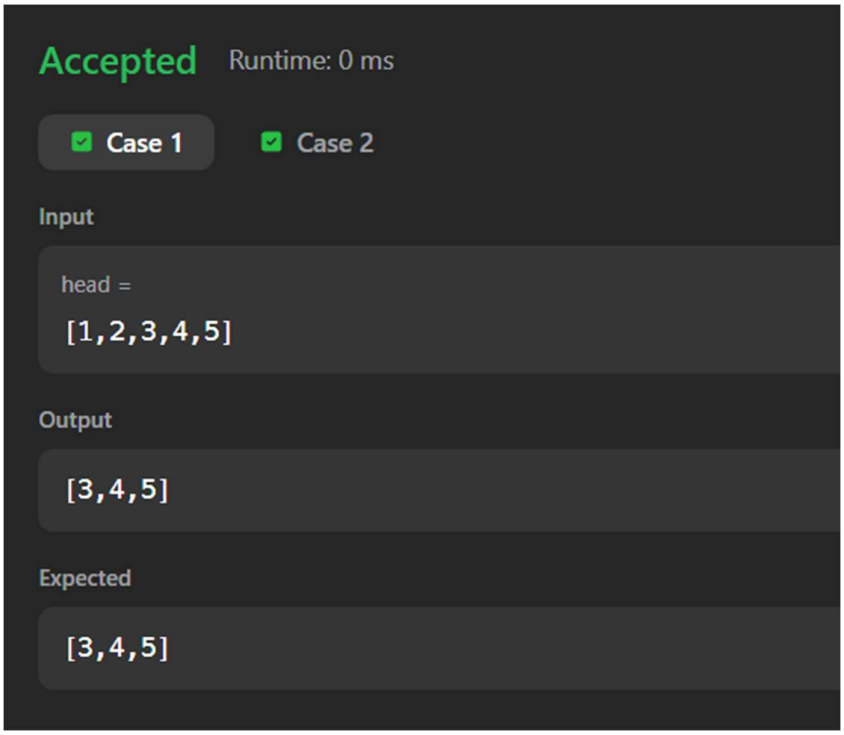

```
Input: head = [1,2,3,4,5]
Output: [3,4,5]
Explanation: The middle node of the list is node 3.
```

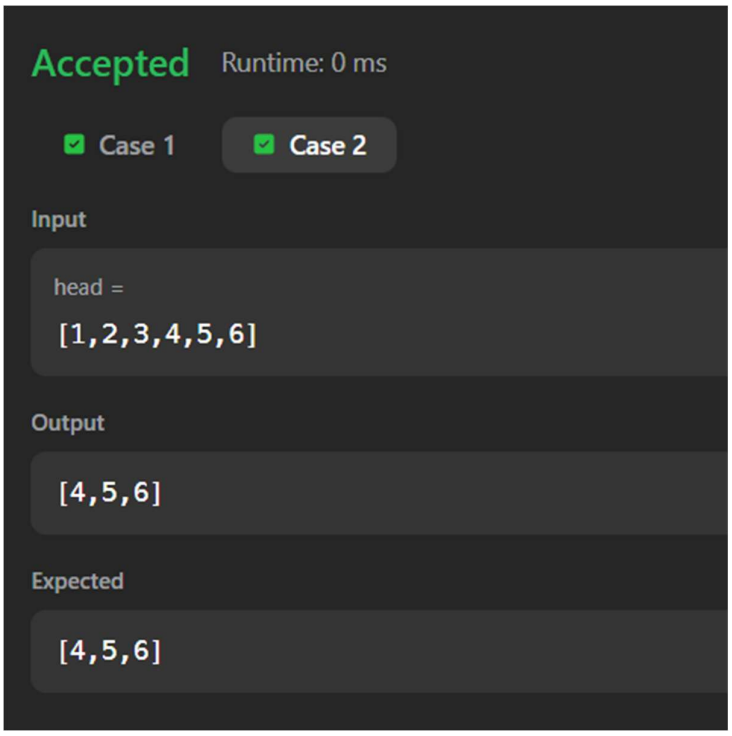
Example 2:




```
Input: head = [1,2,3,4,5,6]
Output: [4,5,6]
Explanation: Since the list has two middle nodes with values 3 and 4, we return
the second one.
```

```c
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode* middleNode(struct ListNode* head) {
    struct ListNode* temp=head;
    struct ListNode* prev=head;
    while(prev!=NULL && prev->next!=NULL){
            temp=temp->next;
            prev=prev->next->next;
    }
    return temp;
}
```


Accepted
Runtime: 0 ms
Case 1
Case 2
Input
head =
[1,2,3,4,5]
Output
[3,4,5]
Expected
[3,4,5]


Accepted
Runtime: 0 ms
Case 1
Case 2
Input
head =
[1,2,3,4,5,6]
Output
[4,5,6]
Expected
[4,5,6]