

## 206. Reverse Linked List

Given the head of a singly linked list, reverse the list, and return *the reversed list*.

```
/*
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode* reverseList(struct ListNode* head) {
    struct ListNode *prev = NULL;
    struct ListNode *curr = head;
    struct ListNode *next = NULL;

    while (curr != NULL) {
        next = curr->next;
        curr->next = prev;
        prev = curr;
        curr = next;
    }
    return prev;
}
```

The screenshot shows a programming test interface with the following details:

- Testcase**: The current tab is "Testcase".
- Output**: The status is "Accepted" with a runtime of "0 ms".
- Input**: The input is defined as `head = [1,2,3,4,5]`.
- Output**: The output is shown as ` [5,4,3,2,1]`.
- Expected**: The expected output is also shown as ` [5,4,3,2,1]`.
- Buttons**: There are checkboxes for "Case 1", "Case 2", and "Case 3", all of which are checked.
- Submit**: A "Submit" button is visible in the top right.
- Contribute**: A link "Contribute a testcase" is located at the bottom right.

## 876. Middle of the Linked List

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.

```
/*
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
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;
}
```

Testcase |  Test Result

**Accepted** Runtime: 0 ms

Case 1  Case 2

Input

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

Output

```
[3,4,5]
```

Expected

```
[3,4,5]
```

 Contribute a testcase