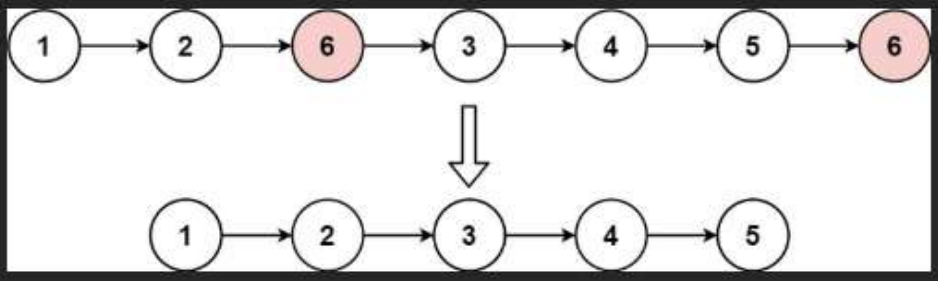
203. Remove Linked List Elements

Solved

Easy | Topics | Companies

Given the `head` of a linked list and an integer `val`, remove all the nodes of the linked list that has `Node.val == val`, and return *the new head*.

Example 1:




```
Input: head = [1,2,6,3,4,5,6], val = 6
Output: [1,2,3,4,5]
```

Example 2:

```
Input: head = [], val = 1
Output: []
```

Example 3:

```
Input: head = [7,7,7,7], val = 7
Output: []
```

Constraints:

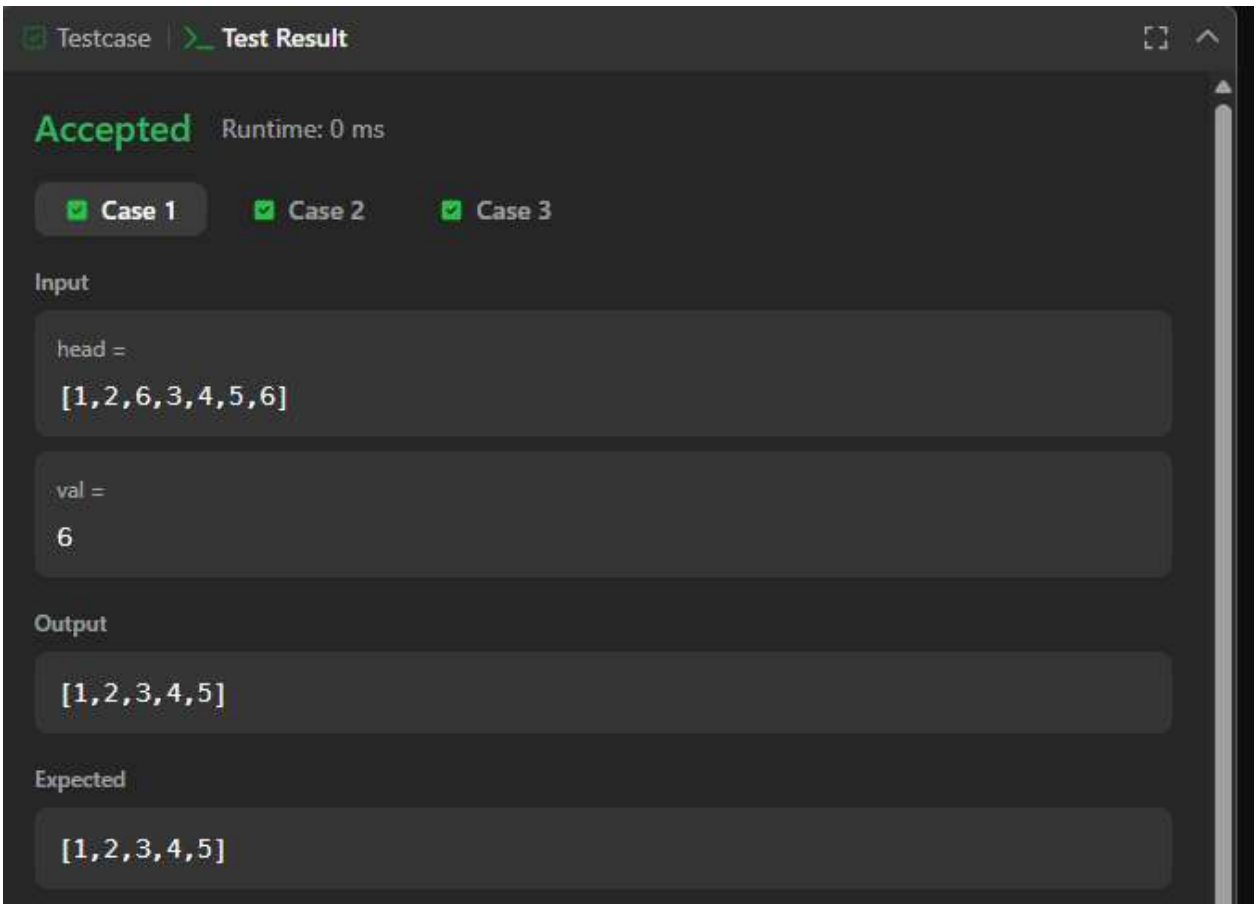
- The number of nodes in the list is in the range `[0, 10^4]`.
- `1 <= Node.val <= 50`
- `0 <= val <= 50`

C Auto

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

    if (head == NULL) return head;

    struct ListNode* current = head;
    while (current->next != NULL) {
        if (current->next->val == val) {
            temp = current->next;
            current->next = current->next->next;
            free(temp);
        } else {
            current = current->next;
        }
    }

    return head;
}
```

Saved Ln 1, Col 1

Saved Ln 1, Col 1


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