

Lab_05_LEETCODE_203. Remove Linked List Elements

```
/*
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode* removeElements(struct ListNode* head, int val) {
    struct ListNode* dummy = (struct
ListNode*)malloc(sizeof(struct ListNode));
    if (dummy == NULL) {
        return head;
    }

    dummy->val = 0;
    dummy->next = head;
    struct ListNode* current = dummy;

    while (current->next != NULL) {

        if (current->next->val == val) {

            struct ListNode* node_to_delete = current->next;
            current->next = current->next->next;
            free(node_to_delete);

        } else {
            current = current->next;
        }
    }

    struct ListNode* new_head = dummy->next;
    free(dummy);

    return new_head;
}
```

LeetCode - Remove Linked List Elements - LeetCode

Problem List | Accepted | Editorial | Solutions | Submissions

Accepted 66 / 66 testcases passed

SHRIHARI_VISWANATHAN submitted at Dec 08, 2025 18:32

Runtime: 0 ms Beats 100.00% | Memory: 12.96 MB Beats 5.19%

Analyze Complexity

Runtime Analysis: A chart showing execution time distribution. The y-axis ranges from 0% to 100%. The x-axis shows time intervals: 1ms, 2ms, 3ms, and 4ms. The distribution is heavily skewed towards 0ms, with a large blue bar reaching nearly 100%.

Code | C

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     struct ListNode *next;
6  * };
7 */
8 struct ListNode* removeElements(struct ListNode* head, int val) {
```

Editorial | Solution

C Auto

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     struct ListNode *next;
6  * };
7 */
8 struct ListNode* removeElements(struct ListNode* head, int val) {
9     struct ListNode* dummy = (struct ListNode*)malloc(sizeof(struct ListNode));
10    if (dummy == NULL) {
11        return head;
12    }
13
14    dummy->val = 0;
15    dummy->next = head;
16    struct ListNode* current = dummy;
17
18    while (current->next != NULL) {
19
20        if (current->next->val == val) {
21
22            struct ListNode* node_to_delete = current->next;
23            current->next = current->next->next;
24            free(node_to_delete);
25
26        } else {
27            current = current->next;
28        }
29
30
31    }
32
33    struct ListNode* new_head = dummy->next;
34    free(dummy);
35
36    return new_head;
37}
```

Saved | Line 12, Col 6

Testcase | Test Result