

Leetcode:

```
struct ListNode* removeElements(struct ListNode* head, int val) {  
    // Create a dummy node pointing to head  
    struct ListNode* dummy = (struct ListNode*)malloc(sizeof(struct ListNode));  
    dummy->next = head;  
  
    struct ListNode* prev = dummy;  
    struct ListNode* curr = head;  
  
    while (curr != NULL) {  
        if (curr->val == val) {  
            // Skip this node  
            prev->next = curr->next;  
            free(curr);  
            curr = prev->next;  
        } else {  
            prev = curr;  
            curr = curr->next;  
        }  
    }  
  
    struct ListNode* newHead = dummy->next;  
    free(dummy);  
    return newHead;  
}
```

The screenshot shows a code editor interface with a dark theme. On the left, there is a code editor window titled "Code" containing C code. The code defines a function `removeElements` that removes all occurrences of a given value `val` from a linked list. It uses a dummy node to handle the head of the list. The code is as follows:

```
1 struct ListNode* removeElements(struct ListNode* head, int val) {
2     // Create a dummy node pointing to head
3     struct ListNode* dummy = (struct ListNode*)malloc(sizeof(struct ListNode));
4     dummy->next = head;
5
6     struct ListNode* prev = dummy;
7     struct ListNode* curr = head;
8
9     while (curr != NULL) {
10         if (curr->val == val) {
11             // Skip this node
12             prev->next = curr->next;
13             free(curr);
14             curr = prev->next;
15         } else {
16             prev = curr;
17             curr = curr->next;
18         }
19     }
20
21     struct ListNode* newHead = dummy->next;
22     free(dummy);
23     return newHead;
}
```

On the right side of the interface, there is a sidebar for a user named "Chaithanya J". The sidebar includes links to "My Lists", "Notebook", "Progress", "Points", "Orders", "My Playgrounds", "Settings", and "Appearance". There is also a "Premium" subscription banner.

OUTPUT:

The screenshot shows a "Test Result" page. At the top, it says "Accepted" with a runtime of "0 ms". Below that, there are three test cases: "Case 1" (checked), "Case 2" (unchecked), and "Case 3" (unchecked). The "Input" section shows the input array as [1,2,6,3,4,5,6]. The "Output" section shows the output array as [1,2,3,4,5]. The "Expected" section shows the expected array as [1,2,3,4,5]. At the bottom, there is a link to "Contribute a testcase".

The screenshot shows a "Test Result" page from a programming platform. At the top, it says "Accepted" with a runtime of "0 ms". There are three test cases listed: Case 1, Case 2, and Case 3, all of which are checked. The "Input" section shows the following code snippet:

```
head =  
[]
```

The "val" field contains the value "1". The "Output" section shows the result as "[]". The "Expected" section also shows "[]". At the bottom right, there is a link to "Contribute a testcase".

This screenshot shows the same "Test Result" page after changing the input. The "Input" section now contains:

```
head =  
[7, 7, 7, 7]
```

The "val" field still contains the value "7". The "Output" section shows the result as "[]". The "Expected" section also shows "[]". The "Contribute a testcase" link is present at the bottom right.

OBSERVATION:

21/12/25

Pg- 5b.

Loctcode

Page No:	
Date:	

Remove Linked List Elements:-

Code:-

```
struct ListNode* removeElements(struct ListNode* head, int val) {
    struct ListNode* dummy = (struct ListNode*) malloc(sizeof(struct ListNode));
    dummy->next = head;
    struct ListNode* prev = dummy;
    struct ListNode* curr = head;

    while (curr != NULL) {
        if (curr->val == val) {
            prev->next = curr->next;
            free(curr);
            curr = prev->next;
        } else {
            prev = curr;
            curr = curr->next;
        }
    }

    struct ListNode* newHead = dummy->next;
    free(dummy);
    return newHead;
```

O/p

Case 1 :-

I/p \Rightarrow head = [1, 2, 3, 4, 5, 6]

val = 6

O/p \Rightarrow [1, 2, 3, 4, 5]

Case 2 :-

I/p \Rightarrow head = []

val =

Case 3 :-

I/p : head = [7, 7, 7, 7]

val = 7

O/p \Rightarrow []

~~8~~