

203.Remove Linked List Elements

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
/* Definition for singly-linked list */
```

```
struct ListNode {
```

```
    int val;
```

```
    struct ListNode *next;
```

```
};
```

```
/* Function to create a new node */
```

```
struct ListNode* createNode(int val) {
```

```
    struct ListNode* newNode = (struct ListNode*)malloc(sizeof(struct ListNode));
```

```
    newNode->val = val;
```

```
    newNode->next = NULL;
```

```
    return newNode;
```

```
}
```

```
/* Function to display the linked list */
```

```
void displayList(struct ListNode* head) {
```

```
    struct ListNode* temp = head;
```

```
    while (temp != NULL) {
```

```
        printf("%d -> ", temp->val);
```

```
        temp = temp->next;
```

```
    }
```

```
    printf("NULL\n");
```

```
}
```

```
/* Function to remove all elements with a specific value */
```

```
struct ListNode* removeElements(struct ListNode* head, int val) {
```

```
    /* Remove nodes from the beginning if they match val */
```

```

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

struct ListNode* current = head;

/* Traverse the list and remove matching nodes */
while (current != NULL && current->next != NULL) {
    if (current->next->val == val) {
        struct ListNode* temp = current->next;
        current->next = current->next->next;
        free(temp);
    } else {
        current = current->next;
    }
}

return head;
}

/* Function to create a linked list from user input */
struct ListNode* createList(int n) {
    int val;
    struct ListNode *head = NULL, *tail = NULL;

    for (int i = 0; i < n; i++) {
        printf("Enter element %d: ", i + 1);
        scanf("%d", &val);
        struct ListNode* newNode = createNode(val);
    }
}

```

```

        if (head == NULL) {
            head = tail = newNode;
        } else {
            tail->next = newNode;
            tail = newNode;
        }
    }
}

return head;
}

int main() {
    int n, val;

    printf("Enter number of elements in the list: ");
    scanf("%d", &n);

    printf("Enter elements of the list:\n");
    struct ListNode* head = createList(n);

    printf("Original list:\n");
    displayList(head);

    printf("Enter value to remove: ");
    scanf("%d", &val);

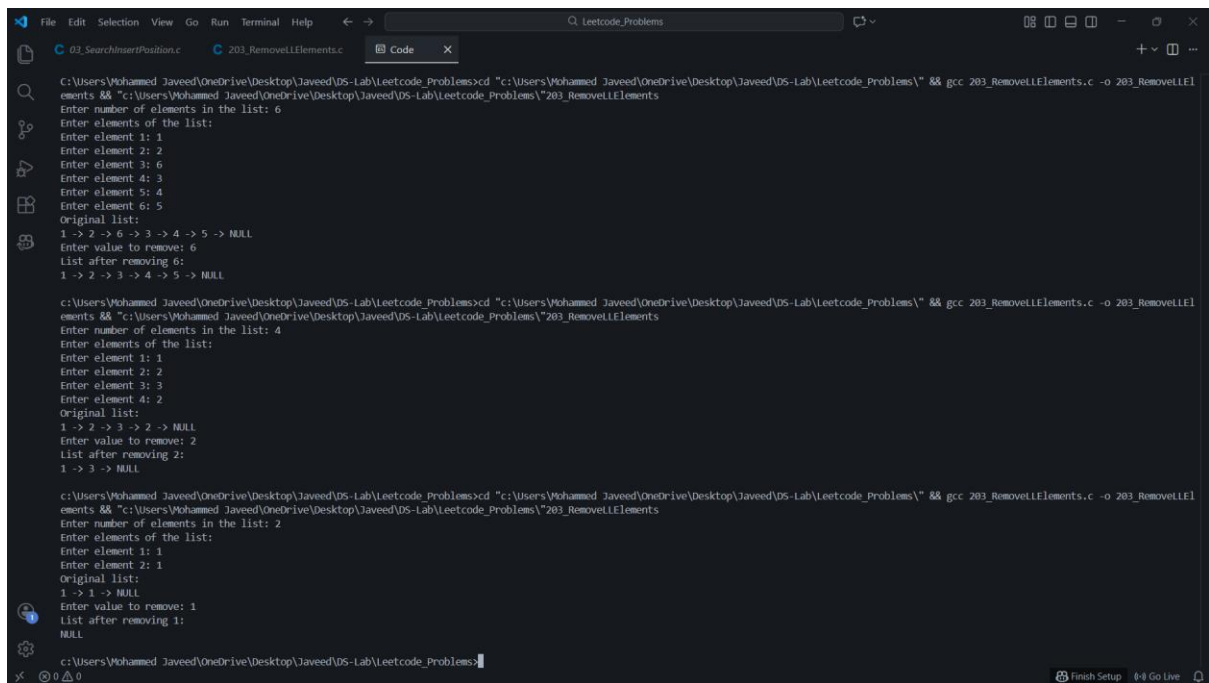
    head = removeElements(head, val);

    printf("List after removing %d:\n", val);
    displayList(head);

```

```
return 0;
}
```

OUTPUT:



```
c:\Users\Mohammed Javeed\OneDrive\Desktop\Javeed\DS-Lab\Leetcode_Problems>cd "c:\Users\Mohammed Javeed\OneDrive\Desktop\Javeed\DS-Lab\Leetcode_Problems\" && gcc 203_RemoveElements.c -o 203_RemoveElements.exe
Enter number of elements in the list: 6
Enter elements of the list:
Enter element 1: 1
Enter element 2: 2
Enter element 3: 6
Enter element 4: 3
Enter element 5: 4
Enter element 6: 5
Original list:
1 -> 2 -> 6 -> 3 -> 4 -> 5 -> NULL
Enter value to remove: 6
List after removing 6:
1 -> 2 -> 3 -> 4 -> 5 -> NULL

c:\Users\Mohammed Javeed\OneDrive\Desktop\Javeed\DS-Lab\Leetcode_Problems>cd "c:\Users\Mohammed Javeed\OneDrive\Desktop\Javeed\DS-Lab\Leetcode_Problems\" && gcc 203_RemoveElements.c -o 203_RemoveElements.exe
Enter number of elements in the list: 4
Enter elements of the list:
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Enter element 4: 2
Original list:
1 -> 2 -> 3 -> 2 -> NULL
Enter value to remove: 2
List after removing 2:
1 -> 3 -> NULL

c:\Users\Mohammed Javeed\OneDrive\Desktop\Javeed\DS-Lab\Leetcode_Problems>cd "c:\Users\Mohammed Javeed\OneDrive\Desktop\Javeed\DS-Lab\Leetcode_Problems\" && gcc 203_RemoveElements.c -o 203_RemoveElements.exe
Enter number of elements in the list: 2
Enter elements of the list:
Enter element 1: 1
Enter element 2: 1
Original list:
1 -> 1 -> NULL
Enter value to remove: 1
List after removing 1:
NULL

c:\Users\Mohammed Javeed\OneDrive\Desktop\Javeed\DS-Lab\Leetcode_Problems>
```