203. Remove Linked List Elements:

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*.

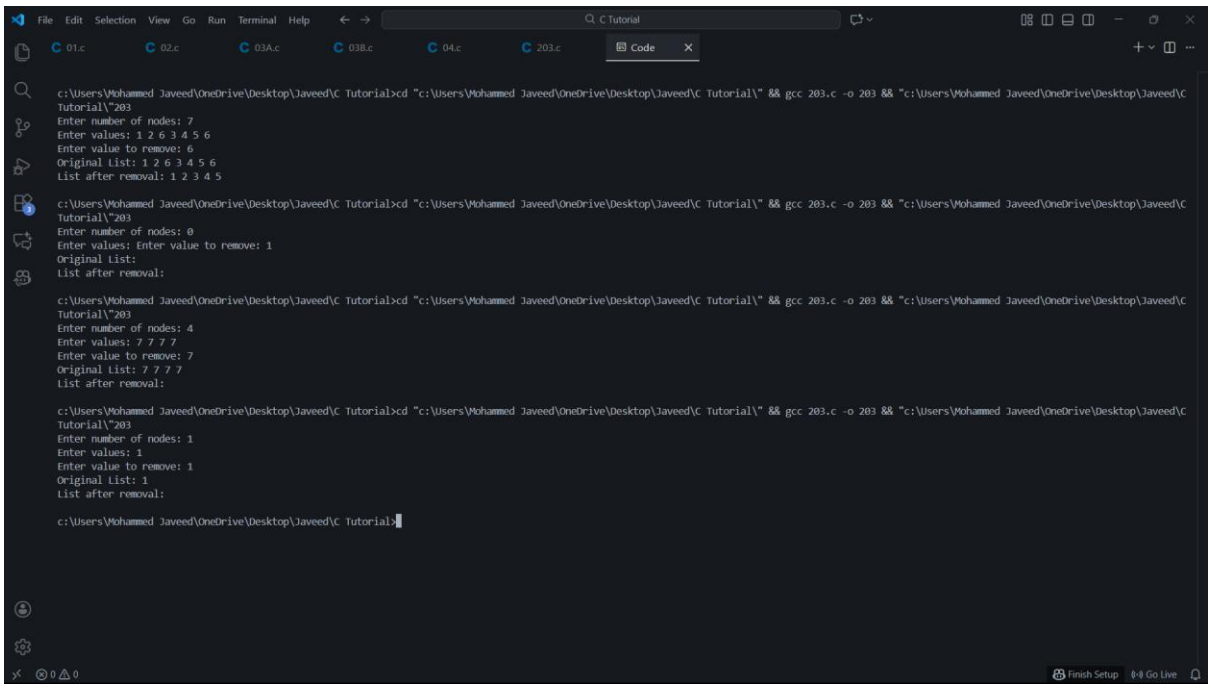
Solution :

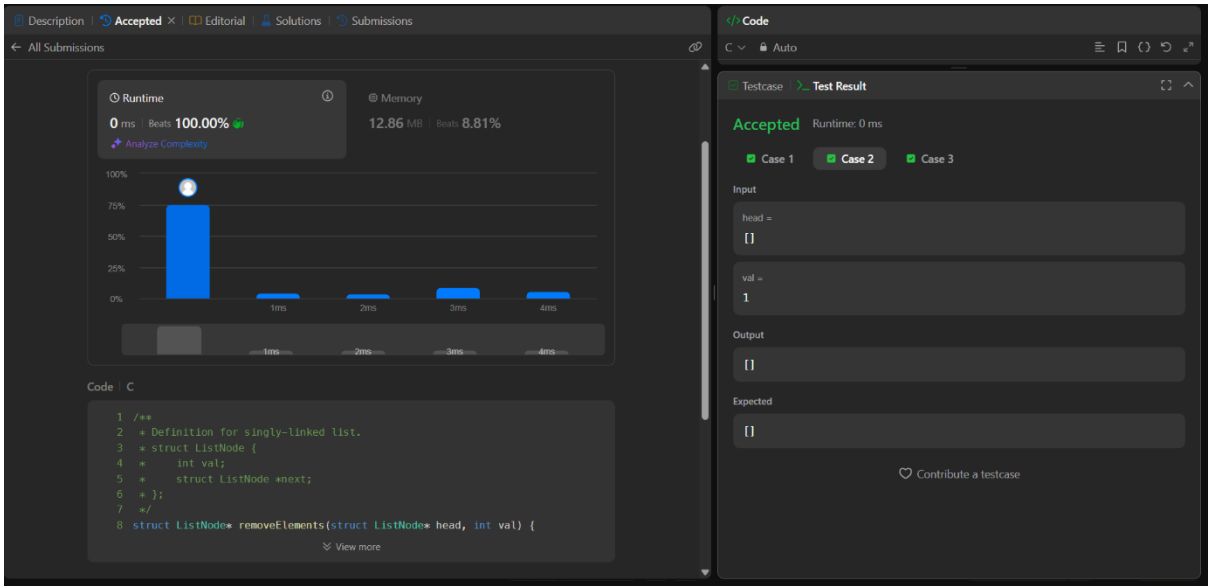
```
#include <stdio.h>
#include <stdlib.h>
// Definition for singly-linked list.
struct ListNode
{
   int val;
   struct ListNode *next;
};
struct ListNode* removeElements(struct ListNode* head, int val)
{
   while (head != NULL && head->val == val) {
      struct ListNode* temp = head;
      head = head->next;
      free(temp);
   }
   struct ListNode* current = head;
   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;
      }
   }
```

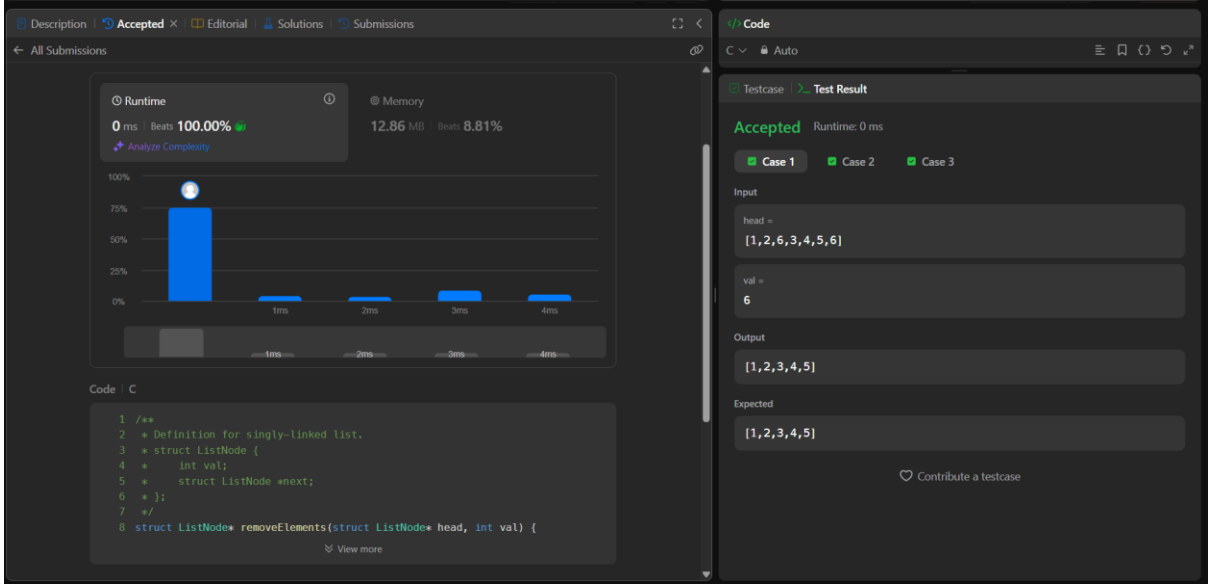
```
    return head;
}


// Function to insert node at end
struct ListNode* insertAtEnd(struct ListNode* head, int val)
{
    struct ListNode* newNode = (struct ListNode*)malloc(sizeof(struct ListNode));
    newNode->val = val;
    newNode->next = NULL;
    if (head == NULL)
        return newNode;
    struct ListNode* temp = head;
    while (temp->next != NULL)
        temp = temp->next;
    temp->next = newNode;
    return head;
}

// Print list
void printList(struct ListNode* head)
{
    while (head != NULL)
 {
        printf("%d ", head->val);
        head = head->next;
    }
    printf("\n");
}

int main()
{
```

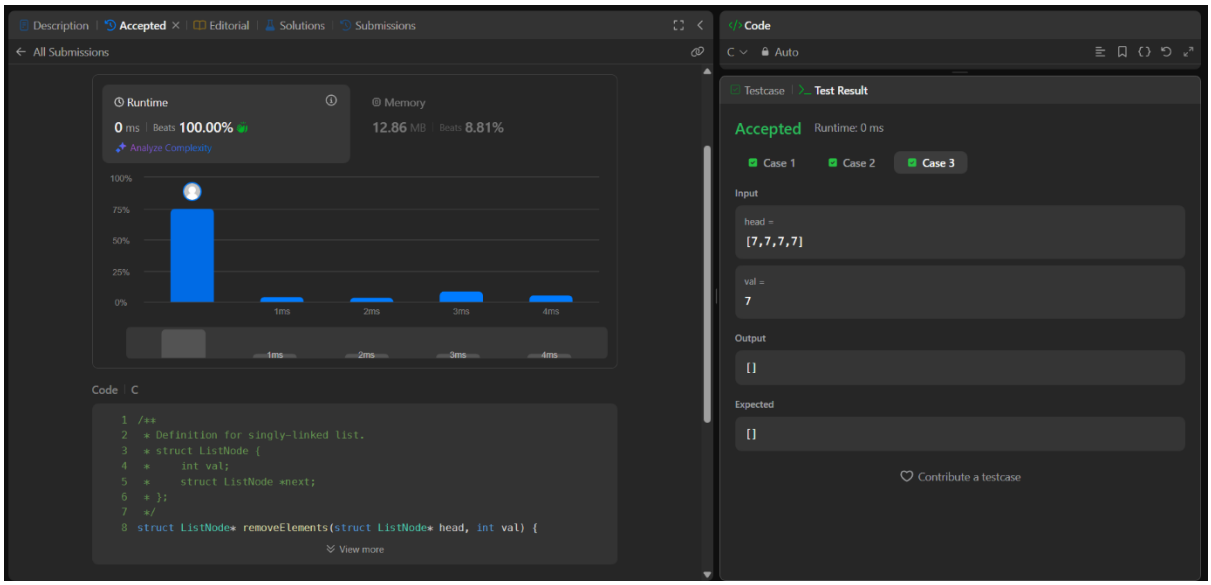
```
    struct ListNode* head = NULL;
    int n, val, x;
    printf("Enter number of nodes: ");
    scanf("%d", &n);
    printf("Enter values: ");
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &x);
        head = insertAtEnd(head, x);
    }
    printf("Enter value to remove: ");
    scanf("%d", &val);
    printf("Original List: ");
    printList(head);
    head = removeElements(head, val);
    printf("List after removal: ");
    printList(head);
    return 0;
}
```

OUTPUT:




Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
12.86 MB | Beats 8.81%
100%
75%
50%
25%
0%
1ms
2ms
3ms
4ms
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) {
View more
Code
C
Auto
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
head =
[]
val =
1
Output
[]
Expected
[]
Contribute a testcase


Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
12.86 MB | Beats 8.81%
100%
75%
50%
25%
0%
1ms
2ms
3ms
4ms
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) {
View more
Code
C
Auto
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]
Contribute a testcase


Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
12.86 MB | Beats 8.81%
100%
75%
50%
25%
0%
1ms
2ms
3ms
4ms
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) {
View more
Code
C
Auto
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
head =
[7,7,7,7]
val =
7
Output
[]
Expected
[]
Contribute a testcase