



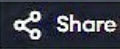
LIVE YOUR
STORIES
CB350
RS



F

Clear

main.c



Run

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 struct Node {
4     int data;
5     struct Node* next;
6 };
7 void deleteNode(struct Node** head, int pos) {
8     if (*head == NULL) {
9         printf("List is empty.\n");
10        return;
11    }
12    struct Node* temp = *head;
13    if (pos == 1) {
14        *head = temp->next;
15        free(temp);
16        return;
17    }
18    for (int i = 1; temp != NULL && i < pos - 1; i++) {
19        temp = temp->next;
20    }
21    if (temp == NULL || temp->next == NULL) {
22        printf("Position not found.\n");
23        return;
24    }
25    struct Node* nextNode = temp->next->next;
26    free(temp->next);
27    temp->next = nextNode;
```

Original List:
10 -> 20 -> 30 -> NULL
After Deletion at position 2:
10 -> 30 -> NULL

=== Code Execution Successful ===

main.c

```
25 struct Node* nextNode = temp->next->next;
26 free(temp->next);
27 temp->next = nextNode;
28 }
29 void printList(struct Node* node) {
30     while (node != NULL) {
31         printf("%d -> ", node->data);
32         node = node->next;
33     }
34     printf("NULL\n");
35 }
36 int main() {
37     struct Node* head = (struct Node*)malloc(sizeof(struct Node));
38     struct Node* second = (struct Node*)malloc(sizeof(struct Node));
39     struct Node* third = (struct Node*)malloc(sizeof(struct Node));
40     head->data = 10;
41     head->next = second;
42     second->data = 20;
43     second->next = third;
44     third->data = 30;
45     third->next = NULL;
46     printf("Original List:\n");
47     printList(head);
48     deleteNode(&head, 2);
49     printf("After Deletion at position 2:\n");
50     printList(head);
51     return 0;
52 }
```

Output

Clear

```
Original List:
10 -> 20 -> 30 -> NULL
After Deletion at position 2:
10 -> 30 -> NULL
```

```
=== Code Execution Successful ===
```