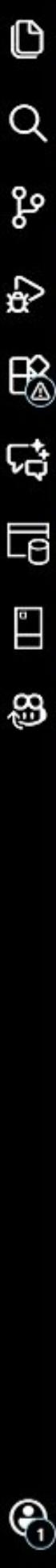


C deletelinked.c > main()

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct Node {
5      int data;
6      struct Node *next;
7  };
8  struct Node* createNode(int data) {
9      struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
10     newNode->data = data;
11     newNode->next = NULL;
12     return newNode;
13 }
14 struct Node* createList() {
15     int n, data;
16     struct Node *head = NULL, *temp = NULL, *newNode = NULL;
17
18     printf("Enter number of nodes: ");
19     scanf("%d", &n);
20
21     for(int i = 0; i < n; i++) {
22         printf("Enter data for node %d: ", i + 1);
23         scanf("%d", &data);
24
25         newNode = createNode(data);
26
27         if(head == NULL) {
28             head = newNode;
29             temp = head;
30         } else {
31             temp->next = newNode;
32             temp = newNode;
33         }
34     }
35     return head;
36 }
37 struct Node* deleteFirst(struct Node* head) {
38     if(head == NULL) {
39         printf("List is empty!\n");
40         return head;
41     }
42     struct Node* temp = head;
43     head = head->next;
44     free(temp);
45     printf("First element deleted.\n");
46     return head;
47 }
48 struct Node* deleteLast(struct Node* head) {
```



C deletelinked.c > main()

```
48     struct Node* deleteLast(struct Node* head) {
49         if(head == NULL) {
50             printf("List is empty!\n");
51             return head;
52         }
53         if(head->next == NULL) {
54             free(head);
55             printf("Last element deleted.\n");
56             return NULL;
57         }
58
59         struct Node *temp = head;
60         while(temp->next->next != NULL) {
61             temp = temp->next;
62         }
63
64         free(temp->next);
65         temp->next = NULL;
66         printf("Last element deleted.\n");
67         return head;
68     }
69     struct Node* deleteSpecific(struct Node* head, int key) {
70         if(head == NULL) {
71             printf("List is empty!\n");
72             return head;
73         }
74
75         if(head->data == key) {
76             return deleteFirst(head);
77         }
78
79         struct Node *temp = head, *prev = NULL;
80
81         while(temp != NULL && temp->data != key) {
82             prev = temp;
83             temp = temp->next;
84         }
85
86         if(temp == NULL) {
87             printf("Element %d not found!\n", key);
88             return head;
89         }
90
91         prev->next = temp->next;
92         free(temp);
93         printf("Element %d deleted.\n", key);
94
95         return head;
```



✖ ⊗ 0 △ 4

C deletelinked.c > main()

```
97 void display(struct Node* head) {
98     if(head == NULL) {
99         printf("List is empty!\n");
100        return;
101    }
102
103    struct Node* temp = head;
104    printf("Linked List: ");
105    while(temp != NULL) {
106        printf("%d -> ", temp->data);
107        temp = temp->next;
108    }
109    printf("NULL\n");
110}
111int main() {
112    struct Node* head = NULL;
113    int choice, key;
114
115    while(1) {
116        printf("\n--- Singly Linked List Menu ---\n");
117        printf("1.Create List\n2.Delete First Element\n3.Delete Last Element\n");
118        printf("4.Delete Specific Element\n5. Display List\n6. Exit\n");
119        printf("Enter your choice: ");
120        scanf("%d", &choice);
121
122        switch(choice) {
123            case 1:
124                head = createList();
125                break;
126            case 2:
127                head = deleteFirst(head);
128                break;
129            case 3:
130                head = deleteLast(head);
131                break;
132            case 4:
133                printf("Enter element to delete: ");
134                scanf("%d", &key);
135                head = deleteSpecific(head, key);
136                break;
137            case 5:
138                display(head);
139                break;
140            case 6:
141                exit(0);
142            default:
143                printf("Invalid choice!\n"); } }
144
return 0;
```



...	PROBLEMS	4	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS	QUERY RESULTS
4					PS C:\Users\gsm22\OneDrive\Documents\DS> cd "c:\Users\gsm22\OneDrive\Documents\DS\" ; if (\$?)		

```
--- Singly Linked List Menu ---
1. Create List
2. Delete First Element
3. Delete Last Element
4. Delete Specific Element
5. Display List
6. Exit
Enter your choice: 1
Enter number of nodes: 5
Enter data for node 1: 25
Enter data for node 2: 43
Enter data for node 3: 62
Enter data for node 4: 85
Enter data for node 5: 96

--- Singly Linked List Menu ---
1. Create List
2. Delete First Element
3. Delete Last Element
4. Delete Specific Element
5. Display List
6. Exit
Enter your choice: 4
Enter element to delete: 62
Element 62 deleted.

--- Singly Linked List Menu ---
1. Create List
2. Delete First Element
3. Delete Last Element
4. Delete Specific Element
5. Display List
6. Exit
Enter your choice: 5
Linked List: 25 -> 43 -> 85 -> 96 -> NULL

--- Singly Linked List Menu ---
1. Create List
2. Delete First Element
3. Delete Last Element
4. Delete Specific Element
5. Display List
6. Exit
Enter your choice: 2
First element deleted.

--- Singly Linked List Menu ---
1. Create List
2. Delete First Element
3. Delete Last Element
4. Delete Specific Element
5. Display List
6. Exit
Enter your choice: 3
Last element deleted.

--- Singly Linked List Menu ---
1. Create List
2. Delete First Element
3. Delete Last Element
4. Delete Specific Element
5. Display List
6. Exit
Enter your choice: 5
Linked List: 43 -> 85 -> NULL

--- Singly Linked List Menu ---
1. Create List
2. Delete First Element
3. Delete Last Element
4. Delete Specific Element
5. Display List
6. Exit
Enter your choice: 6
PS C:\Users\gsm22\OneDrive\Documents\DS> []
```