

LAB-5 (Singly Linked List-DELETE)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct node {
5      int data;
6      struct node *next;
7  };
8
9  struct node *head = NULL;
10
11 void create() {
12     int n, i, val;
13     struct node *temp, *newnode;
14     printf("Enter number of nodes: ");
15     scanf("%d", &n);
16     for (i = 0; i < n; i++) {
17         newnode = (struct node *)malloc(sizeof(struct node));
18         printf("Enter data: ");
19         scanf("%d", &val);
20         newnode->data = val;
21         newnode->next = NULL;
22         if (head == NULL) {
23             head = newnode;
24             temp = head;
25         } else {
26             temp->next = newnode;
27             temp = newnode;
28         }
29     }
30 }
31
32 void delete_first() {
33     struct node *temp;
34     if (head == NULL) {
35         printf("List is empty\n");
36         return;
37     }
38     temp = head;
39     head = head->next;
40     free(temp);
41 }
42
43 void delete_last() {
44     struct node *temp, *prev;
45     if (head == NULL) {
46         printf("List is empty\n");
47         return;
48     }
49     if (head->next == NULL) {
50         free(head);
51         head = NULL;
52         return;
53     }
54     temp = head;
55     while (temp->next != NULL) {
56         prev = temp;
57         temp = temp->next;
58     }
59     prev->next = NULL;
60     free(temp);
61 }
62
63 void delete_element() {
64     int key;
65     struct node *temp, *prev;
66     if (head == NULL) {
67         printf("List is empty\n");
68         return;
69     }
70     printf("Enter element to delete: ");
71     scanf("%d", &key);
72     if (head->data == key) {
73         delete_first();
74         return;
75     }
76     temp = head;
77     while (temp != NULL && temp->data != key) {
78         prev = temp;
79         temp = temp->next;
80     }
81     if (temp == NULL) {
82         printf("Element not found\n");
83         return;
84     }
85     prev->next = temp->next;
86     free(temp);
87 }
```

OUTPUT:

```
1.Create
2.Delete First
3.Delete Specific
4.Delete Last
5.Display
6.Exit
Enter choice: 1
Enter number of nodes: 3
Enter data: 4
Enter data: 5
Enter data: 6

1.Create
2.Delete First
3.Delete Specific
4.Delete Last
5.Display
6.Exit
Enter choice: 3
Enter element to delete: 5

1.Create
2.Delete First
3.Delete Specific
4.Delete Last
5.Display
6.Exit
Enter choice: 4

1.Create
2.Delete First
3.Delete Specific
4.Delete Last
5.Display
6.Exit
Enter choice: 5
4

1.Create
2.Delete First
3.Delete Specific
4.Delete Last
5.Display
6.Exit
Enter choice: 6

Process returned 0 (0x0)   execution time : 23.815 s
Press any key to continue.
```