

Start here X Lab1stack.c X Lab3aLinearqueue.c X Lab3bCircularqueue.c X Lab4SLL.c X Lab5SLL.c X \*Lab7DLL.c X

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 struct node{
5     int data;
6     struct node *next;
7 };
8 struct node *head = NULL;
9
10 void createlist(int n){
11     struct node *newnode, *temp;
12     int data, i;
13     head = (struct node*)malloc(sizeof(struct node));
14     if(head == NULL){
15         [node* head]emory not allocated\n";
16         return;
17     }
18     printf("Enter data for node 1: ");
19     scanf("%d", &data);
20     head->data = data;
21     head->next = NULL;
22     temp = head;
23     for(i = 2; i <= n; i++){
24         newnode = (struct node*)malloc(sizeof(struct node));
25         if(newnode == NULL){
26             printf("Memory not allocated!\n");
27             break;
28         }
29         printf("Enter data for node %d: ", i);
30         scanf("%d", &data);
31
32         newnode->data = data;
33         newnode->next = NULL;
34
35         temp->next = newnode;
36         temp = newnode;
37     }
38     printf("Linked list created successfully\n");
39 }
40
41 void deleteFirst(){
42     if(head == NULL){
43         printf("List is empty.\n");
```

```
Start here X Lab1stack.c X Lab3aLinearqueue.c X Lab3bCircularqueue.c X Lab4SLL.c X Lab5SLL.c X *Lab7DLL.c X
43     printf("List is empty.\n");
44     return;
45 }
46 struct node *temp = head;
47 head = head->next;
48 printf("Deleted %d\n", temp->data);
49 free(temp);
50 }
51
52 void deleteLast(){
53     if(head == NULL){
54         printf("List is empty.\n");
55         return;
56     }
57     struct node *temp = head, *prev = NULL;
58     if(head->next == NULL){
59         printf("Deleted: %d\n", head->data);
60         free(head);
61         head = NULL;
62         return;
63     }
64     while(temp->next != NULL){
65         prev = temp;
66         temp = temp->next;
67     }
68     printf("Deleted %d\n", temp->data);
69     prev->next = NULL;
70     free(temp);
71 }
72
73 void specific(int value){
74     if(head == NULL){
75         printf("List is empty.\n");
76         return;
77     }
78     struct node *temp = head, *prev = NULL;
79     if(head->data == value){
80         head = head->next;
81         printf("Deleted element: %d\n", temp->data);
82         free(temp);
83         return;
84     }
85     while(temp != NULL && temp->data != value){
```

```
Start here X Lab1stack.c X Lab3alinearqueue.c X Lab3bCircularqueue.c X Lab4SLL.c X Lab5SLL.c X *Lab7DLL.c X
85     while(temp != NULL && temp->data != value){
86         prev = temp;
87         temp = temp->next;
88     }
89     if(temp == NULL){
90         printf("Element %d not found in list.\n", value);
91         return;
92     }
93     prev->next = temp->next;
94     printf("Deleted %d\n", temp->data);
95     free(temp);
96 }
97
98 void display(){
99     struct node *temp = head;
100    if(temp == NULL){
101        printf("List is empty.\n");
102        return;
103    }
104    printf("Linked List: ");
105    while(temp != NULL){
106        printf("%d->", temp->data);
107        temp = temp->next;
108    }
109    printf("NULL\n");
110 }
111
112 int main(){
113     int choice, n, value;
114     while(1){
115         printf("\n-- Menu --\n");
116         printf("1. Create Linked list\n");
117         printf("2. Delete 1st element\n");
118         printf("3. Delete specific element\n");
119         printf("4. Delete last element\n");
120         printf("5. Display List\n");
121         printf("6. Exit\n");
122         printf("Enter your choice: ");
123         scanf("%d", &choice);
124         switch(choice){
125             case 1:printf("Enter number of nodes: ");
126                 scanf("%d", &n);
127                 createlist(n);
```

```
Start here X Lab1stack.c X Lab3aLinearqueue.c X Lab3bCircularqueue.c X Lab4SLL.c X Lab5SLL.c X *Lab7DLL.c X
105     while (temp != NULL) {
106         printf("%d\n");
107         temp = temp->next;
108     }
109     printf("NULL\n");
110 }
111
112 int main() {
113     int choice, n, value;
114     while(1) {
115         printf("\n-- Menu --\n");
116         printf("1. Create Linked list\n");
117         printf("2. Delete 1st element\n");
118         printf("3. Delete specific element\n");
119         printf("4. Delete last element\n");
120         printf("5. Display List\n");
121         printf("6. Exit\n");
122         printf("Enter your choice: ");
123         scanf("%d", &choice);
124         switch(choice) {
125             case 1:printf("Enter number of nodes: ");
126             scanf("%d", &n);
127             createlist(n);
128             break;
129             case 2:deleteFirst();
130             break;
131             case 3:printf("Enter element to delete: ");
132             scanf("%d", &value);
133             specific(value);
134             break;
135             case 4:deleteLast();
136             break;
137             case 5:display();
138             break;
139             case 6:printf("Exiting...\n");
140             exit(0);
141             default:printf("Invalid choice! Try again.\n");
142         }
143     }
144     return 0;
145 }
146 }
```

```
D:\Coding\C\LAB\Lab5SLL.exe + - x
```

-- Menu --  
1. Create Linked list  
2. Delete 1st element  
3. Delete specific element  
4. Delete last element  
5. Display List  
6. Exit  
Enter your choice: 1  
Enter number of nodes: 3  
Enter data for node 1: 5  
Enter data for node 2: 6  
Enter data for node 3: 7  
Linked list created successfully

-- Menu --  
1. Create Linked list  
2. Delete 1st element  
3. Delete specific element  
4. Delete last element  
5. Display List  
6. Exit  
Enter your choice: 5  
Linked List: 5->6->7->NULL

-- Menu --  
1. Create Linked list  
2. Delete 1st element  
3. Delete specific element  
4. Delete last element  
5. Display List  
6. Exit  
Enter your choice: 2  
Deleted 5

-- Menu --  
1. Create Linked list  
2. Delete 1st element  
3. Delete specific element  
4. Delete last element  
5. Display List  
6. Exit  
Enter your choice: 5  
Linked List: 6->7->NULL

-- Menu --  
1. Create Linked list