

```

1 //Linked List 6
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 struct node{
6     int data;
7     struct node *next;
8 };
9
10 void insertlast(struct node **headptr){
11     struct node *newnode,*temp;
12     int value;
13     printf("Enter value: ");
14     scanf("%d",&value);
15     newnode = (struct node*)malloc(sizeof(struct node));
16     newnode->data = value;
17     newnode->next = NULL;
18     temp = (*headptr);
19     if(*headptr == NULL)
20         (*headptr) = newnode;
21     else{
22         while(temp->next != NULL)
23             temp = temp->next;
24         temp->next = newnode;
25     }
26 }
27 void deletefirst(struct node **headptr){
28     if((*headptr) == NULL)
29         printf("The list is empty\n");
30     else if((*headptr)->next == NULL)
31         (*headptr) = NULL;
32     else{
33         (*headptr) = (*headptr)->next;
34     }
35 }
36 void deletepos(struct node **headptr){
37     struct node *temp;
38     int count=0,currpos=1,pos;
39     if((*headptr) == NULL){
40         printf("The list is empty\n");
41         return;
42     }
43     else{
44         if((*headptr)->next == NULL){

```

```

45     (*headptr) = NULL;
46     printf("The only element in the list was deleted\n");
47     return;
48 }
49 temp = (*headptr);
50 while(temp != NULL){
51     count++;
52     temp = temp->next;
53 }
54 printf("There are %d elements in the list.Enter pos of element to be deleted: ",count);
55 scanf("%d",&pos);
56 if(pos > (count+1)){
57     printf("No such position is present\n");
58     return;
59 }
60 if(pos == (count+1)){
61     temp = (*headptr);
62     while((temp->next)->next != NULL)
63         temp = temp->next;
64     temp->next = NULL;
65 }
66 else{
67     temp = (*headptr);
68     while(temp->next != NULL){
69         if(pos == 1){
70             (*headptr) = (*headptr)->next;
71             return;
72         }
73         if(currpos == pos-1){
74             temp->next = (temp->next)->next;
75             return;
76         }
77         currpos++;
78         temp = temp->next;
79     }
80     printf("No such element was found!!!\n");
81 }
82 }
83 }
84 void deletelast(struct node **headptr){
85     struct node *temp;
86     temp = (*headptr);
87     if((*headptr) == NULL)
88         printf("The list is empty\n");

```

```

89     else if((*headptr)->next == NULL)
90     {
91         (*headptr) = NULL;
92     }
93     else{
94         temp = *headptr;
95         while((temp->next)->next != NULL)
96             temp = temp->next;
97         temp->next = NULL;
98     }
99 }
100 void display(struct node *temp){
101     if(temp == NULL){
102         printf("The list is empty\n");
103         return;
104     }
105     else{
106         while(temp != NULL){
107             printf("%d\t",temp->data);
108             temp = temp->next;
109         }
110         printf("\n");
111     }
112 }
113 int main(int argc,char **argv){
114     int choice;
115     struct node *head = NULL;
116     while(choice != 6){
117         printf("Enter choice 1)insertlast 2)deletefirst 3)deletapos 4)deletelast 5)display 6)exit : ");
118         scanf("%d",&choice);
119         switch(choice){
120             case 1:insertlast(&head);break;
121             case 2:deletefirst(&head);break;
122             case 3:deletapos(&head);break;
123             case 4:deletelast(&head);break;
124             case 5:display(head);break;
125             case 6:|
126                 default:exit(0);
127         }
128     }
129     return 0;
130 }

```