

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct node
5  {
6      int data;
7      struct node *next;
8  };
9
10 struct node *head= NULL;
11
12 void create()
13 {
14     int ele;
15     struct node *newnode,*temp;
16     newnode =(struct node *) malloc (sizeof(struct node));
17     printf("Enter data to be inserted: ");
18     scanf("%d",&ele);
19     newnode -> data = ele;
20     if(head == NULL)
21     {
22         newnode -> next = NULL;
23         head = newnode;
24     }
25     else
26     {
27         temp = head;
28         while(temp -> next != NULL)
29         {
30             temp = temp ->next;
31         }
32         temp -> next = newnode;
33         newnode -> next = NULL;
34     }
35 }
36 void delfunct()
37 {
38     int pos,count=1,trav=1;
39     struct node *temp;
40     temp = head;
41     if(head == NULL)
42     {
43         printf("There are no elements in the list!\n");

```

```

44         return;
45     }
46
47     {
48         while(temp -> next != NULL)
49         {
50             temp = temp -> next;
51             count++;
52         }
53         printf("There are %d elements in the list!\n",count);
54     }
55     printf("Which element do you want to delete?: ");
56     scanf("%d",&pos);
57     temp = head;
58     if(pos == count)
59     {
60         if(head->next == NULL)
61             head = NULL;
62         else
63         {
64             while((temp->next)->next != NULL)
65             {
66                 temp = temp -> next;
67             }
68             temp -> next = NULL;
69         }
70         return;
71     }
72     temp = head;
73     while(temp->next != NULL)
74     {
75         if(trav == pos-1)
76         {
77             temp->next = (temp->next)->next;
78             return;
79         }
80         temp = temp->next;
81         trav++;
82     }
83     printf("Position not found!\n");
84 }
85 void display()
86 {

```

```

87     struct node *temp = NULL;
88     temp = head;
89     if(temp == NULL)
90         printf("No elements in list!\n");
91     else
92         while(temp != NULL)
93         {
94             printf("%d\t",temp -> data);
95             temp = temp -> next;
96         }
97     printf("\n");
98 }
99
100 void delfirst()
101 {
102     if(head == NULL)
103     {
104         printf("The list is empty!\n");
105         return;
106     }
107     else if(head->next == NULL)
108         head = NULL;
109     else
110         head = head->next;
111 }
112
113 void dellast()
114 {
115     struct node *temp;
116     temp = head;
117     if(head == NULL)
118     {
119         printf("The list is empty!\n");
120         return;
121     }
122     else if(head->next == NULL)
123         head = NULL;
124     else
125     {
126         while((temp->next)->next != NULL)
127         {
128             temp = temp -> next;
129         }

```

```

130     }
131     temp -> next = NULL;
132 }
133
134 int main(int argc,char** argv)
135 {
136     int choice;
137     while(choice != 6)
138     {
139         printf("---Linked List---\nEnter choice:\n1.Insert to end\n2.delete at the beginning\n3.delete at the end\n4.Delete a position\n5.Display\n6.exit\n");
140         scanf("%d",&choice);
141         switch(choice)
142         {
143             case 1:create();
144                 break;
145             case 2:delfirst();
146                 break;
147             case 3:dellast();
148                 break;
149             case 4:delfunct();
150                 break;
151             case 5:display();
152                 break;
153             case 6:exit(0);
154                 break;
155             default:exit(0);
156         }
157     }
158     return 0;
159 }

```