

```

1  #include<stdio.h>
2
3  #include<stdlib.h>
4  struct node
5  {
6      int data;
7      struct node *next;
8      struct node *prev;
9  };
10 struct node *head=NULL;
11 void insert_beg()
12 {
13     struct node *new_node;
14     new_node=(struct node*)malloc(sizeof(struct node));
15     printf("Enter the item\n");
16     scanf("%d",&new_node->data);
17     new_node->next=NULL;
18     new_node->prev=NULL;
19
20     if(head==NULL)
21     {
22         head=new_node;
23     }
24     else
25     {
26         new_node->next=head;
27         head->prev=new_node;
28         head=new_node;
29     }
30
31 }
32
33 void insert_end()
34 {

```



```

34 * {
35     struct node *new_node, *temp;
36     new_node=(struct node*)malloc(sizeof(struct node));
37     printf("Enter the item\n");
38     scanf("%d",&new_node->data);
39     new_node->next=NULL;
40     new_node->prev=NULL;
41     if(head==NULL)
42     {
43         head=new_node;
44     }
45     else
46     {
47         temp=head;
48         while(temp->next!=NULL)
49             temp=temp->next;
50         temp->next=new_node;
51         new_node->prev=temp;
52     }
53 }
54
55 }
56 void insert_between()
57 {
58     int listele;
59     struct node *new_node, *temp;
60     printf("Enter the element in the list\n");
61     scanf("%d",&listele);
62     new_node=(struct node*)malloc(sizeof(struct node));
63     printf("Enter the new node data\n");
64     scanf("%d",&new_node->data);
65     new_node->next=NULL;
66     new_node->prev=NULL;
67     if(head==NULL)

```



```

56 new_node->prev=NULL;
57 if(head==NULL)
58 {
59     printf("Empty list\n"); return;
60 }
61 temp=head;
62 while(temp->data!=listele)
63 {
64     temp=temp->next;
65     if(temp==NULL)
66     {
67         printf("Element is not in the list");
68         return;
69     }
70 }
71 new_node->next=temp->next;
72 temp->next=new_node;
73 new_node->prev=temp;
74 new_node->next->prev=new_node;
75 }
76 void del()
77 {
78     struct node *temp;
79     int ele;
80     if(head==NULL)
81     {
82         printf("Empty List \n");
83         return;
84     }
85     printf("Enter the element to be deleted\n");
86     scanf("%d",&ele);
87     temp=head;
88     while(temp->data!=ele)
89     {
90         temp=temp->next;
91     }

```

```

100     temp=temp->next;
101     if(temp==NULL)
102     {
103         printf("Element is not in the list\n");
104         break;
105     }
106 }
107 if(temp==head)
108 {
109     head=head->next;
110 }
111 else if(temp->next==NULL)
112 {
113     temp=temp->prev;
114     temp->next=NULL;
115 }
116
117 else
118 {
119     temp->prev->next=temp->next;
120     temp->next->prev=temp->prev;
121 }
122 }
123 void display()
124 {
125     struct node *temp;
126     temp=head;
127     while(temp!=NULL)
128     {
129         printf("%d\t",temp->data);
130         temp=temp->next;
131     }
132     printf("\n");

```



```

while(temp != NULL)
{
    printf("%d\t", temp->data);
    temp=temp->next;
}
printf("\n");
}

int main()
{
    int choice;

    while(1)
    {
        printf(" 1. Insert at the beg \n");
        printf(" 2. Insert at the end \n");
        printf(" 3. Insert after a given node\n");
        printf(" 4. Delete \n");
        printf(" 5. Display\n");
        printf(" 6. Exit\n");
        printf("Enter your choice\n");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: insert_beg(); break;
            case 2: insert_end();break;
            case 3: insert_between();break;
            case 4: del(); break;
            case 5: display(); break;
            case 6: exit(0);
        }
    }
}

```