

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  struct node
4  {
5      int data;
6      struct node *next;
7  };
8  struct node *start=0;
9  struct node *createNode()
10 {
11     struct node *cn;
12     cn = (struct node*)malloc(sizeof(struct node));
13     printf("\nEnter data:");
14     scanf("%d",&cn->data);
15     cn->next = 0;
16     return cn;
17 }
18 void insertStart()
19 {
20     struct node *is;
21     is = createNode();
22     if(start == 0)
23     {
24         start = is;
25     }
26     else
27     {
28         is->next = start;
29         start=is;
30     }
31 }
32 void insertMiddle()
33 {
34     int data;
35     struct node *t2, *k3;
36     t2=createNode();
37     printf("Enter the data after which you want to insert: ");
38     scanf("%d", &data);
39     if(start==0)
40     {
41         start=t2;
42     }
43     else
44     {
45         k3=start;
46         while(k3->data!=data)
47         {
48             k3=k3->next;
49         }
50         t2->next=k3->next;
51         k3->next=t2;
52     }
53 }
54 }
55 void insertEnd()
56 {
57     struct node *ie,*search;
58     ie=createNode();
59     if(start == 0)
60     {
61         start = ie;
62     }
63     else
64     {
65         search=start;
66         while(search->next != 0)
67         {
68             search=search->next;
69         }

```

```

70         search->next = ie;
71     }
72 }
73 void deleteStart()
74 {
75     struct node *ds;
76     if(start == 0)
77     {
78         printf("\nNo Data found for delete.");
79     }
80     else
81     {
82         ds = start;
83         start = start->next;
84         ds->next=0;
85         free(ds);
86         printf("\nDeleted Succesfully");
87     }
88 }
89 void deleteMiddle()
90 {
91     struct node *dm,*search;
92     int loc_data;
93     if(start==0 || start->next == 0)
94     {
95         printf("\nNot Sufficient data found to apply this operation.");
96         return;
97     }
98     search=start;
99     printf("\nEnter the data that you want to delete :");
100     scanf("%d",&loc_data);
101     if(start->data == loc_data)
102     {
103         printf("\nPlease apply delete from start method.");
104         return;
105     }
106     while(search->next->data != loc_data)
107     {
108         search=search->next;
109     }
110     dm = search->next;
111     search->next=dm->next;
112     dm->next=0;
113     free(dm);
114     printf("\nDeleted Succesfully");
115 }
116 void deleteEnd()
117 {
118     struct node *search,*de;
119     if(start == 0)
120     {
121         printf("\nNo Data found for delete.");
122         return;
123     }
124     if(start->next == 0)
125     {
126         de=start;
127         start=0;
128         free(de);
129     }
130     else
131     {
132         search=start;
133         while(search->next->next != 0)
134         {
135             search=search->next;
136         }
137         de=search->next;
138         search->next=0;

```

```

139         free(de);
140     }
141     printf("\nDeleted Succesfully.");
142 }
143 void display()
144 {
145     struct node *ds;
146     if(start == 0)
147     {
148         printf("\nNo data found in list.");
149     }
150     else
151     {
152         ds=start;
153         printf("\nYour data = [");
154         while(ds != 0)
155         {
156             printf("%d,",ds->data);
157             ds = ds->next;
158         }
159         printf("]");
160     }
161 }
162 int main()
163 {
164     int ch;
165     printf("1. Insert at Start.\n");
166     printf("2. Insert at middle.\n");
167     printf("3. Insert at end.\n");
168     printf("4. Delete from Start.\n");
169     printf("5. Delete from Middle.\n");
170     printf("6. Delete from End.\n");
171     printf("7. Display.\n");
172     printf("8. Exit.");
173     while(1)
174     {
175         printf("\n\n>>Enter your choice :");
176         scanf("%d",&ch);
177         switch(ch)
178         {
179             case 1:
180                 insertStart();
181                 break;
182             case 2:
183                 insertMiddle();
184                 break;
185             case 3:
186                 insertEnd();
187                 break;
188             case 4:
189                 deleteStart();
190                 break;
191             case 5:
192                 deleteMiddle();
193                 break;
194             case 6:
195                 deleteEnd();
196                 break;
197             case 7:
198                 display();
199                 break;
200             case 8:
201                 exit(0);
202                 break;
203             default:
204                 printf("You have entered a invaid input.");
205         }
206     }
207 }

```

```
208     return 0;  
209 }  
210
```