

```

1 #include<stdio.h>
2 #include<stdlib.h>
3
4 struct node
5 {
6     struct node *prev;
7     char data;
8     struct node *next;
9 };
10 struct node *start = 0;
11 struct node *CreateNode();
12 void InsertStart();
13 void InsertMiddle();
14 void InsertEnd();
15 void DeleteStart();
16 void DeleteMiddle();
17 void DeleteEnd();
18 void display();
19
20 int main()
21 {
22     int ch;
23     printf("\n 1. Insert at start.");
24     printf("\n 2. Insert at middle.");
25     printf("\n 3. Insert at end.");
26     printf("\n 4. Delete The start.");
27     printf("\n 5. Delete The middle one.");
28     printf("\n 6. Delete The end.");
29     printf("\n 7. Display.");
30     printf("\n 8. Exit.");
31     while(1)
32     {
33         printf("\n\n>Enter your choice :");
34         scanf("%d",&ch);
35         (void)getchar();
36         switch(ch)
37         {
38             case 1:
39                 InsertStart();
40                 break;
41             case 2:
42                 InsertMiddle();
43                 break;
44             case 3:
45                 InsertEnd();
46                 break;
47             case 4:
48                 DeleteStart();
49                 break;
50             case 5:
51                 DeleteMiddle();
52                 break;
53             case 6:
54                 DeleteEnd();
55                 break;
56             case 7:
57                 display();
58                 break;
59             case 8:
60                 exit(0);
61             default:
62                 printf("\n You have enterd a wrong input.");
63         }
64     }
65     return 0;
66 }
67
68 struct node *CreateNode()
69 {

```

```

70     struct node *cn;
71     cn = (struct node*)malloc(sizeof(struct node));
72     printf("\n Enter data :");
73     scanf("%c",&cn->data);
74     cn->prev = 0;
75     cn->next=0;
76     return cn;
77 }
78 void InsertStart()
79 {
80     struct node *is;
81     is = CreateNode();
82     if(start == 0)
83     {
84         start = is;
85     }
86     else
87     {
88         is->next = start;
89         start->prev = is;
90         start = is;
91     }
92 }
93 void InsertMiddle()
94 {
95     struct node *im,*search;
96     char data;
97     im = CreateNode();
98     if(start == 0)
99     {
100         start = im;
101     }
102     else
103     {
104         printf("\n-->Enter data which after you want to insert:");
105         (void)getchar();
106         scanf("%c",&data);
107         search = start;
108         while(search != 0 && search->data != data)
109         {
110             search = search->next;
111         }
112         if(search == 0)
113         {
114             printf("\n%c is not found.",data);
115         }
116         if(search -> next == 0)
117         {
118             printf("\nCannot apply this method , please try insert the method.");
119             return;
120         }
121         im->next = search->next;
122         search->next->prev = im;
123         search->next = im;
124         im->prev = search;
125     }
126 }
127 void InsertEnd()
128 {
129     struct node *ie,*search;
130     ie = CreateNode();
131     if(start == 0)
132     {
133         start = ie;
134     }
135     else
136     {
137         search = start;
138         while(search->next != 0)

```

```

139     {
140         search = search->next;
141     }
142     search->next = ie;
143     ie->prev = search;
144 }
145 }
146
147 void DeleteStart()
148 {
149     struct node *ds;
150     if(start == 0)
151     {
152         printf("\n No node present for delete.");
153         return;
154     }
155     else if(start->next == 0)
156     {
157         ds = start;
158         start = 0;
159         free(ds);
160     }
161     else
162     {
163         ds = start;
164         start = start->next;
165         start->prev = 0;
166         ds->next = 0;
167         free(ds);
168     }
169     printf("\n Deleted succesfully.");
170 }
171 void DeleteEnd()
172 {
173     struct node *search,*de;
174     if(start == 0)
175     {
176         printf("\nNo data present for delete.");
177     }
178     else if(start->next == 0)
179     {
180         DeleteStart();
181     }
182     else
183     {
184         search = start;
185         while(search->next->next != 0)
186         {
187             search = search->next;
188         }
189         de = search->next;
190         search->next = 0;
191         de->prev = 0;
192         free(de);
193         printf("\nDeleted succesfully.");
194     }
195 }
196 void DeleteMiddle()
197 {
198     struct node *dm,*search;
199     char data;
200     if(start == 0)
201     {
202         printf("\n No data present to be delete.");
203     }
204     else if(start->next == 0)
205     {
206         DeleteStart();
207     }

```

```

208 else
209 {
210     printf("\n--> Enter the data that you want to delete :");
211     scanf("%c",&data);
212     if(start->data == data)
213     {
214         printf("\nCannot apply this method , please apply delete the start method.");
215         return;
216     }
217     search = start;
218     while( search->next != 0 && search->next->data != data)
219     {
220         search = search->next;
221     }
222     if(search ->next == 0)
223     {
224         printf("\n%c is not found.",data);
225         return;
226     }
227     dm = search->next;
228     if(dm->next == 0)
229     {
230         printf("\nCannot apply this method,please apply delete the end method.");
231         return;
232     }
233     search->next = dm->next;
234     dm->next->prev = search;
235     dm->prev = 0;
236     dm->next=0;
237     free(dm);
238     printf("\nDeleted succesfully");
239 }
240 }
241
242 void display()
243 {
244     struct node *d;
245     if(start ==0)
246     {
247         printf("\n There are no any data to be print.");
248     }
249     else
250     {
251         d = start;
252         printf("\nFrom start to end : [");
253         while(d->next != 0)
254         {
255             printf("%c ",d->data);
256             d = d->next;
257         }
258         printf("%c ]",d->data);
259         printf("\nFrom end to start : [");
260         while(d->prev != 0)
261         {
262             printf("%c ",d->data);
263             d=d->prev;
264         }
265         printf("%c ]",d->data);
266     }
267 }
268

```