

```

1 #include<stdio.h>
2 #include<stdlib.h>
3 struct node
4 {
5     int data;
6     struct node *next;
7 };
8 struct node *start;
9 struct node *createnode();
10 void InsertStart();
11 void InsertMiddle();
12 void InsertEnd();
13 void DeleteStart();
14 void DeleteMiddle();
15 void DeleteEnd();
16 void display();
17
18 int main()
19 {
20     int ch;
21     printf("\n1. Insert at start.");
22     printf("\n2. Insert at middle.");
23     printf("\n3. Insert at end.");
24     printf("\n4. Delete from start.");
25     printf("\n5. Delete from middle.");
26     printf("\n6. Delete from end.");
27     printf("\n7. Display.");
28     printf("\n8. Exit.");
29     choice:
30     printf("\n\n>Enter your choice:");
31     scanf("%d",&ch);
32     switch(ch)
33     {
34         case 1:
35             InsertStart();
36             break;
37         case 2:
38             InsertMiddle();
39             break;
40         case 3:
41             InsertEnd();
42             break;
43         case 4:
44             DeleteStart();
45             break;
46         case 5:
47             DeleteMiddle();
48             break;
49         case 6:
50             DeleteEnd();
51             break;
52         case 7:
53             display();
54             break;
55         case 8:
56             exit(0);
57         default:
58             printf("\nWrong input ! Try again.");
59     }
60     goto choice;
61     return 0;
62 }
63
64 struct node *createnode()
65 {
66     struct node *cn;
67     cn = (struct node*)malloc(sizeof(struct node));
68     printf("\nEnter data:");
69     scanf("%d",&cn->data);

```

```

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

```

```

139     search->next = ie;
140     ie->next = start;
141 }
142 }
143 void DeleteStart()
144 {
145     struct node *ds,*search;
146     if(start == 0)
147     {
148         printf("\nNo node aviable to delete.");
149     }
150     else if(start == start->next)
151     {
152         ds=start;
153         start=0;
154         free(ds);
155     }
156     else
157     {
158         search=start;
159         while(search->next != start)
160         {
161             search=search->next;
162         }
163         ds=start;
164         start=start->next;
165         search->next = start;
166         ds->next = 0;
167         free(ds);
168     }
169     printf("\nDeleted Succesfully.");
170 }
171 }
172 void DeleteMiddle()
173 {
174     struct node *dm,*search;
175     int data;
176     if(start==0)
177     {
178         printf("\nNo data aviable to delete.");
179     }
180     else
181     {
182         printf("\nEnter data tharts you want to delete:");
183         scanf("%d",&data);
184         if(data == start->data)
185         {
186             printf("\nSorry! Cannot delete . Please try delete from start method.");
187             return;
188         }
189         search=start;
190         while(search->next->data != data)
191         {
192             search=search->next;
193             if(search==start)
194             {
195                 printf("\n%d is not found.",data);
196                 return;
197             }
198         }
199         dm=search->next;
200         search->next = dm->next;
201         dm->next = 0;
202         free(dm);
203         printf("\nDeleted Succesfully.");
204     }
205 }
206 void DeleteEnd()
207 {

```

```

208     struct node *de,*search;
209     if(start == 0)
210     {
211         printf("\nNo data aviable to delete.");
212     }
213     else if(start->next == start)
214     {
215         de = start;
216         start = 0;
217         free(de);
218     }
219     else
220     {
221         search=start;
222         while(search->next->next != start)
223         {
224             search=search->next;
225         }
226         de=search->next;
227         search->next = de->next;
228         de->next = 0;
229         free(de);
230     }
231     printf("\nDeleted Succesfully.");
232 }
233 void display()
234 {
235     struct node *d;
236     if(start == 0)
237     {
238         printf("\nNo data aviable to print");
239     }
240     else
241     {
242         d = start;
243         printf("\nYour data = [");
244         do
245         {
246             printf("%d ",d->data);
247             d=d->next;
248         }
249         while(d != start);
250         printf("]");
251     }
252 }
```