```c
#include<stdio.h>
#include<stdlib.h>
struct node
{
   struct node *prev;
   int n;
   struct node *next;
}*h,*temp,*temp1,*temp2,*temp3;
void beg_insert();
void end_insert();
void spe_insert();
void display();
void search();
void idelete();
int count=0;
void main()
{
   int ch;
   h=NULL;
   temp=temp1=NULL;
   printf("\n 1.Insert at beggining");
   printf("\n 2.Insert at end");
   printf("\n 3.Insert at specific location");
   printf("\n 4.Delete at specific location");
   printf("\n 5.Display at specific loca
   printf("\n 6.Search for element");
```

```
27      printf("\n 7.Exit");
28      while(1)
29      {
30          printf("\n Enter choice:");
31          scanf("%d",&ch);
32          switch(ch)
33          {
34              case 1:beg_insert();
35              break;
36              case 2:end_insert();
37              break;
38              case 3:spe_insert();
39              break;
40              case 4:idelete();
41              break;
42              case 5:display();
43              break;
44              case 6:search();
45              break;
46              case 7:exit(0);
47              default:printf("\n Wrong choice menu");
48          }
49      }
50  }
51  void create()
52  {
```

```
52  {
53          int data;
54          temp=(struct node*)malloc(1*
    sizeof(struct node));
55          temp->prev=NULL;
56          temp->next=NULL;
57          printf("Enter value to node:");
58          scanf("%d",&data);
59          temp->n=data;
60          count++;
61  }
62  void beg_insert()
63  {
64          if(h==NULL)
65          {
66             create();
67             h=temp;
68             temp1=h;
69          }
70          else
71  {
72          create();
73          temp->next=h;
74          h->prev=temp;
75          h=temp;
76  }
```

```
77  }
78  void end_insert()
79  {
80          if(h==NULL)
81          {
82             create();
83             h=temp;
84             temp1=h;
85          }
86          else
87  {
88          create();
89          temp->next=temp;
90          temp->prev=temp1;
91          temp1=temp;
92  }
93  }
94  void spe_insert()
95  {
96          int pos,i=2;
97          printf("Enter position to be inserted:");
98          scanf("%d",&pos);
99          temp2=h;
100         if((pos<1)||(pos>=count+1))
101         {
102            printf("Position out of rang
```

```
102             printf("Position out of range to
    insert\n");
103             return;
104         }
105         if((h==NULL)&&(pos!=1))
106         {
107             printf("Empty list cannot insert other
    than 1st position\n");
108             return;
109         }
110         if((h==NULL)&&(pos==1))
111         {
112             create();
113             h=temp;
114             temp1=h;
115             return;
116         }
117         else
118         {
119             while(i<pos)
120             {
121                 temp2=temp2->next;
122                 i++;
123             }
124             create();
125             temp->prev=temp2;
```

```
125             temp->prev=temp2;
126             temp->next=temp2->next;
127             temp2->next->prev=temp;
128             temp2->next=temp;
129         }
130 }
131 void idelete()
132 {
133         int i=1,pos;
134         printf("Enter position to be deleted:");
135         scanf("%d",&pos);
136         temp2=h;
137         if((pos<1)||(pos>=count+1))
138         {
139             printf("Position out of range to
    delete\n");
140             return;
141         }
142         if(h==NULL)
143         {
144             printf("Empty list no elements to
    delete\n");
145             return;
146         }
147         else
148 {
```

```
148 {
149         while(i<pos)
150         {
151            temp2=temp2->next;
152            i++;
153         }
154         if(i==1)
155         {
156            if(temp2->next==NULL)
157            {
158               printf("Node deleted from list");
159               free(temp2);
160               temp2=h=NULL;
161               return;
162            }
163         }
164         if(temp2->next==NULL)
165         {
166            temp2->prev->next=NULL;
167            free(temp2);
168            printf("Node deleted from list");
169            return;
170         }
171         temp2->next->prev=temp2->pr
172         if(i!=1)
173         temp2->prev->next=temp2->n
```

```
173             temp2->prev->next=temp2->next;
174             if(i==1)
175             h=temp2->next;
176             printf("Node deleted \n");
177             free(temp2);
178 }
179 count--;
180 }
181 void display()
182 {
183             temp2=h;
184             if(temp2==NULL)
185             {
186                printf("List empty to display\n");
187                return;
188             }
189             printf("Linked list elements from
    beginning:");
190             while(temp2->next!=NULL)
191             {
192                printf("%d",temp2->n);
193                temp2=temp2->next;
194             }
195             printf("%d",temp2->n);
196 }
197 void search()
```

```
198 {
199         int data,count =0;
200         temp2=h;
201         if(temp2==NULL)
202         {
203             printf("\n Error:List empty to search
    for data");
204             return;
205         }
206         printf("\n Enter value to search:");
207         scanf("%d",&data);
208         while(temp2!=NULL)
209         {
210             if(temp2->n==data)
211             {
212                 printf("Data found in %d position",
    count+1);
213                 return;
214             }
215             else
216 temp2=temp2->next;
217 count++;
218         }
219         printf("\n Error:%d not found in list",
    data);
220 }
```

```
 1.Insert at beggining
 2.Insert at end
 3.Insert at specific location
 4.Delete at specific location
 5.Display at specific location
 6.Search for element
 7.Exit
 Enter choice:1
Enter value to node:2

 Enter choice:1
Enter value to node:3

 Enter choice:2
Enter value to node:5

 Enter choice:3
Enter position to be inserted:4
Position out of range to insert

 Enter choice:4
Enter position to be deleted:2
Node deleted from list
 Enter choice:5
Linked list elements from beginning:3
 Enter choice:6

 Enter value to search:3
Data found in 1 position
 Enter choice:7

[Program finished]
```