```c
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
  int info;
  struct node *link;
};
typedef struct node *NODE;
NODE getnode()
{
NODE x;
x=(NODE)malloc(sizeof(struct node));
if(x==NULL)
 {
  printf("mem full\n");
  exit(0);
 }
 return x;
}
void freenode(NODE x)
{
free(x);
}
NODE insert_front(NODE first,int item)
{
NODE temp;
temp=getnode();
temp->info=item;
temp->link=NULL;
if(first==NULL)
return temp;
temp->link=first;
first=temp;
return first;
}
NODE delete_front(NODE first)
{
```

```
35  return first;
36  }
37  NODE delete_front(NODE first)
38  {
39  NODE temp;
40  if(first==NULL)
41  {
42  printf("List is empty cannot delete\n");
43  return first;
44  }
45  temp=first;
46  temp=temp->link;
47  printf("Item deleted at front-end is=%d\n",first->info);
48  free(first);
49  return temp;
50  }
51  NODE insert_rear(NODE first,int item)
52  {
53  NODE temp,cur;
54  temp=getnode();
55  temp->info=item;
56  temp->link=NULL;
57  if(first==NULL)
58   return temp;
59  cur=first;
60  while(cur->link!=NULL)
61   cur=cur->link;
62  cur->link=temp;
63  return first;
64  }
65  NODE delete_rear(NODE first)
66  {
67  NODE cur,prev;
68  if(first==NULL)
69  {
70  printf("List is empty cannot delete\n");
71  return first;
72  }
```

```
70  printf("List is empty cannot delete\n");
71  return first;
72  }
73  if(first->link==NULL)
74  {
75  printf("Item deleted is %d\n",first->info);
76  free(first);
77  return NULL;
78  }
79  prev=NULL;
80  cur=first;
81  while(cur->link!=NULL)
82  {
83  prev=cur;
84  cur=cur->link;
85  }
86  printf("Item deleted at rear-end is %d",cur->info);
87  free(cur);
88  prev->link=NULL;
89  return first;
90  }
91  NODE order_list(int item,NODE first)
92  {
93  NODE temp,prev,cur;
94  temp=getnode();
95  temp->info=item;
96  temp->link=NULL;
97  if(first==NULL) return temp;
98  if(item<first->info)
99  {
100 temp->link=first;
101 return temp;
102 }
103 prev=NULL;
104 cur=first;
105 while(cur!=NULL&&item>cur->info)
106 {
107 prev=cur;
```

```
105  while(cur!=NULL&&item>cur->info)
106  {
107  prev=cur;
108  cur=cur->link;
109  }
110  prev->link=temp;
111  temp->link=cur;
112  return first;
113  }
114
115  void display(NODE first)
116  {
117   NODE temp;
118   if(first==NULL)
119   printf("List empty cannot display items\n");
120   printf("Contents of the list:\n");
121   for(temp=first;temp!=NULL;temp=temp->link)
122    {
123    printf("%d\n",temp->info);
124    }
125  }
126  NODE concat(NODE first,NODE second)
127  {
128   NODE cur;
129   if(first==NULL)
130    return second;
131   if(second==NULL)
132    return first;
133   cur=first;
134   while(cur->link!=NULL)
135    cur=cur->link;
136   cur->link=second;
137   return first;
138  }
139  NODE reverse(NODE first)
140   {
141   NODE cur,temp;
142   cur=NULL;
```

```
142   cur=NULL;
143   while(first!=NULL)
144    {
145     temp=first;
146     first=first->link;
147     temp->link=cur;
148     cur=temp;
149    }
150   return cur;
151  }
152
153  int main()
154  {
155  int item,choice,n,i;
156  NODE first=NULL,a,b;
157  for(;;)
158  {
159  printf("\n1:Insert_front\n2:Delete_front\n3:Insert_rear\n4:Delete_rear\n");
160  printf("5:Order_list\n6:Display_list\n7:Concat\n8:Reverse\n9:Exit\n");
161  printf("Enter the choice :");
162  scanf("%d",&choice);
163  switch(choice)
164   {
165    case 1:printf("Enter the item at front-end\n");
166       scanf("%d",&item);
167       first=insert_front(first,item);
168       break;
169    case 2:first=delete_front(first);
170       break;
171    case 3:printf("Enter the item at rear-end\n");
172       scanf("%d",&item);
173       first=insert_rear(first,item);
174       break;
175    case 4:first=delete_rear(first);
176       break;
177    case 5:printf("Enter the item to be inserted in ordered_list\n");
178       scanf("%d",&item);
179       first=order_list(item,first);
```

```
178        scanf("%d",&item);
179        first=order_list(item,first);
180        break;
181
182     case 6:display(first);
183        break;
184     case 7:printf("Enter the no of nodes in 1\n");
185            scanf("%d",&n);
186            a=NULL;
187            for(i=0;i<n;i++)
188             {
189              printf("Enter the item\n");
190              scanf("%d",&item);
191              a=insert_rear(a,item);
192             }
193             printf("Enter the no of nodes in 2\n");
194            scanf("%d",&n);
195            b=NULL;
196            for(i=0;i<n;i++)
197             {
198              printf("Enter the item\n");
199              scanf("%d",&item);
200              b=insert_rear(b,item);
201             }
202             a=concat(a,b);
203             display(a);
204            break;
205      case 8:first=reverse(first);
206            display(first);
207            break;
208     case 9:exit(0);
209        break;
210        default:printf("Invalid choice\n");
211   }
212  }
213
214  }
215
```

C:\WINDOWS\SYSTEM32\cmd.exe

```
1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:Order_list
6:Display_list
7:Concat
8:Reverse
9:Exit
Enter the choice :1
Enter the item at front-end
1

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:Order_list
6:Display_list
7:Concat
8:Reverse
9:Exit
Enter the choice :1
Enter the item at front-end
2

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:Order_list
6:Display_list
7:Concat
8:Reverse
9:Exit
Enter the choice :1
Enter the item at front-end
3

1:Insert_front
2:Delete_front
```

C:\WINDOWS\SYSTEM32\cmd.exe

```
3:Insert_rear
4:Delete_rear
5:Order_list
6:Display_list
7:Concat
8:Reverse
9:Exit
Enter the choice :6
Contents of the list:
3
2
1

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:Order_list
6:Display_list
7:Concat
8:Reverse
9:Exit
Enter the choice :5
Enter the item to be inserted in ordered_list
5

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:Order_list
6:Display_list
7:Concat
8:Reverse
9:Exit
Enter the choice :5
Enter the item to be inserted in ordered_list
10

1:Insert_front
2:Delete_front
3:Insert_rear
```

C:\WINDOWS\SYSTEM32\cmd.exe

```
1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:Order_list
6:Display_list
7:Concat
8:Reverse
9:Exit
Enter the choice :5
Enter the item to be inserted in ordered_list
4

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:Order_list
6:Display_list
7:Concat
8:Reverse
9:Exit
Enter the choice :6
Contents of the list:
3
2
1
4
5
10

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:Order_list
6:Display_list
7:Concat
8:Reverse
9:Exit
Enter the choice :4
Item deleted at rear-end is 10
```

C:\WINDOWS\SYSTEM32\cmd.exe

```
9:Exit
Enter the choice :4
Item deleted at rear-end is 10
1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:Order_list
6:Display_list
7:Concat
8:Reverse
9:Exit
Enter the choice :4
Item deleted at rear-end is 5
1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:Order_list
6:Display_list
7:Concat
8:Reverse
9:Exit
Enter the choice :4
Item deleted at rear-end is 4
1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:Order_list
6:Display_list
7:Concat
8:Reverse
9:Exit
Enter the choice :6
Contents of the list:
3
2
1

1:Insert_front
2:Delete_front
```

C:\WINDOWS\SYSTEM32\cmd.exe

```
1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:Order_list
6:Display_list
7:Concat
8:Reverse
9:Exit
Enter the choice :7
Enter the no of nodes in 1
2
Enter the item
1
Enter the item
4
Enter the no of nodes in 2
2
Enter the item
9
Enter the item
8
Contents of the list:
1
4
9
8

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:Order_list
6:Display_list
7:Concat
8:Reverse
9:Exit
Enter the choice :8
Contents of the list:
1
2
```

C:\WINDOWS\SYSTEM32\cmd.exe

```
1
4
9
8

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:Order_list
6:Display_list
7:Concat
8:Reverse
9:Exit
Enter the choice :8
Contents of the list:
1
2
3

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5:Order_list
6:Display_list
7:Concat
8:Reverse
9:Exit
Enter the choice :9


------------------
(program exited with code: 0)

Press any key to continue . . .
```