main.c

```c
#include<stdio.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;
}
int freenode(NODE x)
{
free(x);
return 0;
}
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)
{
NODE temp;
if(first==NULL)
{
printf("list is empty cannot delete\n");
return first;
}
```

main.c

```
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  }
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("iten deleted at rear-end is %d",cur->info);
87  free(cur);
88  prev->link=NULL;
```

main.c

```
88   prev->link=NULL;
89
90   return first;
91   }
92   void display(NODE first)
93   {
94   NODE temp;
95   if(first==NULL)
96   printf("list empty cannot display items\n");
97   for(temp=first;temp!=NULL;temp=temp->link)
98   {
99   printf("%d\n",temp->info);
100  }
101  }
102  NODE delete_pos(int pos, NODE first)
103  {
104  NODE cur;
105  NODE prev;
106  int count;
107  if(first==NULL || pos<=0)
108  {
109  printf("invalid position \n");
110  return NULL;
111  }
112  if (pos==1)
113  {
114  cur=first;
115  first=first->link;
116  freenode(cur);
117  return first;
118  }
119  prev=NULL;
120  cur=first;
121  count=1;
122  while(cur!=NULL)
123  {
124  if(count==pos) break;
125  prev=cur;
126  cur=cur->link;
127  count++;
128  }
129  if(count!=pos)
130  {
131  printf("invalid position \n");
```

main.c

```
{
printf("invalid position \n");
return first;
}
if(count!=pos)
{
printf("invalid position specified \n");
return first;
}
prev->link=cur->link;
freenode(cur);
return first;}
int main{
int item,choice,pos;
NODE first=NULL;
system("cls");
for(;;)
{printf("\n 1:Insert_front\n 2:Delete_front\n 3:Insert_rear\n
4:Delete_rear\n 5.Delete at specified position \n 6:Display_list\n
7:Exit\n");
printf("enter the choice\n");
scanf("%d",&choice);
switch(choice)
{
case 1:printf("enter the item at front-end\n");
scanf("%d",&item);
first=insert_front(first,item);
break;
case 2:first=delete_front(first);
break;
case 3:printf("enter the item at rear-end\n");
scanf("%d",&item);
first=insert_rear(first,item);
break;
case 4:first=delete_rear(first);
break;
case 5:printf("enter the position of the item to be deleted: \n");
        scanf("%d",&pos);
        first=delete_pos(pos,first);
    break;
case 6:display(first);
break;
default:exit(0);
break;}}
return 0;}
```

Console

```
1
enter the item at front-end
23

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5.Delete at specified position
 6:Display_list
 7:Exit
enter the choice
1
enter the item at front-end
43

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5.Delete at specified position
 6:Display_list
 7:Exit
enter the choice
1
enter the item at front-end
52

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5.Delete at specified position
 6:Display_list
 7:Exit
enter the choice
```

Console

```
enter the choice
1
enter the item at front-end
52

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5.Delete at specified position
 6:Display_list
 7:Exit
enter the choice
2
item deleted at front-end is=52

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5.Delete at specified position
 6:Display_list
 7:Exit
enter the choice
3
enter the item at rear-end
18

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5.Delete at specified position
 6:Display_list
 7:Exit
enter the choice
```

Console

```
3
enter the item at rear-end
18

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5.Delete at specified position
 6:Display_list
 7:Exit
enter the choice
5
enter the position of the item to be deleted:
3

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5.Delete at specified position
 6:Display_list
 7:Exit
enter the choice
6
43
23

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5.Delete at specified position
 6:Display_list
 7:Exit
enter the choice
```