main.c

```c
#include<stdio.h>
#include<stdlib.h>
#include<conio.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("memory full\n");
        exit(0);
    }
    return x;
}
void freenode(NODE x)
{
    free(x);
}
NODE insert_front(NODE first,int item)
```

main.c

```
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;
    }
    temp=first;
    temp=temp->link;
    printf("item deleted at front end is=%d \n",first->info);
    free(first);
    return temp;
}
NODE insert_rear(NODE first,int item)
```

main.c

```
51  }
52  NODE insert_rear(NODE first,int item)
53  {
54      NODE temp,cur;
55      temp=getnode();
56      temp->info=item;
57      temp->link=NULL;
58      if(first==NULL)
59      return temp;
60      cur=first;
61      while(cur->link!=NULL)
62      cur=cur->link;
63      cur->link=temp;
64      return first;
65  }
66  NODE delete_rear(NODE first)
67  {
68      NODE cur,prev;
69      if(first==NULL)
70      {
71          printf("list is empty cannot delete\n");
72          return first;
73      }
74      if(first->link==NULL)
75      {
76          printf("item deleted is %d\n",first->info);
```

main.c

```
75      {
76          printf("item deleted is %d\n",first->info);
77          free(first);
78          return NULL;
79      }
80      prev=NULL;
81      cur=first;
82      while(cur->link!=NULL)
83      {
84          prev=cur;
85          cur=cur->link;
86      }
87      printf("item deleted at rear end is %d",cur->info);
88      free(cur);
89      prev->link=NULL;
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     }
```

main.c

```
        printf("%d\n",temp->info);
    }
}
void main()
{
    int item,choice;
    NODE first=NULL;
    for(;;)
    {
        printf("\n 1:Insert_front\n 2:Delete_front\n 3:Insert_rear\n 4:Delete_rear\n 5:Display_list\n 6:EXIT\n");
        printf("enter the choice\n");
        scanf("%d",&choice);
        printf("---------\n");
        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);
```

main.c

```
111         printf("---------\n");
112         switch(choice)
113         {
114             case 1:printf("enter the item at front end\n");
115             scanf("%d",&item);
116             first=insert_front(first,item);
117             break;
118             case 2:first=delete_front(first);
119             break;
120             case 3:printf("enter the item at rear end\n");
121             scanf("%d",&item);
122             first=insert_rear(first,item);
123             break;
124             case 4:first=delete_rear(first);
125             break;
126             case 5:display(first);
127             break;
128             default:exit(0);
129             break;
130         }
131     }
132 }
133
134
135
136
```

input

```
 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
 6:EXIT
enter the choice
1
---------
enter the item at front end
23

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
 6:EXIT
enter the choice
1
---------
enter the item at front end
34

 1:Insert_front
 2:Delete_front
```

```
 5:Display_list
 6:EXIT
enter the choice
5
---------
34
23

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
 6:EXIT
enter the choice
3
---------
enter the item at rear end
45

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
 6:EXIT
enter the choice
```

```
 3:Insert_rear
 4:Delete_rear
 5:Display_list
 6:EXIT
enter the choice
3
--------
enter the item at rear end
78

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
 6:EXIT
enter the choice
5
--------
34
23
45
78

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

```
 3:Insert_rear
 4:Delete_rear
 5:Display_list
 6:EXIT
enter the choice
2
---------
item deleted at front end is=34

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
 6:EXIT
enter the choice
4
---------
item deleted at rear end is 78
 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
 6:EXIT
```

```
enter the choice
2
---------
item deleted at front end is=23

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
 6:EXIT
enter the choice
4
---------
item deleted is 45

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
 6:EXIT
enter the choice
4
---------
list is empty cannot delete
```

```
list is empty cannot delete

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
 6:EXIT
enter the choice
5
---------
list empty cannot display items

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
 5:Display_list
 6:EXIT
enter the choice
6
---------


...Program finished with exit code 0
Press ENTER to exit console.
```