```
#include<stdio.h>
#include<stdlib.h>
int n=0;
struct node{
    int data;
    struct node* plink;
    struct node *nlink;
};
struct node *getnode(int val)
{
    n++;
    struct node *temp=(struct node*)malloc(sizeof(struct node));
    temp->nlink=temp;
    temp->plink=temp;
    temp->data=val;
    return temp;
}
struct node *insertbyorder(struct node *head,int val)
{
    struct node *temp=getnode(val);
    struct node *cur=NULL;
    struct node *prev=NULL;
    if(head==NULL)
    return temp;
    else if(val<head->data)
    {
        cur=head;
        temp->nlink=cur;
        temp->plink=cur->plink;
        cur->plink->nlink=temp;
        cur->plink=temp;
        return temp;
    }
    else if(val>head->plink->data){
        cur=head->plink;
        cur->nlink=temp;
        temp->nlink=head;
        temp->plink=cur;
        head->plink=temp;
        return head;
    }
    else{
        cur=head;
        while(cur->data<val){
            prev=cur;
            cur=cur->nlink;
        }
        temp->nlink=cur;
        cur->plink=temp;
```

```
        prev->nlink=temp;
        temp->plink=prev;
        return head;
    }
}
struct node *deletebypos(struct node *head,int pos)
{
    struct node *cur=NULL;
    struct node *next=NULL;
    struct node *prev=NULL;
    if(pos>n){
        printf("invalid pos");
        return head;
    }
    if(n==1)
    {
        n--;
        return NULL;
    }
    cur=head;
    next=cur->nlink;
    prev=head->plink;
    while(pos!=1)
    {
        prev=cur;
        cur=next;
        next=next->nlink;
        pos--;
    }
    prev->nlink=next;
    next->plink=prev;
    n--;
    if(cur==head)
    {
        free(cur);
        return next;
    }
    free(cur);
    return head;
}
struct node *insertfront(struct node *head,int val)
{
    struct node *temp=getnode(val);
    struct node *cur=NULL;
    if(head==NULL)
    return temp;
    cur=head->plink;
    temp->nlink=head;
    temp->plink=cur;
```

```
    cur->nlink=temp;
    return temp;
}
struct node *insertrear(struct node *head,int val)
{
    struct node *cur=NULL;
    struct node *temp=getnode(val);
    if(head==NULL)
    return temp;
    cur=head->plink;
    cur->nlink=temp;
    temp->plink=cur;
    temp->nlink=head;
    head->plink=temp;
    return head;
}
struct node *deletebykey(struct node *head,int key)
{
    int flag=0;
    struct node *cur=NULL;
    struct node *next=NULL;
    struct node *prev=NULL;
    if(head==NULL)
    return head;
    if(head->data==key && n==1)
    {
        n--;
        return NULL;
    }
    prev=head->plink;
    cur=head;
    next=cur->nlink;
    do{
        if(cur->data==key){
        flag=1;
        break;
        }
        prev=cur;
        cur=next;
        next=next->nlink;
    }while(cur!=head);
    if(flag==1)
    {
        prev->nlink=next;
        next->plink=prev;
        n--;
        if(cur==head)
        {
            free(cur);
```

```
            return next;
        }
        free(cur);
        return head;
    }
    printf("key not found\n");
    return head;
}
void display(struct node *head)
{
    struct node *cur=NULL;
    if(head==NULL)printf("list is empty\n");
    else{
        cur=head;
        do{
            printf("%d ",cur->data);
            cur=cur->nlink;
        }while(cur!=head);
    }
}
struct node *deletefront(struct node *head)
{
    struct node *cur=NULL;
    struct node *prev=NULL;
    if(head==NULL){
    n--;
    return head;
    }
    if(n==1)
    {
        n--;
        return NULL;
    }
    n--;
    cur=head->plink;
    head=head->nlink;
    head->plink=cur;
    cur->nlink=head;
    free(prev);
    return head;
}
struct node *deleterear(struct node *head)
{
    struct node *cur=NULL;
    struct node *prev=NULL;
    if(head==NULL)
    return NULL;
    if(n==1)
    {
```

```
        n--;
        return NULL;
    }
    cur=head->plink;
    prev=cur->plink;
    prev->nlink=head;
    head->plink=prev;
    free(cur);
    n--;
    return head;
}
void main()
{
    int val,choice,pos;
    struct node *head=NULL;
    printf("main
menu\n1.insertbyorder\n2.deletebypos\n3.deletebykey\n4.insertfront\n5.ins
ertrear\n6.deletfront\n7.deleterear\n8.exit\n");
   for(;;)
   {
       printf("enter your choice: ");
       scanf("%d",&choice);
       switch(choice)
       {
           case 1:printf("enter the value to be inserted: ");
                    scanf("%d",&val);
                    head=insertbyorder(head,val);
                    display(head);
                    break;
            case 2:printf("enter the pos: ");
                    scanf("%d",&pos);
                    head=deletebypos(head,pos);
                    display(head);
                    break;
            case 3:printf("enter the  key: ");
                    scanf("%d",&val);
                    head=deletebykey(head,val);
                    display(head);
                    break;
            case 4:printf("enter the value: ");
                    scanf("%d",&val);
                    head=insertfront(head,val);
                    display(head);
                    break;
             case 5:printf("enter the value: ");
                    scanf("%d",&val);
                    head=insertrear(head,val);
                    display(head);
                    break;
```

```
            case 6:head=deletefront(head);
                   display(head);
                   break;
            case 7:head=deleterear(head);
                   display(head);
                   break;
            case 8:exit(0);
            default:printf("invalid choice");
        }
    }
}
```