#include<stdlib.h>

#include<stdio.h>

#include "adt.h"

void insertFront(struct listADT \*l,char c)

{

    struct listADT\*temp=(struct listADT\*)malloc(sizeof(struct listADT));

    temp->c=c;

    if(l->next==NULL)

    {

        temp->next=l->next;

        l->next=temp;

        temp->prev=l;

    }

    else

    {

        temp->next=l->next;

        l->next=temp;

        temp->prev=l;

        temp->next->prev=temp;

    }

    return;

}

void display(struct listADT \*l)

{

    struct listADT\*ptr=(struct listADT\*)malloc(sizeof(struct listADT));

    ptr=l;

    while(ptr->next!=NULL)

    {

        printf("%c ",ptr->next->c);

        ptr=ptr->next;

    }

    printf("\nReverse order:\n");

    while(ptr!=l)

    {

        printf("%c ",ptr->c);

        ptr=ptr->prev;

    }

    return;

}

void insertEnd(struct listADT \*l,char c)

{

    struct listADT\*temp=(struct listADT\*)malloc(sizeof(struct listADT)),\*ptr=l;

    temp->c=c;

    while(ptr->next!=NULL)

    {

        ptr=ptr->next;

    }

    temp->next=ptr->next;

    temp->prev=ptr;

    ptr->next=temp;

    return;

}

void insertMiddle(struct listADT\*l, char c, char d)

{

    struct listADT\*ptr=l->next;

    while(ptr->c!=c)

    {

        ptr=ptr->next;

    }

    insertFront(ptr,d);

    return;

}

void deleteItem(struct listADT\*l, char c)

{

    struct listADT\*ptr=l,\*temp;

    while(ptr->next->c!=c)

    {

        ptr=ptr->next;

    }

    temp=ptr->next;

    ptr->next=ptr->next->next;

    ptr->next->prev=ptr;

    free(temp);

    return;

}

int searchItem(struct listADT\*l, char c)

{

    struct listADT\*ptr=l->next;

    int count=0;

    while(ptr!=NULL)

    {

        if(ptr->c==c)

            count++;

        ptr=ptr->next;

    }

    return count;

}