#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

typedef struct no{

int info;

struct Tno \*prox;

}Tno;

void inserirInicio(Tno \*\*L, int valor);

void listar(Tno \*L);

void removerPrimeiro(Tno \*\*aux);

void removerF(Tno \*\*L,int n);

void inserirFinal(Tno \*\*L,int n);

void limpar(Tno \*\*L);

int buscar(Tno \*L,int n);

int main(){

Tno \*aux,\*lista = NULL;

int nun,qtd=0;

char op;

do{

printf("1-Inserir valor no inicio da lista\n2-Exibir todos da lista\n3-Remover no inicio da lista\n"

"4-Inserir no final da lista\n5-Remover no fim da lista\n6-Procurar valor na lista"

"7-Limpar lista\n8-Sair\nDigite a opcao: ");

op = getchar();fflush(stdin);

switch(op){

case '1': printf("informar o valor: ");

scanf("%d",&nun);

inserirInicio(&lista, nun);

break;

case '2':

listar(lista);

break;

case '3':

removerPrimeiro(&lista);

break;

case '4':

printf("informar o valor: ");

scanf("%d",&nun);

inserirFinal(&lista,nun);

break;

case '5':

printf("informar o valor: ");

scanf("%d",&nun);

removerF(&lista,nun);

break;

case '6':

printf("Digite o valor a ser buscado: ");

scanf("%d",&nun);

qtd = buscar(lista,nun);

if(aux == NULL)

printf("Valor nao encontrado!\n");

else

printf("Valor encontrado apareceu %i vezes na lista\n",qtd);

case '7':

limpar(&lista);

break;

case '8':

break;

default:

printf("opcao invalida!\n");

break;

}

}while(op != '8');

return 0;

}

void inserirInicio(Tno \*\*L,int valor){

Tno \*novo;

novo=(Tno \*) malloc(sizeof(Tno));

novo->info = valor;

if(\*L==NULL){

\*L=novo;

novo->prox = NULL;

}

else{

novo->prox = \*L;

\*L = novo;

}

}

void listar(Tno \*L){

if(L ==NULL)

printf("lista vazia\n");

else{

while(L != NULL){

printf("%d",L -> info);

L = L -> prox;

}

}

}

void removerPrimeiro(Tno \*\*L){

Tno \*Aux;

if(\*L == NULL)

printf("Lista vazia");

else{

Aux = \*L;

\*L = (\*L)->prox;

free(Aux);

}

}

void inserirFinal(Tno \*\*L,int n){

Tno \*novo, \*aux;

novo=(Tno \*)malloc(sizeof(Tno));

novo->info = n;

novo->prox = NULL;

if(\*L == NULL)

\*L = novo;

else{

Aux = \*L;

while(Aux->prox != NULL){

aux = aux -> prox;

}

aux -> prox = novo;

}

}

void removerF(Tno \*\*L,int n){

}

int Buscar(Tno \*L,int n){

Tno \*aux = L;

int qtd=0;

while(aux != NULL){

if(aux->info == n)

qtd++;

aux = aux->prox;

}

return qtd;

}

void limpar(Tno \*\*L){

Tno \*aux;

aux = (\*L)->prox;

while(\*L!=NULL){

free(\*L);

\*L=aux;

aux = aux->prox;

}

}