Jasmani Israel Martinez Zamorano

Listas dobles

#include <stdio.h>

#include <stdlib.h>

#include<conio.h>

#include <iostream>

//namespace std;

int inicio=-1,elemento;

int tamVector=5,contador=0;

int vacio=-1,ult=0,p=0,a,s;

struct nodo /\* Comienzo de definiciÃ³n del "patrÃ³n". \*/{

int dato;

int sgt;

int ant;

} lista[5];

class opción{

in op=0;

}

void inicializa(){

inicio=-1;

contador=0;

for(p=0;p<tamVector;p++){

lista[p].dato=-1;

lista[p].sgt=-1;

lista[p].ant=-1;

}

}

void buscar\_vacio (){

//i=0;

vacio=0;

while(lista[vacio].dato!=(-1)){

vacio++;

}

}

void buscar(){

//j=inicio;

p=inicio;

while(p!=-1 and lista[p].dato!=elemento){

//i=j;

p=lista[p].sgt;

}

}

void buscar\_ult(){

p=inicio;

ult=0;

while(p!=(-1)){

ult=p;

p=lista[p].sgt;

}

}

void altas(){

if(contador<tamVector){

printf("Introducir elemento:");

scanf("%d",&elemento);

if(contador>0){

buscar\_ult();

buscar\_vacio();

lista[vacio].dato =elemento;

lista[vacio].sgt=-1;

lista[vacio].ant =ult;

lista[ult].sgt =vacio;

}

else{

inicio=0;

lista[inicio].dato =elemento;

lista[inicio ].sgt =-1;

lista[inicio ].ant =-1;

}

contador ++;

}

else{

printf("Lista llena \n");

}

}

void MLista(){

p=inicio;

while(p!=(-1))

{

printf("[%d]<--Anterior DATO=%d en lista[%d] Siguiente-->[%d] \n ",lista[p].ant,lista[p].dato,p,lista[p].sgt);

p=lista[p].sgt;

}

}

void MAlma(){

for(p=0;p<tamVector;p++)

{

printf("lista[%d]<--Anterior lista[%d].DATO=%d Siguiente-->lista[%d] \n ",lista[p].ant,p,lista[p].dato,lista[p].sgt);

}

}

void bajas(){

if (contador>0)

{

printf("Elemento a eliminar:");

scanf("%d",&elemento );

buscar();// p es el encontrado si no esta en -1

if (p==-1)

{

printf("Elemento no encontrado \n");

}

else

{

a=lista[p].ant;

s=lista[p].sgt;

lista[a].sgt=lista[p].sgt;

lista[s].ant=lista[p].ant;

lista[p].dato=-1;

contador--;

}

}

else

{ printf("Lista Vacia");

}

}

void bajas\_primero(){

if (contador>0)

{

printf("Elemento a eliminar:");

scanf("%d",&elemento );

buscar();// p es el encontrado si no esta en -1

if (p==-1)

{

printf("Elemento no encontrado \n");

}

else

{

a=lista[p].ant;

s=lista[p].sgt;

lista[a].sgt=lista[p].sgt;

lista[s].ant=lista[p].ant;

lista[p].dato=-1;

contador--;

if(p==inicio){

inicio=lista[p].sgt;

}

}

}

else

{

printf("Lista Vacia");

}

}

void modifica(){

if (contador>0)

{

printf("Elemento a modificar:");

scanf("%d",&elemento );

buscar();

if (p==-1)

{

printf("Elemento no encontrado \n");

}

else

{

printf("Elemento nuevo:");

scanf("%d",&elemento );

lista[p].dato=elemento;

}

}

else

{ printf("Lista Vacia");

}

}

int main() {

inicializa();

do{

printf("Menu \n");

printf("1.Adicionar \n");

printf("2.Eliminar \n");

printf("3.Modificar \n");

printf("4.Mostrar Lista \n");

printf("5.Mostrar Almacenamiento \n");

printf("0. Salir \n");

scanf("%d",&op);

switch(op)

{

case 1: altas();

break;

case 2: bajas\_primero();

break;

case 3: modifica();

break;

case 4: MLista();

break;

case 5: MAlma();

break;

case 0: printf("bye");

break;

default: printf("Opcion inexistente");

}

} while(op!=0);

printf("Gracias por utilizar este programa");

return 0;

}