#include <iostream>

#include<conio.h>

#include<stdio.h>

#include <stdlib.h>

// Gerson Omar Perez Salguero Carné: 2490-16-4198

// Víctor Rene Cortez Sis Carné: 2490-19-14590

using namespace::std;

typedef struct nodo{

int carnet;

int semestre;

string nombre;

string carrera;

struct nodo \*izq, \*der;

}\*ABB;

int numNodos = 0; // nummero de nodos del arbol ABB

int numK = 0, k; // nodos menores que un numero K ingresado

ABB crearNodo(int carnet, string nombre, string carrera, int semestre)

{

ABB nuevoNodo = new(struct nodo);

nuevoNodo->carnet = carnet;

nuevoNodo->nombre = nombre;

nuevoNodo->carrera = carrera;

nuevoNodo->semestre = semestre;

nuevoNodo->izq = NULL;

nuevoNodo->der = NULL;

return nuevoNodo;

}

void insertar(ABB &arbol, int carnet, string nombre, string carrera, int semestre)

{

if(arbol==NULL)

{

arbol = crearNodo(carnet, nombre, carrera, semestre);

cout<<"\n\t Insertado..!"<<endl<<endl;

}

else if(carnet < arbol->carnet)

insertar(arbol->izq, carnet, nombre, carrera, semestre);

else if(carnet > arbol->carnet)

insertar(arbol->der, carnet, nombre, carrera, semestre);

}

void preOrden(ABB arbol)

{

if(arbol!=NULL)

{

cout <<"| Carnet: "<< arbol->carnet <<"| Nombre: "<< arbol->nombre <<"| Carrera: "<< arbol->carrera <<"| Semestre: " << arbol->semestre<<" |"<< "\n\n";

preOrden(arbol->izq);

preOrden(arbol->der);

}

}

void enOrden(ABB arbol)

{

if(arbol!=NULL)

{

enOrden(arbol->izq);

cout <<"| Carnet: "<< arbol->carnet <<"| Nombre: "<< arbol->nombre <<"| Carrera: "<< arbol->carrera <<"| Semestre: " << arbol->semestre<<" |"<< "\n\n";

enOrden(arbol->der);

}

}

void postOrden(ABB arbol)

{

if(arbol!=NULL)

{

enOrden(arbol->izq);

enOrden(arbol->der);

cout <<"| Carnet: "<< arbol->carnet <<"| Nombre: "<< arbol->nombre <<"| Carrera: "<< arbol->carrera <<"| Semestre: " << arbol->semestre<<" |"<< "\n\n";

}

}

bool busquedaRec(ABB arbol, int dato)

{

int r=0; // 0 indica que lo encontre

if(arbol==NULL)

return r;

if(dato<arbol->carnet)

r = busquedaRec(arbol->izq, dato);

else if(dato> arbol->carnet)

r = busquedaRec(arbol->der, dato);

else

{

r = 1;

cout <<"| Carnet: "<< arbol->carnet <<"| Nombre: "<< arbol->nombre <<"| Carrera: "<< arbol->carrera <<"| Semestre: " << arbol->semestre<<" |";

} // son iguales, lo encontre

return r;

}

ABB unirABB(ABB izq, ABB der)

{

if(izq==NULL) return der;

if(der==NULL) return izq;

ABB centro = unirABB(izq->der, der->izq);

izq->der = centro;

der->izq = izq;

return der;

}

void elimina(ABB &arbol, int x)

{

if(arbol==NULL) return;

if(x<arbol->carnet)

elimina(arbol->izq, x);

else if(x>arbol->carnet)

elimina(arbol->der, x);

else

{

ABB p = arbol;

arbol = unirABB(arbol->izq, arbol->der);

delete p;

}

}

void menu()

{

system("cls");

cout << "\n\t\t ..[ ARBOL BINARIO DE BUSQUEDA ].. \n\n";

cout << "\t [1] Insertar elemento \n";

cout << "\t [2] Recorrido Post-Orden \n";

cout << "\t [3] Eliminar elemento \n";

cout << "\t [4] SALIR \n";

cout << "\n\t Ingrese opcion : ";

}

int main()

{

ABB arbol = NULL;

int x;

int carnet;

string nombre;

string carrera;

int semestre;

int op, op2;

//system("color f9"); // para cambiar color a la consola

do

{

menu(); cin>> op;

cout << endl;

switch(op)

{

case 1:

cout << " Nombre : "; cin>> nombre;

cout << " Carnet : "; cin>> carnet;

cout << " Semestre : "; cin>> semestre;

cout << " Carrera : "; cin>> carrera;

insertar(arbol, carnet, nombre, carrera, semestre);

break;

case 2:

if(arbol!=NULL)

{

postOrden(arbol);

system("pause");

break;

}

else

cout << "\n\t Arbol vacio..!";

cout<<"\n\n";

system("pause");

break;

case 3:

if(arbol!=NULL)

{

cout<<" Valor a eliminar (numero de carnet): "; cin>> x;

elimina(arbol, x);

cout << "\n\tEliminado..!";

system("pause");

break;

}

else

cout << "\n\t Arbol vacio..!";

cout<<"\n\n";

system("pause");

break;

case 4:

exit(0);

}

cout<<"\n\n\n";

//system("pause"); // hacer pausa y presionar una tecla para continuar

}while(op!=6);

}