#include<iostream>

#include<conio.h>

#include<stdlib.h>

using namespace std;

struct Nodo {

int dato;

Nodo \*derecho;

Nodo \*izquierdo;

};

//Prototipos

void menu();

Nodo \*crearNodo(int);

void insertarNodo(Nodo \*&,int);

void mostrarArbol(Nodo \*,int);

bool busqueda(Nodo \*,int);

void preOrden(Nodo \*);

Nodo \*arbol = NULL;

int main(){

menu();

getch();

return 0;

}

//FUNCION DE MENU

void menu(){

int dato,opcion,contador=0;

do{

cout<<"\t.:MENU:."<<endl;

cout<<"1. Insertar un nuevo nodo"<<endl;

cout<<"2. Mostrar el arbol completo"<<endl;

cout<<"3. buscar un elemento en el arbol"<<endl;

cout<<"4. Recorrer el arbol en preOrden"<<endl;

cout<<"5. salir"<<endl;

cout<<"opcion: ";

cin>>opcion;

switch (opcion){

case 1: cout<<"\n Digite un numero: ";

cin>>dato;

insertarNodo(arbol,dato); //INSERTAMOS UN NUEVO NODO

cout<<"\n";

system("pause");

break;

case 2: cout<<"\nMostrar el arbol completo:\n\n";

mostrarArbol(arbol,contador);

cout<<"\n";

system("pause");

break;

case 3: cout<<"\nDigite el elemento que desea buscar: ";

cin>>dato;

if(busqueda(arbol,dato) == true){

cout<<"\nEl elemento "<<dato<<" ha sido encontrado en el arbol\n";

}

else{

cout<<"\nElemento no encontrado\n";

}

cout<<"\n";

system("pause");

break;

case 4: cout<<"\nRecorrido en preOrden: ";

preOrden(arbol);

cout<<"\n\n";

system("pause");

}

system("cls"); //para limpiar pantalla

}while(opcion != 5);

}

// FUNCION PARA CREAR UN NUEVO NODO

Nodo \*crearNodo(int n) {

Nodo \*nuevo\_nodo = new Nodo ();

nuevo\_nodo->dato = n;

nuevo\_nodo->derecho = NULL;

nuevo\_nodo->izquierdo = NULL;

return nuevo\_nodo;

}

//FUNCION PARA INSERTAR ELEMENTOS EN EL ARBOL

void insertarNodo (Nodo \*&arbol,int n) {

if(arbol == NULL) { //SI EL ARBOL ESTA VACIO

Nodo \*nuevo\_nodo = crearNodo(n);

arbol = nuevo\_nodo;

}

else{ //SI EL ARBOL TIENENE UN NODO O MAS

int valorRaiz = arbol->dato; //obtener el valor de la raiz

if(n < valorRaiz) { // si el elemento es menor a la raiz insertamos en izquiierdo

insertarNodo(arbol->izquierdo,n);

}else{ //si es mayor en el derecho

insertarNodo(arbol->derecho,n);

}

}

}

//funcion para mostrar el arbolo completo

void mostrarArbol(Nodo \*arbol,int cont){

if(arbol == NULL){

return;

}

else{

mostrarArbol(arbol->derecho,cont+1);

for(int i=0;i<cont;i++){

cout<<" ";

}

cout<<arbol->dato<<endl;

mostrarArbol(arbol->izquierdo,cont+1);

}

}

//funcion para buscar un elemento en el arbol

bool busqueda(Nodo \*arbol,int n){

if(arbol == NULL){ //SI EL ARBOL ESTA VACIO RETORNA FALSE

return false;

}

else

if(arbol->dato == n){ //si el nodo es igual al alemento retorname true

return true;

}

else if(n < arbol->dato) //si ele elemento es menor, ahora busca por la izquierda

return busqueda (arbol->izquierdo,n);

else{

return busqueda(arbol->derecho,n);

}

}

//funcion apra recorrido en profundidad, preorden

void preOrden(Nodo \*arbol){

if(arbol == NULL){

return;

}

else{

cout<<arbol->dato<<" - ";

preOrden(arbol->izquierdo);

preOrden(arbol->derecho);

}

}

#include <iostream>

#include <stdlib.h>

#include <time.h>

using namespace std;

//Input : Numero de Elementos del Arbol

typedef struct ABB {

int info;

ABB \*izq, \*der;

ABB() {

info = NULL;

izq = der = NULL;

}

}ARBOL;

ABB \*RAIZ = NULL; //Raiz del Arbol

ARBOL \*CrearNodo(int elem) {

ARBOL \*Nuevo = new ARBOL();

Nuevo->info = elem;

return Nuevo;

}

void Insertar(ARBOL \*&arb, int dato) {

ARBOL \*nodo = CrearNodo(dato);

if(arb == NULL) {

arb = nodo;

}else{

if(arb->info > dato) {

Insertar(arb->izq, dato);

}else{

Insertar(arb->der, dato);

}

}

}

void Mostrar(ARBOL \*arb, int cont) {

if(arb == NULL) {

return;

}else{

Mostrar(arb->der, cont + 1);

for(int i = 0; i < cont; i++) {

cout << " ";

}

cout << arb->info << endl;

Mostrar(arb->izq, cont + 1);

}

}

int main(int argc, char \*\*argv) {

int num; cin >> num;

srand((unsigned int)time(NULL));

for(int i = 0; i < num; i++) {

Insertar(RAIZ, (rand() % 20) + 1);

}

Mostrar(RAIZ, 0);

return EXIT\_SUCCESS;

}