**Solucionario del Examen Final de SI202-Algoritmia y Estructura de Datos**

**Pregunta 1:**

#include<iostream>

#include<fstream>

#include<string.h>

typedef char cadena[40];

struct Nodo{

int codigo;

cadena apellidos;

int edad;

Nodo \*puntero1;

};

using namespace std;

void Buscar(Nodo \*nodo1,Nodo \*&ant,Nodo \*&act,bool &Enlista){

bool existe=false;

Enlista=false;

ant=NULL;

act=nodo1;

while(act!=NULL && !existe){

if(act->edad>20){

existe=true;

Enlista=true;

}

else{

ant=act;

act=act->puntero1;

}

}

}

void Eliminar(Nodo \*&nodo1){

Nodo \*ant,\*act;

bool Enlista;

Buscar(nodo1,ant,act,Enlista);

if(Enlista){

if(nodo1==act){

nodo1=nodo1->puntero1;

delete act;

Eliminar(nodo1);

}

else{

ant->puntero1=act->puntero1;

delete act;

Eliminar(nodo1);

}

}

}

void Crea\_Nodo(Nodo \*&nodo1,int codigo,cadena apellidos,int edad){

if(nodo1==NULL){

nodo1=new Nodo();

nodo1->codigo=codigo;

strcpy(nodo1->apellidos,apellidos);

nodo1->edad=edad;

nodo1->puntero1;

}

else{

Crea\_Nodo(nodo1->puntero1,codigo,apellidos,edad);

}

}

void Ordenar(Nodo \*nodo1){

Nodo \*nodo2;

cadena aux1;

int aux2,aux3;

nodo2=nodo1;

while(nodo1!=NULL){

nodo2=nodo1->puntero1;

while(nodo2!=NULL){

if(strcmp(nodo1->apellidos,nodo2->apellidos)>0){

strcpy(aux1,nodo1->apellidos);aux2=nodo1->codigo;aux3=nodo1->edad;

strcpy(nodo1->apellidos,nodo2->apellidos);nodo1->codigo=nodo2->codigo;nodo1->edad=nodo2->edad;

strcpy(nodo2->apellidos,aux1);nodo2->codigo=aux2;nodo2->edad=aux3;

}

nodo2=nodo2->puntero1;

}

nodo1=nodo1->puntero1;

}

}

void Escritura\_Eliminados(Nodo \*nodo1){

ofstream archivo2;

archivo2.open("ELIMINADOS.txt");

if(!archivo2){

cout<<"No se pudo abrir el archivo..."<<endl;

}

else{

while(nodo1!=NULL){

if(nodo1->edad>20){

archivo2<<nodo1->codigo<<"\t"<<nodo1->apellidos<<"\t"<<nodo1->edad<<endl;

}

nodo1=nodo1->puntero1;

}

archivo2.close();

}

}

void Recorrer(Nodo \*nodo1){

if(nodo1!=NULL){

cout<<nodo1->codigo<<"\t"<<nodo1->apellidos<<"\t"<<nodo1->edad<<endl;

Recorrer(nodo1->puntero1);

}

}

int main(){

ifstream archivo;

int codigo,edad;

cadena apellidos,apellido\_p,apellido\_m;

Nodo \*Lista=NULL;

archivo.open("ALUMNOS.TXT");

if(!archivo){

cout<<"No se pudo abrir correctamente el archivo...";

}

else{

while(!archivo.eof()){

if(archivo>>codigo>>apellido\_p>>apellido\_m>>edad){

strcpy(apellidos,apellido\_p);

strcat(apellidos," ");

strcat(apellidos,apellido\_m);

Crea\_Nodo(Lista,codigo,apellidos,edad);

}

}

archivo.close();

}

Ordenar(Lista);

cout<<"Lista creada y ordenada:"<<endl;

Recorrer(Lista);

Escritura\_Eliminados(Lista);

Eliminar(Lista);

cout<<"\nLista de alumnos no eliminados:"<<endl;

Recorrer(Lista);

return 0;

}

**Pregunta 2:**

#include<iostream>

#include<fstream>

#include<string.h>

using namespace std;

typedef char str[21];

struct nodo {

str pal;

int n;

nodo \*d;

nodo \*i;

};

void crea (str pal,nodo \*&p) {

if (p==NULL) {

p=new(nodo);

strcpy(p->pal,pal);

p->n=1;

p->d=NULL;

p->i=NULL;

}

else {

if (strcmpi(pal,p->pal)>0) {

crea(pal,p->d);

}

else {

if (strcmpi(pal,p->pal)==0) {

p->n++;

}

else {

crea(pal,p->i);

}

}

}

}

void imprimir (nodo \*p) {

if (p!=NULL) {

imprimir(p->i);

cout<<p->pal<<"\t"<<p->n<<endl;

imprimir(p->d);

}

}

int main () {

ifstream ent;

str pal;

nodo \*p;p=NULL;

ent.open("FRASES.TXT");

if (!ent) {

cout<<"Error de apertura de FRASES.TXT"<<endl;

}

else {

while (!ent.eof()) {

if (ent>>pal) {

crea(pal,p);

}

}

ent.close();

}

imprimir(p);

return 0;

}

**Pregunta 3:**

#include <iostream>

using namespace std;

bool busqueda\_binaria(int inicio,int fin,int dato,int matriz[],int n,int &pos){

int medio = (inicio+fin)/2;

if (dato==matriz[medio]){

pos=medio;

return true;

}

else{

if(dato < matriz[medio]){

if (inicio<=fin){

busqueda\_binaria(inicio,medio-1,dato,matriz,n,pos);

}

else{

return false;

}

}

else {

if (inicio<=fin){

busqueda\_binaria(medio+1,fin,dato,matriz,n,pos);

}

else{

pos=-1;

return false;

}

}

}

}

int main(){

int n,dato,pos=0;

//Se debe digitar en forma ordenada

do{

cout<<"Digite la cantidad de elementos: ";cin>>n;

}while(!(n>0));

int matriz[n],inicio=0,final=n-1;

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

cout<<"Digite elemento "<<i+1<<": ";cin>>matriz[i];

}

cout<<"Digite el dato que desea buscar: ";cin>>dato;

cout<<"[ ";

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

cout<<matriz[i]<<" ";

}

cout<<"]"<<endl;

if (busqueda\_binaria(inicio,final,dato,matriz,n,pos)){

cout<<dato<<" se encuentra en la posicion "<<pos;

}

else{

cout<<dato<<" se encuentra en la posicion "<<pos;

}

}