Logotipo, nombre de la empresa

Descripción generada automáticamenteMétodo de búsqueda binaria

Nombre: Oscar Alejandro Penilla Skakievich

Fecha: 08/09/22

Grupo:5D1

Registro: 20300701

**Descripción del programa:**

Se va a crear un programa que sirva para almacenar varios datos sobre estudiantes, este tendrá la opción de agregar hasta un máximo de 50 estudiantes y después de esos 50, ya no dejará introducir mas, además, este podrá buscar los valores y detectar si alguno se llegara a repetir entre estos mismos, y este haciendo además del ordenamiento antes de buscarlo, va a filtrar los datos para de una manera mas rápida hacer la busqueda

**Requerimientos:**

a) El programa permitirá la inserción y búsqueda de datos.

b) Se podrán ingresar de 1 –50 valores.

c)Emplee cualquier método de ordenación.

d)Los datos que se ingresarán son estructuras u objetos, con lo siguientes campos o propiedades.

•Nombre (Cadena de 50 caracteres).

•Apellido(Cadena de 50 caracteres).

•Registro (Entero positivo)

•Edad (Entero positivo)

e)La búsqueda se puede realizar con dos campos por ejemplo nombre y registro.

f)Una vez localizado un valor se mostrará toda la información localizada.

g)El programa mostrará todas las coincidencias.

**PSP:**

|  |  |  |  |
| --- | --- | --- | --- |
| Contenido del PSP – Busqueda binaria | | | |
| Código fuente: | Estimado | Real | Error |
| (min) | (min) |
| Tiempo de Diseño | 60 | 30 | 50% |
| Tiempo de Codificación. | 120 | 210 | 75% |
| Tiempo estimado de fabricación | 130 | 240 | 85% |
| Total, de líneas de código nuevas. | 40 | 60 | 50% |
| Total, de líneas de código reusadas. | 120 | 263 | 119% |
| Total, de líneas de código del programa. | 160 | 323 | 102% |
| Cantidad de errores de compilación. | 80 | 90 | 13% |
| Cantidad de errores de ejecución. | 90 | 98 | 9% |

**Código fuente:**

#include <iostream>

#include <cctype>

#include <string.h>

#include <cmath>

using namespace std;

#define HashMod 50

//errores 120

struct persona {

int registro;

char nombre[50] = "null";

char apellido[50];

int edad;

};

unsigned long long CharToInt(char entrada)

{

unsigned long long Salida;

Salida = entrada;

return Salida;

}

unsigned long long ObtenerNumero(string entrada)

{

unsigned long long Salida=0;

for(int i=0;entrada[i] != '\0';i++)

{

if(entrada[1+i] == '\0' && i==0)

Salida+=CharToInt(entrada[i]);

else

{

if(i==0)

{

Salida += CharToInt(entrada[i]);

continue;

}

else

{

int miChar = CharToInt(entrada[i]);

if(miChar>99)

{

Salida\*=1000;

}

else if(miChar>9)

{

Salida\*=100;

}

Salida += miChar;

}

}

}

return Salida;

}

int FuncHash(string entrada)

{

unsigned long long Amodular = ObtenerNumero(entrada);

int Salida = Amodular%HashMod;

return Salida;

}

void busqueda(persona\* valor, char\* bus\_nomb, int max){

if(max==0){

cout<< "No se ha ingresado ningun valor"<<endl;

}else

{

int valorrep=0, posencon[50];

for (int i = 0; i < max; i++)

{

// cout <<"Valor "<<valor[i].nombre<<endl;

// cout <<"Busqueda "<<bus\_nomb<<endl;

cout <<!strcmp(valor[i].nombre, bus\_nomb)<<endl;

if(!strcmp(valor[i].nombre, bus\_nomb))

valorrep++;

posencon[i]=1;

}

// cout << "Datos encontrados "<<valorrep<<endl;

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

if(posencon[i]==1 && !strcmp(valor[i].nombre, bus\_nomb)) {

cout <<"Registro: "<< valor[i].registro <<endl;

cout <<"Nombre: "<< valor[i].nombre <<endl;

cout <<"Apellido: "<< valor[i].apellido <<endl;

cout <<"Edad: "<< valor[i].edad <<endl;

}

cout<<endl;

}

}

}

void shell(persona\* arreglo, int cantidad) {

int i, a=1, aux;

bool b\_door;

int i\_redond = cantidad;

while (i\_redond > 1)//comenzamos a ordenar

{

i\_redond = ceil(float(i\_redond)/2);

b\_door = true;

while (b\_door == true)

{

b\_door = false;

i = 0;

while ((i + i\_redond) <= cantidad)

{

if (arreglo[i].registro > arreglo[i + i\_redond]. registro)

{

aux = arreglo[i].registro;

arreglo[i].registro = arreglo[i + i\_redond].registro;

arreglo[i + i\_redond].registro = aux;

b\_door = true;

}

i++;

}//aqui no se muestra

}//aqui se puede mostrar

}

}

void busqueda(persona\* valor, int bus\_regis, int max){//sobrecarga de funcion para registro

//ya jala no le muevas paro

if(max==0){

cout<< "No se ha ingresado ningun valor"<<endl;

}else{

int valorrep=0, posencon[50];

for (int i = 0; i < max; i++)

{

if(valor[i].registro == bus\_regis) {

valorrep++;

// cout<<"Se encontro similitud\n";

posencon[i]=1;

}

}

cout << "Datos encontrados "<<valorrep<<"\n"<<endl;

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

if(posencon[i]==1 &&valorrep>=1) {

cout <<"Registro: "<< valor[i].registro <<endl;

cout <<"Nombre: "<< valor[i].nombre <<endl;

cout <<"Apellido: "<< valor[i].apellido <<endl;

cout <<"Edad: "<< valor[i].edad <<endl;

}

cout<<endl;

}

}

}

void insercion(persona\* arreglo, int cantidad, int buscregis) {

int i\_auxreg = 0, pos=0;

int i\_auxeda =0;

char c\_auxnom[50];

char c\_auxape[50];

for (int i = 0; i <= cantidad-1; i++)

{

pos = i;//va a ser la posicion

i\_auxreg = arreglo[i].registro;//una variable auxiliar depende de i

i\_auxeda = arreglo[i].edad;

strcpy(c\_auxnom, arreglo[i].apellido);

strcpy(c\_auxape, arreglo[i].apellido);

while (pos>0 && (arreglo[pos-1].registro>i\_auxreg)){//mientras posicion sea mayor a 0 y el numero del arreglo sea mayor que aux

arreglo[pos].registro=arreglo[pos-1].registro;

arreglo[pos].edad=arreglo[pos-1].edad;

strcpy(arreglo[pos].apellido, arreglo[pos-1].apellido);

strcpy(arreglo[pos].nombre, arreglo[pos-1].nombre);

pos--;//reducimos la posicion para el siguiente ciclo

}

arreglo[pos].registro=i\_auxreg;//se hace el intercambio de variable

arreglo[pos].edad=i\_auxeda;

strcpy(arreglo[pos].nombre, c\_auxnom);

strcpy(arreglo[pos].apellido, c\_auxape);

// cout << "Iteracion " << i << ": ";//mostramos

for (int j = 0; j < cantidad; j++)

{//"Sin importar cual caso sea se deberá mostrar el arreglo en cada inserción."

}

cout << "\n";

}

busqueda(arreglo,buscregis,cantidad);

}

int main() {

persona estudiantes[50];//pasar el numero de usuarios con sus variables correspondientes

int i\_cont\_rep =0;//almacenar el valor de nuestras variables

int i\_cont\_rep2 =0/\*, verificador=0\*/;//busqueda char

bool b\_door=true;//para repetir nuestro codigo

//variables para la busqueda

char ingreso = 'S', buscar='S';

char buscnomb[50]=" ";

int buscregis = 0;

cout << "Programa de busqueda binaria"<<endl;

do{

do

{

buscar = 'S';

cout << "A- Ingresar datos. \nB- Buscar datos.\nC- Salir" << endl;

cin >> ingreso;

if(ingreso != 'A' && ingreso != 'B' && ingreso !='C'){

cout<<"Valor no corresponde\n";

}

switch (ingreso)

{

case 'A':

{

if(i\_cont\_rep>=50){

cout << "Se llego a la cantidad maxima de usuarios"<<endl;

break;//para salir si hemos llegado al tope de usuarios

}else{

do

{

cout << "Ingresa su registro: ";

cin >> estudiantes[i\_cont\_rep].registro;

// cout << estudiantes[i\_cont\_rep].registro <<endl;

} while (estudiantes[i\_cont\_rep].registro < 1);

cout << "Ingresa su nombre: ";

cin >> estudiantes[i\_cont\_rep].nombre;//ya sabemos que van a ser char

cout << "Ingresa su apellido: ";

cin >> estudiantes[i\_cont\_rep].apellido;

do

{

cout << "Ingresa su edad: ";

cin >> estudiantes[i\_cont\_rep].edad;

// cout << estudiantes[i\_cont\_rep].edad <<endl;

} while (estudiantes[i\_cont\_rep].edad < 1);

// cout << i\_cont\_rep<<endl;//ver en que usuario estamos

i\_cont\_rep++;//a la hora de que termine esto la posicion del arreglo aumentara y asi la siguiente vuelta sera otras nuevas variables, que vamos a recorrer en el caso B

}

}

break;

case 'B':

{

int verificador=0;

char Clave[50];

// char Clave[50]=" ";

while(buscar=='S'){

cout << "Quieres buscar por el nombre?(S/N) ";

cin >> buscar;

if(buscar == 'S'){

cout<<"IMPORTANTE: el nombre debe de ser exacto al ingresado "<<endl;

cout<<"Ingresa el nombre que buscas: "<<endl;

cin>> buscnomb;//no la usamos wtf?

//aqui ordenamos

int Pos = FuncHash(estudiantes[i\_cont\_rep].nombre);

// cout<<"\nValor hash: "<<Pos<<endl;

if(estudiantes[Pos].nombre=="null" || estudiantes[Pos].nombre==Clave){

//estudiantes[Pos].nombre = Clave;

strcpy(estudiantes[Pos].nombre, Clave);

}

else{

for(int i=Pos; i<HashMod; i++){

if(estudiantes[i].nombre=="null")

{

strcpy(estudiantes[i].nombre, Clave);

//estudiantes[i].nombre = Clave;

break;

}

}

}

//aqui buscamos el valor ya ordenado

for (int i = 0; i < HashMod; i++)

{

if( !strcmp(estudiantes[i].nombre, buscnomb) ){

cout <<"Registro: "<< estudiantes[i].registro <<endl;

cout <<"Nombre: "<< estudiantes[i].nombre <<endl;

cout <<"Apellido: "<< estudiantes[i].apellido <<endl;

cout <<"Edad: "<< estudiantes[i].edad<<"\n" <<endl;

verificador++;

}

}

if(verificador==0){

cout<< "No se ha encontrado el nombre"<<endl;

}else if(verificador>1){

cout<< "Hay ("<<verificador<<") nombres iguales\n" <<endl;

}

i\_cont\_rep2++;

break;//para salir al terminar

}else if(buscar == 'N'){

// cout<<"ola, No"<<endl;

break;//solo sale y lo pasa

}else{

cout<<"Ingresa un valor correspondiente"<<endl;

buscar = 'S';//para volver a repetir

}

};//while nombre

while(buscar=='N'){//aqui es N pq buscar sera N si se cancela nombre

cout << "Quieres buscar por el registro?(S/N) ";

cin >> buscar;

if(buscar == 'S'){

cout<<"Ingresa el registro que buscas: "<<endl;

cin>> buscregis;

insercion(estudiantes, i\_cont\_rep, buscregis);//dentro de este esta busqueda

break;//para salir al terminar

}else if(buscar == 'N'){

// cout<<"ola, No"<<endl;

break;//solo sale y lo pasa

}else{

cout<<"Ingresa un valor correspondiente"<<endl;

buscar = 'N';//para volver a repetir

}

}

}

break;

case 'C':

{

cout << "Decidió salir"<<endl;

b\_door = false;

break;

}

}

} while (ingreso != 'A' && ingreso != 'B' && ingreso !='C');

}while(b\_door == true);

}