Logotipo, nombre de la empresa

Descripción generada automáticamenteMétodo de búsqueda secuencial

Nombre: Oscar Alejandro Penilla Skakievich

Fecha: 04/09/22

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

**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 – búsqueda secuencial | | | |
| Código fuente: | Estimado | Real | Error |
| (min) | (min) |
| Tiempo de Diseño | 10 | 40 | 300% |
| Tiempo de Codificación. | 60 | 140 | 133% |
| Tiempo estimado de fabricación | 90 | 180 | 100% |
| Total, de líneas de código nuevas. | 80 | 160 | 100% |
| Total, de líneas de código reusadas. | 100 | 14 | 86% |
| Total, de líneas de código del programa. | 130 | 174 | 34% |
| Cantidad de errores de compilación. | 20 | 47 | 135% |
| Cantidad de errores de ejecución. | 15 | 62 | 313% |

**Código fuente:**

#include <iostream>

#include <cctype>

#include <string.h>

using namespace std;

struct persona {

int registro;

char nombre[50];

char apellido[50];

int edad;

};

void busqueda(persona\* valor, char\* bus\_nomb, int max){//sobrecarga de funcion para nombre

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 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) {

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;

}

}

}

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 cant = 0;//borrar

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 secuencial"<<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':

{

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;

busqueda(estudiantes, buscnomb, i\_cont\_rep);

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;

busqueda(estudiantes, buscregis, i\_cont\_rep);

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);

}