Logotipo, nombre de la empresa

Descripción generada automáticamenteMétodo tablas hash

Nombre: Oscar Alejandro Penilla Skakievich

Fecha: 06/09/22

Grupo:5D1

Registro: 20300701

**Descripción del programa:**

Se va a crear un método de ordenamiento y búsqueda de datos llamado tablas hash, en el cual podrá tener 1 hasta 50 valores con 113 elementos, este método utiliza unas variables llamadas “llaves”, que estas nos ayudan a encontrar el valor que buscamos y así ubicarlo en nuestro arreglo de estructuras (En este caso eso se va a usar)

**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)

d)La tabla se creará empleando el registro como llave.

e)La tabla hash contendrá113 elementos.

f)En caso de una colisión los datos se apilarán(pueden estar 2 o más enla misma posición).

g)En el caso de búsqueda: se ingresará un registro y este será buscado en caso de ser encontrado mostrará el nombre, apellido, registro y edad, y en caso de no localizarse se enviará un mensaje de elemento no encontrado.

**PSP:**

|  |  |  |  |
| --- | --- | --- | --- |
| Contenido del PSP – Tabla hash | | | |
| Código fuente: | Estimado | Real | Error |
| (min) | (min) |
| Tiempo de Diseño | 60 | 30 | 50% |
| Tiempo de Codificación. | 60 | 60 | 0% |
| Tiempo estimado de fabricación | 120 | 90 | 25% |
| Total, de líneas de código nuevas. | 50 | 76 | 52% |
| Total, de líneas de código reusadas. | 70 | 110 | 57% |
| Total, de líneas de código del programa. | 150 | 186 | 24% |
| Cantidad de errores de compilación. | 50 | 60 | 20% |
| Cantidad de errores de ejecución. | 30 | 100 | 233% |

**Código fuente:**

#include <iostream>

#include <cctype>

#include <string.h>

#define HashMod 50

using namespace std;

struct persona {

string registro;

char nombre[50];

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;

}

}

}

int main() {

persona estudiantes[113];//pasar el numero de usuarios con sus variables correspondientes

//string Tabla[50];

for(int i = 0; i<113; i++)

{

estudiantes[i].registro ="null";

}

int aux=0, verificador=0;

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

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

//variables para la busqueda

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

//char buscnomb[50]=" ";

string buscregis=" ";

cout << "Programa de Tabla hash"<<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':

{

string Clave;

do

{

cout << "Ingresa su registro: ";

cin >> aux;

estudiantes[i\_cont\_rep].registro = to\_string(aux);

} while (aux < 1);

cout << "Ingresa su nombre: ";

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

cout << "Ingresa su apellido: ";

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

do

{

cout << "Ingresa su edad: ";

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

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

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

// cout<<"\nEl valor ascii es: "<<ObtenerNumero(Clave);

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

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

estudiantes[Pos].registro=Clave;

}

else{

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

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

{

estudiantes[i].registro=Clave;

break;

}

}

}

}

i\_cont\_rep++;

break;

case 'B':

{

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

cin>> buscregis;

for(int i = 0; i<50; i++)

{

}

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

{

if(buscregis==estudiantes[i].registro){

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

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

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

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

verificador++;

}

}

if(verificador==0){

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

}

}

break;

case 'C':

{

cout << "Decidió salir"<<endl;

b\_door = false;

break;

}

}

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

}while( b\_door == true);

}