AIM:- a. Create a database using array of structures and perform following operations on it: i.Add record

Assignment 3

ii.Display Database

iii.Search record (binary search)

b.For database implemented using array, perform

iv.Modify record

v.Delete record

vi.Sort records(Bubble sort)

OBJECTIVE:-

To create database with functions such as add record, display record, search record, modify record, delete record, sort record using array of structures.

Theory :-

## Arrays of Structures

Since an array can contain similar elements, the combination having structures within an array is an array of structures. To declare an array of structures, you must first define a structure and then declare an array variable of that type. For example, to store addresses of 100 members of the council, you need to create an array.

Now, to declare a 100-element array of structures of type addr (defined in previous chapters), we will write :

addr mem\_addr [100];

This creates 100 sets of variables that are organised as defined in the structure addr. To access a specific structure, index the structure name. For instance, to print the houseno of structure 8, write :

cout << mem\_addr[7].houseno ;

Algorithm :- Step 1 Create structure of student having variables as roll no. and name of the student.

.

Step 2 Get the input of the data of the student using objects created.

Step 3 Display the input data of the student.

Step 4 Using the switch statement display the functions insert ,delete ,Display,search and modify.

Step 5 input the choice of the user using the switch statement and respectively call the functions.

Step 6 for choice ==4 implement binary search for searching the element.

SOURCE CODE:

/\*

PROGRAM BY ANOM DEVGUN

GR: 21810017

Implemented a student database using array of structures.

Implemented various operations through switch case.

\*/

#include<iostream>

#include<string.h>

using namespace std;

int nrec;

struct stu

{

int gno;

string fna,lna;

string addr;

};

int main()

{

int rn=0;

struct stu s[100];

int l;

int la=1;

string fn,ln;

string ad;

int i=0,sr,fl=0,j;

int ch;

cout<<"Enter the number of records you want initially\n";

cin>>nrec;

int lb,ub,mid,hl,pa,sw=1;

cout<<"Enter details for each student\n";

for(i=0;i<nrec;i++)

{

cout<<"Enter GR number\n";

cin>>s[i].gno;

cout<<"\nEnter First name of student\n";

cin>>s[i].fna;

cout<<"\nEnter Last name of student\n";

cin>>s[i].lna;

cout<<"\nEnter Address\n";

cin.ignore();

getline(cin,s[i].addr);

}

while(1)

{

cout<<"1)To BubbleSort database\n";

cout<<"2)To display database\n";

cout<<"3)To Binary search record from database\n";

cout<<"4)To delete record from database\n";

cout<<"5)To modify record in database\n";

cout<<"6)To add record to database\n";

cout<<"7)To exit\n";

cin>>ch;

switch(ch)

{

case 1:

for(pa=0;pa<nrec-1&& sw==1;pa++) //BUBBLESORT ARRAY

{

sw==0;

for(j=0;j<nrec-pa-1;j++)

{

if(s[j].gno > s[j+1].gno)

{

sw=1;

hl=s[j].gno;

s[j].gno=s[j+1].gno;

s[j+1].gno=hl;

fn=s[j].fna;

s[j].fna=s[j+1].fna;

s[j+1].fna=fn;

ln=s[j].lna;

s[j].lna=s[j+1].lna;

s[j+1].lna=ln;

ad=s[j].addr;

s[j].addr=s[j+1].addr;

s[j+1].addr=ad;

}

}

}

break;

case 2: //DISPLAY DATABASE

for(i=0;i<nrec;i++)

{

cout<<"GR Number is :"<<s[i].gno<<"\n";

cout<<"Name is "<<s[i].fna<<" "<<s[i].lna<<"\n";

cout<<"Address is : "<<s[i].addr<<"\n";

}

break;

case 3: //BINARY SEARCH

lb=0;

ub=nrec-1;

mid=(lb+ub)/2;

cout<<"Enter Record to be searched by GR number:\n";

cin>>sr;

while(lb <= ub)

{

if(sr == s[mid].gno)

{

cout<<"The record "<<s[mid].gno<<" has been found : \n";

cout<<"Name is "<<s[mid].fna<<" "<<s[mid].lna<<"\n";

cout<<"Address is : "<<s[mid].addr<<"\n";

break;

}

else if(sr > s[mid].gno)

{

lb=mid+1;

}

else

{

ub=mid-1;

}

mid=(lb+ub)/2;

}

if(lb>ub)

{

cout<<"The Record to be searched in the database was not found.\n";

}

break;

case 4 : //TO DELETE RECORD

cout<<"Enter The record you want deleted from database:\n";

cin>>sr;

for(i=0;i<nrec;i++)

{

if(s[i].gno == sr)

{

la=0;

for(j=i;j<nrec;j++)

{

s[j].gno=s[j+1].gno;

s[j].fna=s[j+1].fna;

s[j].lna=s[j+1].lna;

s[j].addr=s[j+1].addr;

}

nrec--;

cout<<"Record deleted.\n";

break;

}

}

if(la == 1)

cout<<"Record not found.\n";

break;

case 5: //TO MODIFY RECORD

cout<<"Input record you want to modify: \n";

cin>>sr;

for(i=0;i<nrec;i++)

{

if(s[i].gno == sr)

{

cout<<"Enter First name and last name: \n";

cin>>s[i].fna>>s[i].lna;

cout<<"Enter Address: \n";

cin.ignore();

getline(cin,s[i].addr);

cout<<"\nRecord updated successfully\n";

la=0;

}

}

if(la == 1)\

cout<<"Record not found.\n";

break;

case 6:

cout<<"Enter Number of Records you want to add to database:\n";

cin>>rn;

for(i=nrec;i<nrec+rn;i++)

{

cout<<"Enter GR number\n";

cin>>s[i].gno;

cout<<"\nEnter First name of student\n";

cin>>s[i].fna;

cout<<"\nEnter Last name of student\n";

cin>>s[i].lna;

cout<<"\nEnter Address\n";

cin.ignore();

getline(cin,s[i].addr);

}

nrec+=rn;

break;

case 7: cout<<"\nNow Exiting.\n";

exit(0);

break;

default:

cout<<"Invalid choice.\n";

}

}

return 0;

}

