**Name : Resham Landge**

**Roll No : 2339**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***Assignment No : 11**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Title** **:** Department maintains a student information. The file contains roll number, name, division and address. Write a program to create a sequential file to store and maintain student data. It should allow the user to add, delete information of student. Display information of particular employee. If record of student does not exist an appropriate message is displayed. If student record is found it should display the student details.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include<iostream>

#include<fstream>

#include<stdlib.h>

#include<string.h>

using namespace std;

class file //class for file

{

public:

int roll;

float age;

char name[100];

void input();

void show();

char \*getn()

{

return name;

}

};

file fileobj;

fstream fil;

void file::input() //function to take input

{

cout<<"\n\tEnter the Roll, Age and Name : ";

cin>>roll>>age>>name;

}

void file::show() //function to show records in the file

{

cout<<"\n\tRoll==> "<<roll;

cout<<"\tAge ==> "<<age;

cout<<"\tName==> "<<name;

}

void Create() //Function to Create Data File

{

char ch='y';

fil.open("binary.dat",ios::out| ios::binary); //to open the file in write mode to file

while(ch=='y' || ch =='Y')

{

fileobj.input(); //insert the records in the file

fil.write((char\*)&fileobj, sizeof(fileobj));

cout<<"\n\tDo U Want to Continue(y/n) : ";

cin>>ch;

}

fil.close(); //to close existing file

}

void Add() //Function to Add New Record in Data File

{

char ch='y';

fil.open("binary.dat",ios::out|ios::app| ios::binary);//open file in append or write mode

while(ch=='y' || ch =='Y')

{

fileobj.input(); //insert the records need to add

fil.write((char\*)&fileobj, sizeof(fileobj));

cout<<"\n\tDo U Want to Continue(y/n) : ";

cin>>ch;

}

fil.close(); //close the exxisting file

}

void Display() //Function to Display All Record from Data File

{

fil.open("binary.dat",ios::in| ios::binary); //open the file in reading mode

if(!fil) //file not found

{

cout<<"\n\tFile not Found "; // no file exsit

exit(0);

}

else

{

fil.read((char\*)&fileobj, sizeof(fileobj));

while(!fil.eof()) // continue until END OF FILE pointer

{

fileobj.show(); //display the records until the END OF FILE

fil.read((char\*)&fileobj, sizeof(fileobj));

}

}

fil.close(); //close the file

}

void DisplayP() //Function to Display particular Record from Data File

{

char n[100];

cout<<"\n\tEnter Name that should be searched:";

cin>>n;

fil.open("binary.dat",ios::in| ios::binary);//open the file in reading mode

if(!fil)

{

cout<<"\n\tFile not Found ";

exit(0);

}

else

{

fil.read((char\*)&fileobj, sizeof(fileobj));//read thhe data from file

while(!fil.eof()) //continue the file until the END OF FILE

{

if(strcmp(n,fileobj.getn())==0)//search for the records in the file

{

fileobj.show(); //if the name enterd and searched record name is same display

fil.close(); // close the file

return;

}

fil.read((char\*)&fileobj, sizeof(fileobj));

}

cout<<"\n\tRecord not found\n";

}

fil.close(); //close the file

}

void Modify() //Function to Modify Particular Record from Data File

{

char n[100];

cout<<"\n\tEnter Name that should be searched:";

cin>>n;

fil.open("binary.dat",ios::in| ios::out|ios::binary);//open the file in //reading and writing mode

if(!fil) //if file is not found then no file exist

{

cout<<"\n\tFile not Found";

exit(0);

}

else

{

fil.read((char\*)&fileobj, sizeof(fileobj));//read records form file

while(!fil.eof()) //continue until END OF FILE

{

if(strcmp(n,fileobj.getn())==0)//search the records by name

{

fil.seekg(0,ios::cur);//jump the file pointer to record

cout<<"\n\tEnter New Record..";

fileobj.input();//insert the new records

int p=fil.tellg();

int q=sizeof(fileobj);

fil.seekp(p-q);

fil.write((char\*)&fileobj, sizeof(fileobj));

fil.close(); // close the file

return;

}

fil.read((char\*)&fileobj, sizeof(fileobj));//read records from file

}

cout<<"\n\tRecord not found\n";

}

fil.close();

}

void Delete() //Function to Delete Particular Record from Data File

{

int flag=0;

char n[100];

cout<<"\n\tEnter Name that should be Deleted :";

cin>>n;

ofstream o;

o.open("new.dat",ios::out|ios::binary);//open file in write mode

fil.open("binary.dat",ios::in| ios::binary);//open file reading mode

if(!fil) //find the file

{

cout<<"\n\tFile not Found";

exit(0);

}

else

{

fil.read((char\*)&fileobj, sizeof(fileobj));//read the records from the file

while(!fil.eof()) // continue until END OF FILE

{

if(strcmp(n,fileobj.getn())!=0) //search the records by using name

o.write((char\*)&fileobj, sizeof(fileobj));//open write mode

else

flag=1;

fil.read((char\*)&fileobj, sizeof(fileobj));//read the ecords

}

if(flag==0) // if flag=0 then no records exist

cout<<"Record not found\n";

}

o.close(); //close the file object

fil.close(); //close the file

remove("binary.dat");

rename("new.dat", "binary.dat");

}

main()

{

int opt;

while(1)

{

cout<<"\n\t1.Create Data File";

cout<<"\n\t2.Add New Record in Data File";

cout<<"\n\t3.Display Record From Data File";

cout<<"\n\t4.Display Particular Record From Data File";

cout<<"\n\t5.Modify Paricular Record From Data File";

cout<<"\n\t6.Delete Particular Record From Data File";

cout<<"\n\t7.Exit From the Program";

cout<<"\n\n\tEnter your choice : ";

cin>>opt;

switch(opt)

{

case 1:

Create();

cout<<"\n\tDisplay Main Menu\n";

break;

case 2:

Add();

cout<<"\n\tDisplay Main Menu\n";

break;

case 3:

Display();

cout<<"\n\tDisplay Main Menu\n";

break;

case 4:

DisplayP();

cout<<"\n\tDisplay Main Menu\n";

break;

case 5:

Modify();

cout<<"\n\tDisplay Main Menu\n";

break;

case 6:

Delete();

cout<<"\n\tDisplay Main Menu\n";

break;

case 7:

exit(0);

default:

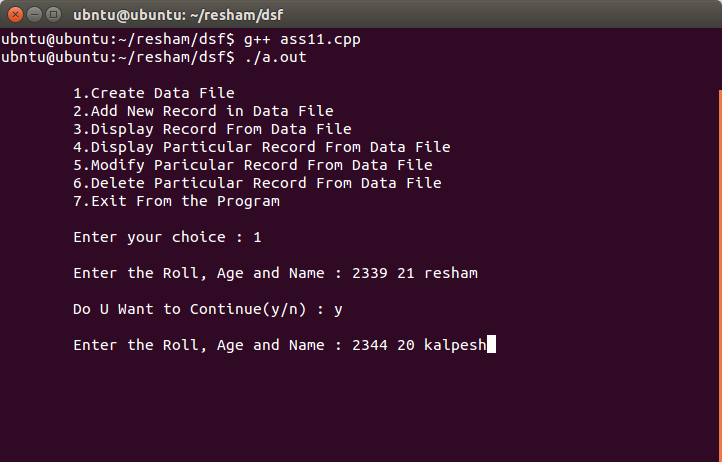
cout<<"\n\tWrong Choice....Press Key For View the Main Menu";

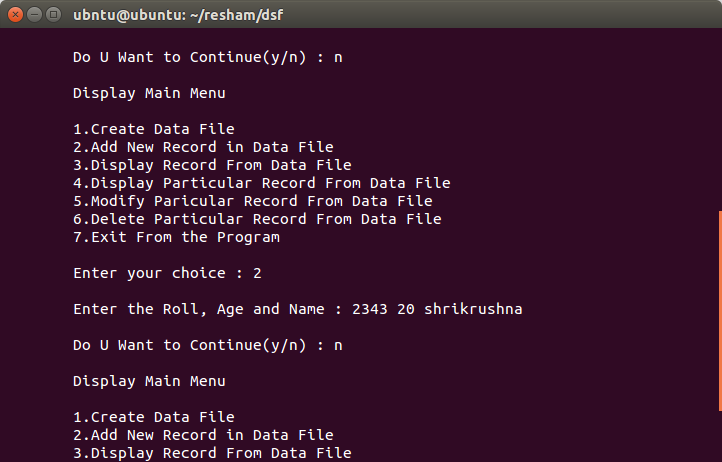
}

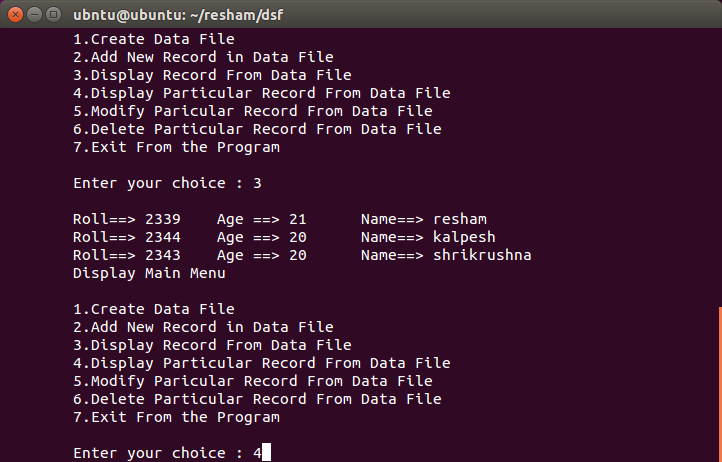
}

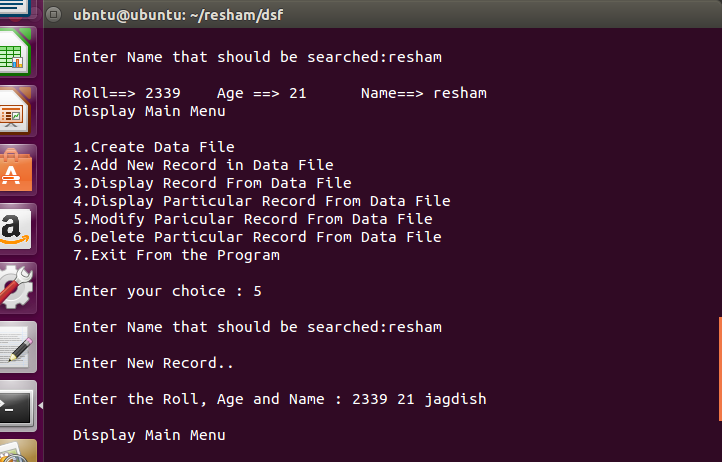
}

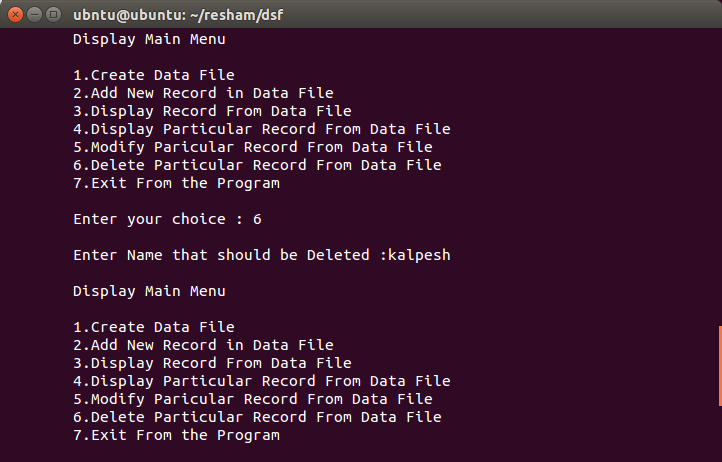
**Output :**

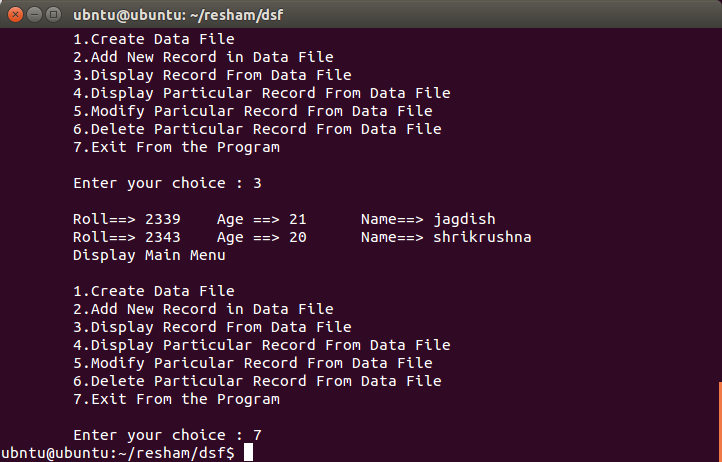
****

****

****

****

****

****