**Name : Resham Landge**

**Roll No : 2339**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***Assignment No : 12**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Title** **:** Implement direct access file using hashing ( chaining without replacement) perform following operations on it

a. Create Database

b. Display Database

c. Add a record

d. Search a record

e. Modify a record

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

#include<iostream>

#include<fstream>

#include<string.h>

#include<stdlib.h>

using namespace std;

class EMP //class declare

{

public:

typedef struct Employee

{

char name[20];

int id,salary;

}Rec;

typedef struct Index

{

int id,position;

}Ind\_Rec;

Rec record;

Ind\_Rec Ind\_record;

void create();

void display();

void update();

void Delete();

};

void EMP::create() //create function call using super class

{

int i=0;

char ch='y';

fstream seqfile,indexfile;

indexfile.open("IND.DAT",ios::in | ios::binary);

seqfile.open("EMP.DAT",ios::out|ios::app | ios::binary);

indexfile.seekg(0,ios::end);

i=indexfile.tellg() / sizeof(Ind\_record);

i++;

indexfile.close();

indexfile.open("IND.DAT",ios::out | ios::app | ios::binary);

do //do while loop call

{

cout<<"\n\tEnter Name : ";

cin>>record.name;

cout<<"\n\tEnter Employee Id : ";

cin>>record.id;

cout<<"\n\tEnter Salary : ";

cin>>record.salary;

seqfile.write((char\*)&record,sizeof(record))<<flush;

Ind\_record.id=record.id;

Ind\_record.position=i;

indexfile.write((char\*)&Ind\_record,sizeof(Ind\_record))<<flush;

i++;

cout<<"\n\tDo U want Add More Records (y/n) : ";

cin>>ch;

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

seqfile.close();

indexfile.close();

}

void EMP::display() //display function call

{

fstream seqfile,indexfile;

int i;

indexfile.open("IND.DAT",ios::in | ios::binary);

seqfile.open("EMP.DAT",ios::in | ios::binary);

if(!indexfile)

{

cout<<"\n\tIndex File not Found : ";

return;

}

cout<<"\n\tThe contents of file are..";

while(indexfile.read((char\*)&Ind\_record,sizeof(Ind\_record)))

{

i=(Ind\_record.position-1)\*sizeof(Rec);

seqfile.seekg(i,ios::beg);

seqfile.read((char\*)&record,sizeof(record));

if(record.id!=-1)

{

cout<<"\n\tName = "<<record.name;

cout<<"\tEMP-ID = "<<record.id;

cout<<"\tSalary = "<<record.salary<<"\n\n";

}

}

seqfile.close();

indexfile.close();

}

void EMP::update() //update function call

{

int pos=-1,id1,new\_salary;

char new\_name[20];

fstream seqfile,indexfile;

indexfile.open("IND.DAT",ios::in | ios::binary);

seqfile.open("EMP.DAT",ios::out | ios::in | ios::binary);

if(!indexfile)

{

cout<<"\n\tIndex File not Found : ";

return;

}

cout<<"\n\tFor updation : \n\tEnter the ID for searching : ";

cin>>id1;

while(indexfile.read((char\*)&Ind\_record,sizeof(Ind\_record)))

{

if(id1==Ind\_record.id)

{

pos=Ind\_record.position;

break;

}

}

if(pos==-1)

{

cout<<"\n\tThe Record Id not present\n";

return;

}

cout<<"\n\tEnter the values for updation : \n\tName = ";

cin>>new\_name;

cout<<"\n\tSalary = ";

cin>>new\_salary;

int offset=(pos-1)\*sizeof(record);

seqfile.seekg(offset);

record.id=id1;

strcpy(record.name,new\_name);

record.salary=new\_salary;

seqfile.write((char\*)&record,sizeof(record))<<flush;

cout<<"\n\tThe rocord is updated\n";

seqfile.close();

indexfile.close();

}

void EMP::Delete() //delete function call

{

int pos=-1,id1;

fstream seqfile,indexfile;

indexfile.open("IND.DAT",ios::in | ios::binary);

seqfile.open("EMP.DAT",ios::in | ios::out | ios::binary);

if(!indexfile)

{

cout<<"\n\tIndex File not Found : ";

return;

}

cout<<"\n\tFor Deletion : \n\tEnter the ID for searcching : ";

cin>>id1;

while(indexfile.read((char\*)&Ind\_record,sizeof(Ind\_record)))

{

if(id1==Ind\_record.id)

{

pos=Ind\_record.position;

Ind\_record.id=-1;

break;

}

}

if(pos==-1)

{

cout<<"\n\tThe Record Id not present\n";

return;

}

int offset=(pos-1)\*sizeof(record);

seqfile.seekg(offset);

strcpy(record.name," ");

record.id=-1;

record.salary=-1;

seqfile.write((char\*)&record,sizeof(record));

offset=(pos-1)\*sizeof(Ind\_record);

indexfile.seekg(offset);

indexfile.write((char\*)&Ind\_record,sizeof(Ind\_record));

seqfile.close();

indexfile.close();

cout<<"\n\tThe rocord is Deleted\n";

}

int main()

{

EMP e;

char ch='y';

int choice;

do

{

cout<<"\n\t1.Create\n\t2.Display\n\t3.Update\n\t4.Delete\n\t5.Exit";

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

cin>>choice;

switch(choice)

{

case 1:

e.create();

break;

case 2:

e.display();

break;

case 3:

e.update();

break;

case 4:

e.Delete();

break;

case 5:

exit(0);

default:

cout<<"\n\t\*\*\*Invalid Choice\*\*\*";

}

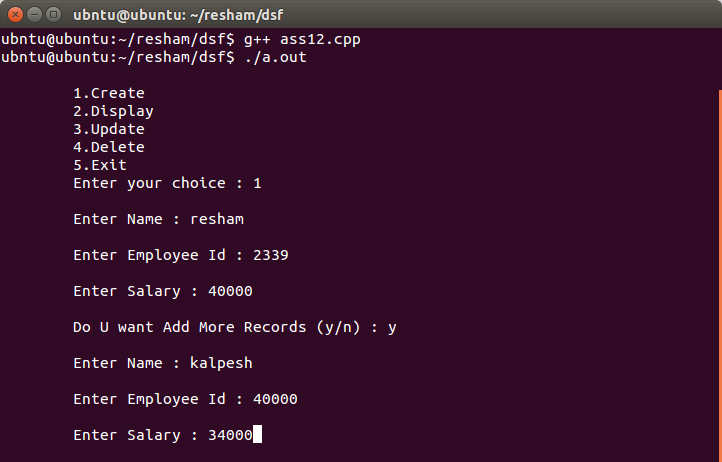
cout<<"\n\tDo U want to Continue(y/n) : ";

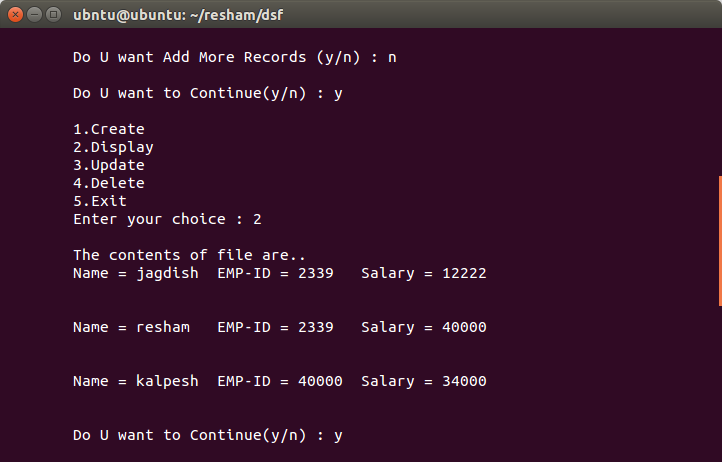
cin>>ch;

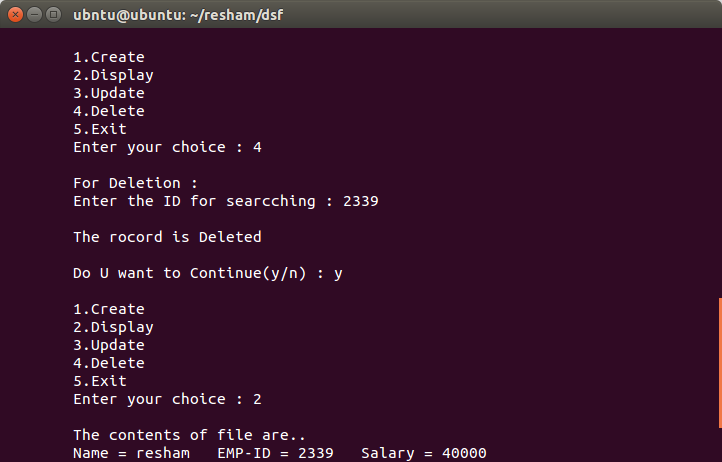
}while(ch=='y'||ch=='Y');

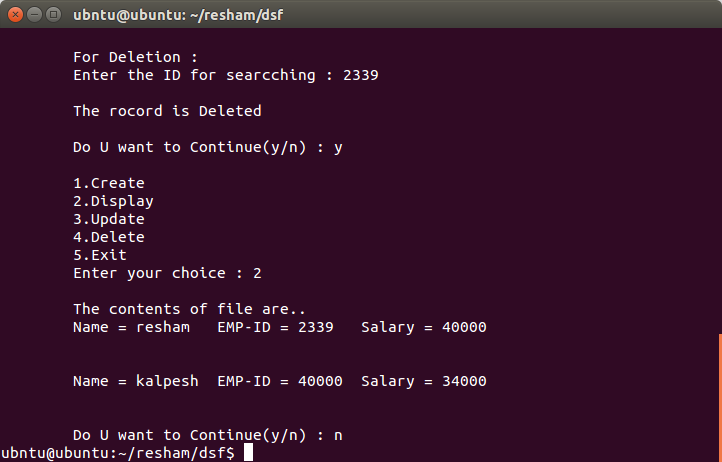
}

**Output :**

****

****

****

****