/\*Program for implementing Direct Access File using the hashing technique.

Collision handling is performed using linear probing and

chaining without replacement.

[hash function=(record\_id mod 10)]\*/

#include<iostream>

#include<iomanip>

#include<fstream>

#include<string.h>

#include<stdlib.h>

using namespace std;

class EMP\_CLASS

{

typedef struct EMPLOYEE

{

char name[10];

int emp\_id;

int salary;

int link;

int loc;

}Rec;

Rec Records;

public:

int size;

int Chain\_tab[10][10];

EMP\_CLASS();

void Insert();

void init();

void Display();

void Search();

void set\_chain();

friend int Hash(int);

};

EMP\_CLASS::EMP\_CLASS()

{

strcpy(Records.name,"");

Records.emp\_id=-1;

Records.salary=-1;

Records.link=-1;

}

void EMP\_CLASS::init()

{

fstream seqfile;

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

cout<<"\n Enter the Hash table size ";

cin>>size;

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

{

strcpy(Records.name,"");

Records.emp\_id=-1;

Records.salary=-1;

Records.link=-1;

Records.link=i;

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

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

for(int j=0;j<size;j++)

Chain\_tab[i][j]=-1;

}

cout<<"\n\n Hash table is initialised...";

cout<<"\n Now, insert the records in the hash table";

seqfile.close();

}

int Hash(int num)

{

int key;

key=num%10;

return key;

}

void EMP\_CLASS::set\_chain()

{

fstream seqfile;

int i,j,h,offset;

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

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

{

h=i;

for(j=0;j<size;j++)

{

if(Chain\_tab[i][j]==1)

{

offset=h\*sizeof(Records);

seqfile.seekg(offset);

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

seqfile.seekp(offset);

Records.link=j;

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

h=j;

}

}

}

seqfile.close();

}

void EMP\_CLASS::Insert()

{

int i,h;

char ch='y';

char new\_name[10];

int new\_emp\_id;

int new\_salary;

fstream seqfile;

init();//initialising the hash table

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

do

{

cout<<"\n Enter Name: ";

cin>>new\_name;

cout<<"\n Enter Emp\_ID: ";

cin>>new\_emp\_id;

cout<<"\n Enter Salary: ";

cin>>new\_salary;

h=Hash(new\_emp\_id);

int offset;

offset=h\*sizeof(Records);

//seeking for reading record

seqfile.seekg(offset);

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

//seeking for writing record

seqfile.seekp(offset);

if(Records.emp\_id==-1)

{

strcpy(Records.name,new\_name);

Records.emp\_id=new\_emp\_id;

Records.salary=new\_salary;

Records.link=-1;

Records.loc=h;//h is used for marking the loc.

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

//thus rec. is inserted at the hashed position in file

}

else//collision occurs

{

int flag=0;

int prev\_link=Records.loc;

do //handling collision

{

h++;//searching down for empty loc.in the file

if(h>size+1)

{

cout<<"\n The hash table is Full, Can't insert record!!!";

return;

}

offset=h\*sizeof(Records);

seqfile.seekg(offset);

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

if(Records.emp\_id==-1) //finding empty loc. using linear probing

{

seqfile.seekp(offset);//seeking the empty slot in the file

strcpy(Records.name,new\_name);//for placing the record

Records.emp\_id=new\_emp\_id;

Records.salary=new\_salary;

Records.link=-1;

Records.loc=h;//setting the location for colliding record

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

//collinding record is placed in the file at proper pos.

//chain table is maintained for keeping track of all the colliding entries.

Chain\_tab[prev\_link][h]=1;

flag=1;//indicates colliding record is inserted

}//end of if

}while(flag==0);//collision handled

}//end of else

cout<<"\nDo you want to add more records?";

cin>>ch;

set\_chain();//setting the chain to handle collision

}while(ch=='y');

seqfile.close();

}

void EMP\_CLASS::Display()

{

fstream seqfile;

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

seqfile.seekg(0,ios::beg);

cout<<"\n The Contents of file are ..."<<endl;

cout<<"\nLoc. Name Emp\_ID Salary Link ";

while(seqfile.read((char \*)&Records,sizeof(Records)))

{

if(strcmp(Records.name,"")!=0)//not displaying empty slots

{

cout<<"\n--------------------------------------------------------\n";

cout<<Records.loc<<" "<<Records.name<<flush<<" "<<Records.emp\_id;

cout<<" "<<Records.salary<<" "<<Records.link;

}

}

seqfile.close();

}

void EMP\_CLASS::Search()

{

fstream seqfile;

int key,h,offset,flag=0;

cout<<"\n Enter the Emp\_ID for searching the record ";

cin>>key;

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

h=Hash(key);//obtaining the location of rec.using hash function

while(seqfile.eof()==0)

{ //h is a hash key

offset=h\*sizeof(Records);

//using h for getting actual position in the file

//hence offset is calculated

seqfile.seekg(offset,ios::beg);//seeking rec.of that offset

seqfile.read((char \*)&Records,sizeof(Records));//reading that rec.

if(key==Records.emp\_id)//checking if it is required rec.

{

cout<<"\n The Record is present in the file and it is...";

cout<<"\n Name: "<<Records.name;

cout<<"\n Emp\_ID: "<<Records.emp\_id;

cout<<"\n Salary: "<<Records.salary;

flag=1;//means desired reocrd is obtained

return;

}

else//following link for colliding record

{

h=Records.link;//moving along the chain

}

}//endof while

if(flag==0)

cout<<"\n The Record is not present in the file";

seqfile.close();

}

int main()

{

EMP\_CLASS List;

char ans='y';

int choice,key;

do

{

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

cout<<"\n 1.Insert";

cout<<"\n 2.Display";

cout<<"\n 3.Search";

cout<<"\n 4.Exit";

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

cin>>choice;

switch(choice)

{

case 1:List.Insert();

break;

case 2:List.Display();

break;

case 3:List.Search();

break;

case 4:exit(0);

}

cout<<"\n\t Do you want to go back to Main Menu?";

cin>>ans;

}while(ans=='y');

}

/\*output

Main Menu

1.Insert

2.Display

3.Search

4.Exit

Enter your choice: 1

Enter the Hash table size 2

Hash table is initialised...

Now, insert the records in the hash table

Enter Name: abc

Enter Emp\_ID: 08

Enter Salary: 10

Do you want to add more records?y

Enter Name: abc

Enter Emp\_ID: 07

Enter Salary: 00

The hash table is Full, Can't insert record!!!

Do you want to go back to Main Menu?y

Main Menu

1.Insert

2.Display

3.Search

4.Exit

\*/