***Skill development project***

**Program:-**

#include<iostream>

#include<string.h>

#include<fstream>

#define MAX 10

using namespace std;

class Employee

{

public : char name[10],telno[10];

int chain;

public :

void show()

{

cout << "\n "<<"\t"<<name<<"\t"<<telno<<"\t"<<chain;

}

void getemp()

{

cout << "\n Enter the name : ";

cin >> name;

cout << "\n Enter the telno : ";

cin >> telno;

chain = -1;

}

};

class HashFile

{

private : char fname[100];

int n;

public : HashFile()

{

cout << "\n Enter the file name : ";

cin >> fname;

}

void menu();

void initalize();

void displayht();

void searchht(char[]);

void add\_chain\_without\_rep();

int divisionmethod(char[]);

};

void HashFile :: searchht(char sname[])

{

fstream finout;

int key,flag=0;

finout.open(fname,ios::in | ios::binary);

if(finout.fail())

{

cout << "\n Error opening file";

}

else

{

Employee e2;

while(1)

{

key = divisionmethod(sname);

finout.seekg(key\*sizeof(e2),ios::beg);

finout.read((char\*)&e2,sizeof(e2));

if( strcmp(e2.name,sname) == 0 )

{

flag=1;

break;

}

else

{

key = e2.chain;

if(key == -1)

{

flag=0;

break;

}

}

}

if(flag==1)

{

cout << "\n Record found .....";

e2.show();

}

else

{

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

}

}

finout.close();

}

int HashFile :: divisionmethod(char name[])

{

return name[0] % MAX;

}

void HashFile :: add\_chain\_without\_rep()

{

fstream finout;

int key,oldkey;

char ch;

finout.open(fname,ios::in | ios::out | ios::binary);

if(finout.fail())

{

cout << "\n Error opening file";

}

else

{

do

{

if(n == MAX)

{

cout << "\n Hast table is full ";

}

else

{

Employee e1,e2;

e1.getemp();

key = divisionmethod(e1.name);

finout.seekg(key\*sizeof(e1),ios::beg);

finout.read((char\*)&e2,sizeof(e2));

if( strcmp(e2.name,"---------") == 0)

{

finout.seekp(key\*sizeof(e1),ios::beg);

finout.write((char\*)&e1,sizeof(e1));

n++;

}

else

{

cout << "\n Collision occured ";

oldkey = key;

// for record

while(1)

{

finout.seekg(key\*sizeof(e1),ios::beg);

finout.read((char\*)&e2,sizeof(e2));

if( strcmp(e2.name,"---------") == 0)

{

finout.seekp(key\*sizeof(e1),ios::beg);

finout.write((char\*)&e1,sizeof(e1));

n++;

break;

}

else

{

key = (key + 1) % MAX;

}

}

//for chain

while(1)

{

finout.seekg(oldkey\*sizeof(e1),ios::beg);

finout.read((char\*)&e2,sizeof(e2));

if( e2.chain == -1)

{

e2.chain = key;

finout.seekp(oldkey\*sizeof(e1),ios::beg);

finout.write((char\*)&e2,sizeof(e2));

break;

}

else

{

oldkey = e2.chain;

}

}

}

}

cout << "\n Do u want to add more rec :y.n -> ";

cin >> ch;

}while(ch!='n');

}

finout.close();

}

void HashFile :: initalize()

{

fstream finout;

finout.open(fname,ios::out | ios::binary);

if(finout.fail())

{

cout << "\n Error opening file";

}

else

{

Employee e;

e.chain = -1;

strcpy(e.name,"---------");

strcpy(e.telno,"---------");

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

{

finout.write((char\*)&e,sizeof(e));

}

}

finout.close();

}

void HashFile :: displayht()

{

fstream finout;

finout.open(fname,ios::in | ios::binary);

if(finout.fail())

{

cout << "\n Error opening file";

}

else

{

cout << "\n NAME \t TELNO \t CHAIN";

Employee e;

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

{

finout.read((char\*)&e,sizeof(e));

e.show();

}

}

finout.close();

}

void HashFile :: menu()

{

int ch;

char sname[10];

initalize();

do

{

cout << "\n 1 : Add Record \n 2 : Display Records \n 3 : Search by name \n 4 : Exit ";

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

cin >> ch;

switch(ch)

{

case 1 : add\_chain\_without\_rep();break;

case 2 : displayht();break;

case 3 : cout << "\n Enter the name : ";

cin >> sname;

searchht(sname);

break;

}

}while(ch!=4);

}

int main()

{

HashFile obj;

obj.menu();

}