**Assignment number: 9**

**Subject: ADVANCED DATA STRUCTURES LAB**

Name: ***RIA MITTAL***

Class: ***SECOND YEAR ENGINEERING***

Division: ***B***

Roll no: ***222008***

Batch: ***B1***

**PROBLEM STATEMENT:**

Consider telephone book database of N clients. Make use of a hash table implementation to quickly look up client‘s telephone number.

**Code:**

#include<iostream>

using namespace std;

int n;

class data

{

public:

long long int arr;

char name[100];

};

class hashing

{

int v[10];

long long int hash[10];

public:

data d[10];

int accept(int);

int display(int);

int hashed();

int display\_hash();

int search();

};

int hashing::accept(int i)

{

cout<<"\nenter the name of client\n";

ws(cin);

cin.getline(d[i].name,100);

cout<<"\nenter the telephone number\n";

cin>>d[i].arr;

}

int hashing::display(int i)

{

cout<<d[i].name<<"\t"<<d[i].arr<<endl;

}

int hashing::hashed()

{

for(int i=0;i<10;i++)

{

v[i]=0;

hash[i]=0;

}

int i=0,j=0;

while(j<n)

{

int mod=d[j].arr%10;

if(v[mod]!=1 )

{

hash[mod]=d[j].arr;

v[mod]=1;

}

else

{

while(v[mod]!=0 )

{

if(mod==9)

{

mod=0;

}

else

mod++;

}

hash[mod]=d[j].arr;

v[mod]=1;

}

j++;

}

}

int hashing::display\_hash()

{

cout<<"\nthe hashed function is\n";

for(int i=0;i<10;i++)

{

if(hash[i]==0)

cout<<i<<"\t----\n";

else

cout<<i<<"\t"<<hash[i]<<"\n";

}

}

int hashing::search()

{

long long int num;

int count=0;

cout<<"\nenter the telephone number to be searched\n";

cin>>num;

for(int i=0;i<10;i++)

{

if(hash[i]==num)

count=1;

else

continue;

}

if(count==1)

cout<<"\nthe number is found in the list\n";

else

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

}

int main()

{

hashing obj;

char ch,c;

int ans;

do

{

cout<<"\nenter the number of records\n";

cin>>n;

if(n>10)

cout<<"\nyou can enter maximum 10 client details\n";

else

{

do

{

cout<<"\n1.)ACCEPT\n2.)DISPLAY DETAILS\n3.)DISPLAY HASH TABLE\n4.)SEARCH TELEPHONE NUMBER\n5.)EXIT\n";

cin>>ans;

switch(ans)

{

case 1:{

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

obj.accept(i);

break;

}

case 2:{

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

obj.display(i);

break;

}

case 3:{

obj.hashed();

obj.display\_hash();

break;

}

case 4:{

obj.search();

break;

}

case 5:continue;

default: cout<<"\ninvalid entry...try again\n";

}

cout<<"\ndo you want to continue?(y/n)\n";

cin>>c;

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

}

cout<<"\ndo you want to make new entries?(y/n)\n";

cin>>ch;

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

}

**OUTPUT:**

enter the number of records

5

1.)ACCEPT

2.)DISPLAY DETAILS

3.)DISPLAY HASH TABLE

4.)SEARCH TELEPHONE NUMBER

5.)EXIT

1

enter the name of client

ria

enter the telephone number

919191919119

enter the name of client

air

enter the telephone number

56426475372669

enter the name of client

abcd

enter the telephone number

67682514999

enter the name of client

cdefg

enter the telephone number

15423321110

enter the name of client

hijklmn

enter the telephone number

4341454365572

do you want to continue?(y/n)

y

1.)ACCEPT

2.)DISPLAY DETAILS

3.)DISPLAY HASH TABLE

4.)SEARCH TELEPHONE NUMBER

5.)EXIT

2

ria 919191919119

air 56426475372669

abcd 67682514999

cdefg 15423321110

hijklmn 4341454365572

do you want to continue?(y/n)

y

1.)ACCEPT

2.)DISPLAY DETAILS

3.)DISPLAY HASH TABLE

4.)SEARCH TELEPHONE NUMBER

5.)EXIT

3

the hashed function is

0 56426475372669

1 67682514999

2 15423321110

3 4341454365572

4 ----

5 ----

6 ----

7 ----

8 ----

9 919191919119

do you want to continue?(y/n)

y

1.)ACCEPT

2.)DISPLAY DETAILS

3.)DISPLAY HASH TABLE

4.)SEARCH TELEPHONE NUMBER

5.)EXIT

4

enter the telephone number to be searched

919191919119

the number is found in the list

do you want to continue?(y/n)

y

1.)ACCEPT

2.)DISPLAY DETAILS

3.)DISPLAY HASH TABLE

4.)SEARCH TELEPHONE NUMBER

5.)EXIT

4

enter the telephone number to be searched

6436737363

--------not found----------

do you want to continue?(y/n)

y

1.)ACCEPT

2.)DISPLAY DETAILS

3.)DISPLAY HASH TABLE

4.)SEARCH TELEPHONE NUMBER

5.)EXIT

5