Advanced Data Structures Assignment

Name : Shrirang R Mhalgi

Roll number : 222006

Class : SE B

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 i,m,j,k;

int temp,key;

string c;

class hash\_t{

public:

int no;

string name;

hash\_t(){

no=0;

name='-';

}

}o[10];

class hash : public hash\_t{

int arr[20];

string n[20];

int index,v;

public:

int create();

void display();

void create\_hasht();

int search(int );

int search1(string);

int hashf(int temp){

index=temp%10;

return index;

}

};

int hash :: create(){

cout<<"\nEnter the telephone number : ";

cin>>arr[i];

cout<<"Enter name : ";

cin>>n[i];

}

void hash :: display(){

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

{

//cout<<n[i]<<"\t"<<arr[i];

cout<<o[i].no<<"\t"<<o[i].name<<endl;

}

}

void hash :: create\_hasht(){

i=0;

while(i<10 && i<m){

temp=arr[i];

c=n[i];

j=hashf(temp);

if(o[j].no==0)

{

o[j].no =temp;

o[j].name =c;

}

else

{

int k=0;

while(k<10)

{

if(o[j].no!=0)

j=(j+1)%10;

else

{

o[j].no =temp;

o[j].name =c;

break;

}

}

}

i++;

}

}

int hash :: search(int key){

j=hashf(key);

i=0;

int flag=0;

while(i<10 && i<m){

if(o[j].no==key)

{

cout<<"\nNumber found\n";

cout<<"Number : "<<o[j].no;

cout<<"\nName : "<<o[j].name;

flag=1; v=2;

break;

}

else

{

j=(j+1)%10;

}

i++;

}

if(flag==0)

cout<<"Number not found ";

}

int hash :: search1(string s){

for(i=0;i<m;i++){

if(n[i]==s)

search(arr[i]);

if(v==2)

return 0;

}

}

int main(){

int ans,temp1;

char choice;

hash b;

cout<<"Enter number of records : ";

cin>>m;

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

b.create();

//v=i;

b.create\_hasht();

do{

cout<<"\n1: Display hash table\n";

cout<<"2: Search\n";

cout<<"3: Update\n";

cout<<"4: Exit\n";

cout<<"Enter your choice : ";

cin>>ans;

switch(ans){

case 1 : b.display(); break;

case 2 : cout<<"Enter the telephone number to be searched : ";

cin>>key;

b.search(key);

break;

case 3 : cout<<"\na: Update telephone number ";

cout<<"\nb: Update name";

cout<<"\nEnter your choice : ";

cin>>choice;

switch(choice){

case 'a' : cout<<"Enter name : ";

cin>>c;

b.search1(c);

cout<<"Enter updated number : ";

cin>>temp1;

k=b.hashf(temp1);

if(o[k].no!=0)

{

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

{

k=(k+1)%10;

if(o[k].no==0)

break;

}

}

o[k].no=temp1;o[k].name=c;

o[j].no=0; o[j].name='-';

break;

case 'b' : cout<<"Enter number : ";

cin>>temp1;

b.search(temp1);

cout<<"Enter updated name : ";

cin>>c;

o[j].name=c;

break;

}break;

}

}while(ans!=4);

}

Output:

Enter number of records : 3

Enter the telephone number : 2580

Enter name : abc

Enter the telephone number : 1234

Enter name : gfg

Enter the telephone number : 5678

Enter name : xyz

1: Display hash table

2: Search

3: Update

4: Exit

Enter your choice : 1

2580 abc

0 -

0 -

0 -

1234 gfg

0 -

0 -

0 -

5678 xyz

0 -

1: Display hash table

2: Search

3: Update

4: Exit

Enter your choice : 2

Enter the telephone number to be searched : 5678

Number found

Number : 5678

Name : xyz

1: Display hash table

2: Search

3: Update

4: Exit

Enter your choice : 3

a: Update telephone number

b: Update name

Enter your choice : a

Enter name : abc

Number found

Number : 2580

Name : abcEnter updated number : 1234

1: Display hash table

2: Search

3: Update

4: Exit

Enter your choice : 4