#include <iostream>

using namespace std;

class node {

public:

long long telno;

string name;

node\* next;

node() {

next = NULL; //constructor

}

};

class hashing {

node\* hashtable1[10];

node\* hashtable2[10];

int flagval[10]; //marks whether a slot in hashtable2 is filled (0 = empty, 1 = full)

public:

hashing() {

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

hashtable1[i] = NULL;

hashtable2[i] = NULL;

flagval[i]=0;

}

}

void insert1(){ //Chaining

long long key;

cout<<"enter the telephone no :-";

cin>>key;

cout<<"enter the name of client :-";

string name;

cin>>name;

node\* p = new node;

p->telno = key;

p->name = name;

int loc = key % 10;

if (hashtable1[loc] == NULL) { //If the slot is empty, place the node there.

hashtable1[loc] = p;

} else {

node\* q = hashtable1[loc]; //Otherwise, add the node at the end of the linked list (chaining).

while (q->next != NULL) {

q = q->next;

}

q->next = p;

}

}

void display1() { //Chaining

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

node\* p = hashtable1[i];

if (p != NULL) {

cout<< i<<" th row:-\n";

while (p != NULL) {

cout << p->telno << "," << p->name << "\n";

p = p->next;

}

cout << endl;

}

}

}

void search1(){ //Linear Probing

long long key;

cout<<"enter which key you want to search:-";

cin>>key;

int loc = key % 10;

node\* p = hashtable1[loc];

while(p != NULL){

if(p->telno==key){

cout<<"the key is found\n";

return;

}

p = p->next;

}

cout<<"the key is not found\n";

}

void insert2() { //Linear Probing

long long key;

cout << "Enter the telephone no: ";

cin >> key;

cout << "Enter the name: ";

string name;

cin >> name;

node\* p = new node;

p->telno = key;

p->name = name;

p->next = NULL;

int loc = key % 10;

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

if (flagval[loc] == 0) {

hashtable2[loc] = p;

flagval[loc] = 1;

break;

} else {

loc = (loc + 1) % 10;

}

}

}

void search2() {

long long key;

cout << "Which key do you want to search: ";

cin >> key;

int loc = key % 10;

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

if (flagval[loc] == 1 && hashtable2[loc]->telno == key) {

cout << "Key is found: " << hashtable2[loc]->name << endl;

return;

} else {

loc = (loc + 1) % 10;

}

}

cout << "Key is not found." << endl;

}

void display2() {

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

if (hashtable2[i] != NULL) {

cout << hashtable2[i]->telno << ", " << hashtable2[i]->name << "\n";

cout << i << " th row:-\n";

}

}

}

void menu(){

int ch;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\* MENU \*\*\*\*\*\*\*\*\*\*\n";

cout << "1. insert1\n2.search1\n3.Display1\n4.insert2\n5.search2\n6.display2\n7.Exit\n";

cout<<"Enter your choice:";

cin>>ch;

while(ch!=7){

switch (ch) {

case 1: insert1(); break;

case 2: search1(); break;

case 3: display1(); break;

case 4: insert2(); break;

case 5: search2(); break;

case 6: display2(); break;

case 7: cout << "Exiting...\n"; break;

default: cout << "Invalid choice\n"; break;

}

cout<<"enter Your Choice:";

cin>>ch;

}

}

};

int main() {

hashing obj;

obj.menu();

return 0;

}