#include<iostream>

#include<string.h>

using namespace std;

struct node {

int value;

node \*next;

} \*HashTable[10];

class hashing {

public:

hashing() {

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

HashTable[i] = NULL;

}

}

int HashFunction(int value) {

return (value % 10);

}

node \*create\_node(int x) {

node \*temp = new node;

temp->next = NULL;

temp->value = x;

return temp;

}

void display() {

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

node \*temp = HashTable[i];

cout << "a[" << i << "]";

while (temp != NULL) {

cout << "->" << temp->value;

temp = temp->next;

}

cout << "\n";

}

}

int searchElement(int value) {

bool flag = false;

int hash\_val = HashFunction(value);

node \*entry = HashTable[hash\_val];

while (entry != NULL) {

cout << "Element found at: ";

if (entry->value == value) {

cout << hash\_val << ":" << entry->value << endl;

flag = true;

break;

}

entry = entry->next;

}

if (!flag) {

cout << "Element is not found" << endl;

return -1;

}

return hash\_val;

}

void deleteElement(int value) {

int hash\_val = HashFunction(value);

node \*entry = HashTable[hash\_val];

node \*prev = NULL;

if (entry == NULL) {

cout << "No Element found" << endl;

return;

}

if (entry->value == value) {

HashTable[hash\_val] = entry->next;

delete entry;

cout << "Element is deleted" << endl;

return;

}

while (entry != NULL && entry->value != value) {

prev = entry;

entry = entry->next;

}

if (entry == NULL) {

cout << "Element not found" << endl;

} else {

prev->next = entry->next;

delete entry;

cout << "Element is deleted" << endl;

}

}

void insertElement(int value) {

int hash\_val = HashFunction(value);

node \*head = create\_node(value);

node \*temp = HashTable[hash\_val];

if (temp == NULL) {

HashTable[hash\_val] = head;

} else {

while (temp->next != NULL) {

temp = temp->next;

}

temp->next = head;

}

}

};

int main() {

int ch;

int data, search, del;

hashing h;

do {

cout << "---------TELEPHONE---------" << endl;

cout << "1.Insert" << endl;

cout << "2.Display" << endl;

cout << "3.Search" << endl;

cout << "4.Delete" << endl;

cout << "5.Exit" << endl;

cout << "Enter your choice: ";

cin >> ch;

switch (ch) {

case 1:

cout << "Enter phone no. to be inserted: ";

cin >> data;

h.insertElement(data);

break;

case 2:

h.display();

break;

case 3:

cout << "Enter the no. to be searched: ";

cin >> search;

if (h.searchElement(search) == -1) {

continue;

}

break;

case 4:

cout << "Enter the phone no. to be deleted: ";

cin >> del;

h.deleteElement(del);

break;

case 5:

cout << "Exiting program." << endl;

break;

}

} while (ch != 5);

return 0;

}

OUTPUT:

---------TELEPHONE---------

1.Insert

2.Display

3.Search

4.Delete

5.Exit

Enter your choice: 1

Enter phone no. to be inserted: 23

---------TELEPHONE---------

1.Insert

2.Display

3.Search

4.Delete

5.Exit

Enter your choice: 1

Enter phone no. to be inserted: 33

---------TELEPHONE---------

1.Insert

2.Display

3.Search

4.Delete

5.Exit

Enter your choice: 1

Enter phone no. to be inserted: 34

---------TELEPHONE---------

1.Insert

2.Display

3.Search

4.Delete

5.Exit

Enter your choice: 2

a[0]

a[1]

a[2]

a[3]->23->33

a[4]->34

a[5]

a[6]

a[7]

a[8]

a[9]

---------TELEPHONE---------

1.Insert

2.Display

3.Search

4.Delete

5.Exit

Enter your choice: 3

Enter the no. to be searched: 45

Element is not found

---------TELEPHONE---------

1.Insert

2.Display

3.Search

4.Delete

5.Exit

Enter your choice: 3

Enter the no. to be searched: 23

Element found at: 3:23

---------TELEPHONE---------

1.Insert

2.Display

3.Search

4.Delete

5.Exit

Enter your choice: 4

Enter the phone no. to be deleted: 34

Element is deleted

---------TELEPHONE---------

1.Insert

2.Display

3.Search

4.Delete

5.Exit

Enter your choice: 4

Enter the phone no. to be deleted: 67

No Element found

---------TELEPHONE---------

1.Insert

2.Display

3.Search

4.Delete

5.Exit

Enter your choice: 5

Exiting program.