Practical no:02

Program:

```
#include <iostream>
#include <cstring>
using namespace std;
class HashFunction {
   struct hash {
      long key;
      char name[10];
   } h[10];
public:
   HashFunction();
   void insert();
   void display();
   int find(long);
   void Delete(long);
};
HashFunction::HashFunction() {
   for (int i = 0; i < 10; i++) {
      h[i].key = -1;
      strcpy(h[i].name, "NULL");
   }
}
void HashFunction::Delete(long k) {
   int index = find(k);
   if (index == -1) {
      cout << "\n\tKey Not Found";
   } else {
      h[index].key = -1;
      strcpy(h[index].name, "NULL");
      cout << "\n\tKey is Deleted";
   }
}
int HashFunction::find(long k) {
   for (int i = 0; i < 10; i++) {
      if (h[i].key == k) {
         cout << "\n\t" << h[i].key << " is Found at " << i << " Location With Name "
<< h[i].name;
         return i;
      }
   }
   return -1;
}
void HashFunction::display() {
   cout << "\n\t\tKey\t\tName";
   for (int i = 0; i < 10; i++) {
      cout << "\n\th[" << i << "]\t" << h[i].key << "\t\t" << h[i].name;
   }
}
```

```
void HashFunction::insert() {
    char ans, n[10], ntemp[10];
    long k, temp;
    int hi, cnt = 0, flag = 0;

    do {
        if (cnt >= 10) {
            cout << "\n\tHash Table is FULL";
            break;
        }

        cout << "\n\tEnter a Telephone No: ";
        cin >> k;
        cout << "\n\tEnter a Client Name: ";
        cin >> n;

        hi = k % 10;

        if (h[hi].key == -1) {
            h[hi].key = k;
            strcpy(h[hi].name, n);
        } else {
            temp = h[hi].key;
            strcpy(ntemp, h[hi].name);
            h[hi].key = k;
            strcpy(h[hi].name, n);

            for (int i = (hi + 1) % 10; i != hi; i = (i + 1) % 10) {
                if (h[i].key == -1) {
                    h[i].key = temp;
                    strcpy(h[i].name, ntemp);
                    flag = 1;
                    break;
                }
            }

            if (flag == 0) {
                cout << "\n\tHash Table is FULL can not insert the elment in the hash
table!";
                break;
            }
        }
        flag = 0;
        cnt++;
        cout << "\n\t..... Do You Want to Insert More Key: y/n";
        cin >> ans;
    } while (ans == 'y' || ans == 'Y');
}
```

```
int main() {
   long k;
   int ch, index;
   char ans;
   HashFunction obj;
   do {
      cout << "\n\t***** Telephone (ADT) *****";
      cout << "\n\t1. Insert\n\t2. Display\n\t3. Find\n\t4. Delete\n\t5. Exit";
      cout << "\n\t..... Enter Your Choice: ";
      cin >> ch;
      switch (ch) {
         case 1:
            obj.insert();
            break;
         case 2:
            obj.display();
            break;
         case 3:
            cout << "\n\tEnter a Key Which You Want to Search: ";
            cin >> k;
            index = obj.find(k);
            if (index == -1) {
               cout << "\n\tKey Not Found";
            }
            break;
         case 4:
            cout << "\n\tEnter a Key Which You Want to Delete: ";
            cin >> k;
            obj.Delete(k);
            break;
         case 5:
          cout<<"Thank for using this program!!"<<endl;
            break;
         default:
            cout << "\n\tInvalid Choice!";
            break;
      }
      cout << "\n\t..... Do You Want to Continue in Main Menu: ";
      cin >> ans;
   } while (ans == 'y' || ans == 'Y');

   return 0;
}
```

Output:

```

      ***** Telephone (ADT) *****
      1. Insert
      2. Display
```

```
    3. Find
    4. Delete
    5. Exit
    ..... Enter Your Choice: 1

    Enter a Telephone No: 101

    Enter a Client Name:
a

    ..... Do You Want to Insert More Key: y/ny

    Enter a Telephone No: 102

    Enter a Client Name: a

    ..... Do You Want to Insert More Key: y/ny

    Enter a Telephone No: 103

    Enter a Client Name: a

    ..... Do You Want to Insert More Key: y/ny

    Enter a Telephone No: 104

    Enter a Client Name: a

    ..... Do You Want to Insert More Key: y/ny

    Enter a Telephone No: 105

    Enter a Client Name: a

    ..... Do You Want to Insert More Key: y/ny

    Enter a Telephone No: 106

    Enter a Client Name: a

    ..... Do You Want to Insert More Key: y/ny

    Enter a Telephone No: 107

    Enter a Client Name: a

    ..... Do You Want to Insert More Key: y/ny

    Enter a Telephone No: 108
Enter a Client Name: a

..... Do You Want to Insert More Key: y/ny

Enter a Telephone No: 109

Enter a Client Name: a

..... Do You Want to Insert More Key: y/nn

..... Do You Want to Continue in Main Menu: y

***** Telephone (ADT) *****
1. Insert
2. Display
3. Find
4. Delete
5. Exit
..... Enter Your Choice: 2

      Key          Name
h[0]   -1            NULL
h[1]   101            a
h[2]   102            a
h[3]   103            a
h[4]   104            a
h[5]   105            a
h[6]   106            a
h[7]   107            a
h[8]   108            a
h[9]   109            a
..... Do You Want to Continue in Main Menu: y

***** Telephone (ADT) *****
1. Insert
2. Display
3. Find
4. Delete
5. Exit
..... Enter Your Choice: 1

Enter a Telephone No: 100

Enter a Client Name: a

..... Do You Want to Insert More Key: y/nn

..... Do You Want to Continue in Main Menu:
```

```
y

     ***** Telephone (ADT) *****
     1. Insert
     2. Display
     3. Find
     4. Delete
     5. Exit
     ..... Enter Your Choice: 1

     Enter a Telephone No: 123

     Enter a Client Name: a

     Hash Table is FULL can not insert the elment in the hash table!
     ..... Do You Want to Continue in Main Menu: y

     ***** Telephone (ADT) *****
     1. Insert
     2. Display
     3. Find
     4. Delete
     5. Exit
     ..... Enter Your Choice: 2

          Key        Name
     h[0]   100        a
     h[1]   101        a
     h[2]   102        a
     h[3]   123        a
     h[4]   104        a
     h[5]   105        a
     h[6]   106        a
     h[7]   107        a
     h[8]   108        a
     h[9]   109        a
     ..... Do You Want to Continue in Main Menu: y

     ***** Telephone (ADT) *****
     1. Insert
     2. Display
     3. Find
     4. Delete
     5. Exit
     ..... Enter Your Choice: 3

     Enter a Key Which You Want to Search: 102

     102 is Found at 2 Location With Name a
```

```
..... Do You Want to Continue in Main Menu: y

***** Telephone (ADT) *****
1. Insert
2. Display
3. Find
4. Delete
5. Exit
..... Enter Your Choice: 3

Enter a Key Which You Want to Search: 128

Key Not Found
..... Do You Want to Continue in Main Menu: y

***** Telephone (ADT) *****
1. Insert
2. Display
3. Find
4. Delete
5. Exit
..... Enter Your Choice: 4

Enter a Key Which You Want to Delete: 100

100 is Found at 0 Location With Name a
Key is Deleted
..... Do You Want to Continue in Main Menu: y

***** Telephone (ADT) *****
1. Insert
2. Display
3. Find
4. Delete
5. Exit
..... Enter Your Choice: 2

      Key         Name
h[0]   -1          NULL
h[1]   101          a
h[2]   102          a
h[3]   123          a
h[4]   104          a
h[5]   105          a
h[6]   106          a
h[7]   107          a
h[8]   108          a
h[9]   109          a
..... Do You Want to Continue in Main Menu: y
```

```
        ***** Telephone (ADT) *****
        1. Insert
        2. Display
        3. Find
        4. Delete
        5. Exit
        ..... Enter Your Choice: 5
Thank for using this program!!
```