## Assignment no:2

## Input

```
#include<iostream>
#include<string.h>
using namespace std;
class HashFunction
 {
        typedef struct hash
        {
                long key;
                char name[10];
}hash;
 hash h[10];
          public:
 HashFunction();
                void insert();
                void display();
                int find(long);
                        void Delete(long);
 };
HashFunction::HashFunction()
 {
        int i;
        for(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)
 {
        int i;
        for(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;
                 }
        }
if(i==10)
        {
                return -1;
  }
 }
void HashFunction::display()
```

```
 {
        int i;
        cout<<"\n\t\tKey\t\tName";
        for(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 v,hi,cnt=0,flag=0,i;
do
        {
                if(cnt>=10)
                {
                        cout<<"\n\tHash Table is FULL";
                        break;
                }
                cout<<"\n\tEnter a Telephone No: ";      //accept no from user
                cin>>k;
                cout<<"\n\tEnter a Client Name: ";      //accept name from user
                cin>>n;
                hi=k%10;// hash function
                if(h[hi].key==-1)                       //if location is empty
                  {
                        h[hi].key=k;
                         strcpy(h[hi].name,n);
                }
```

```
else
{
 if(h[hi].key%10!=hi)     //check location is not empty and wheather the something is present is at actual position?
{
 temp=h[hi].key;
strcpy(ntemp,h[hi].name);
h[hi].key=k;
strcpy(h[hi].name,n);
for(i=hi+1;i<10;i++)
 {
  if(h[i].key==-1)
   {
    h[i].key=temp;
    strcpy(h[i].name,ntemp);
    flag=1;   //done the replacement
    break;
  }
 }
for(i=0;i<hi && flag==0;i++) //To restart the search from first location for empty space
{
  if(h[i].key==-1)
  {
    h[i].key=temp;
    strcpy(h[i].name,ntemp);
    break;
   }
  }
 }
else  //if current key is at correct location then no need of replacement
```

```
                {
                        for(i=hi+1;i<10;i++)
                        {
                                if(h[i].key==-1)
                                {
                                        h[i].key=k;
                                        strcpy(h[i].name,n);
                                        flag=1;    //we store that element at that location
                                        break;
                                }
                        }
                        for(i=0;i<hi && flag==0;i++)  //to restart the search from location for empty space(search upword)
                        {
                                if(h[i].key==-1)
                                {
                                        h[i].key=k;
                                        strcpy(h[i].name,n);
                                        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:
                                  break;
```

```
            }
            cout<<"\n\t..... Do You Want to Continue in Main Menu:y/n ";
            cin>>ans;
        }while(ans=='y'||ans=='Y');
}
```

## Output:

```
*** Telephone (ADT) *****
1. Insert
2. Display
3. Find
4. Delete
5. Exit
..... Enter Your Choice: 1
Enter a Telephone No: 20
Enter a Client Name: Jack
..... Do You Want to Insert More Key: y/ny
Enter a Telephone No: 56
Enter a Client Name: Piyush
..... Do You Want to Insert More Key: y/ny
Enter a Telephone No: 89
Enter a Client Name: Kriya
..... Do You Want to Insert More Key: y/nn
..... Do You Want to Continue in Main Menu:y/n y
*** Telephone (ADT) *****
1. Insert
2. Display
3. Find
4. Delete
5. Exit
```

..... Enter Your Choice: 2

| | Key | Name |
|---|---|---|
| h[0] | 20 | Jack |
| h[1] | -1 | NULL |
| h[2] | -1 | NULL |
| h[3] | -1 | NULL |
| h[4] | -1 | NULL |
| h[5] | -1 | NULL |
| h[6] | 56 | Piyush |
| h[7] | -1 | NULL |
| h[8] | -1 | NULL |
| h[9] | 89 | Kriya |

..... Do You Want to Continue in Main Menu:y/n 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: 89

89 is Found at 9 Location With Name Kriya

..... Do You Want to Continue in Main Menu:y/n 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: 56

56 is Found at 6 Location With Name Piyush

Key is Deleted

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

*** Telephone (ADT) *****

1. Insert

2. Display

3. Find

4. Delete

5. Exit

..... Enter Your Choice: 2

| | Key | Name |
|---|---|---|
| h[0] | 20 | Jack |
| h[1] | -1 | NULL |
| h[2] | -1 | NULL |
| h[3] | -1 | NULL |
| h[4] | -1 | NULL |
| h[5] | -1 | NULL |
| h[6] | -1 | NULL |
| h[7] | -1 | NULL |
| h[8] | -1 | NULL |
| h[9] | 89 | Kriya |

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

*** Telephone (ADT) *****

1. Insert

2. Display

3. Find

4. Delete

5. Exit

..... Enter Your Choice: 5

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

---

Process exited after 166.9 seconds with return value 0

Press any key to continue . . .