

```

#include <iostream>
#include <string.h>
using namespace std;
class Dictionary
{
    typedef struct obj
    {
        long key;
        char clnt_name[10];
    } obj;
    obj h[10];

public:
    Dictionary();
    void add_record();
    void show();
    int search(long);
    void Delete_rec(long);
};
Dictionary::Dictionary()
{
    int i;
    for (i = 0; i < 10; i++)
    {
        h[i].key = -1;
        strcpy(h[i].clnt_name, "NULL");
    }
}
void Dictionary::Delete_rec(long k)
{
    int index = search(k);
    if (index == -1)
    {
        cout << "\n---Key not found--- ";
    }
    else
    {
        h[index].key = -1;
        strcpy(h[index].clnt_name, "NULL");
        cout << "\n\t---Key is deleted--- ";
    }
}
int Dictionary::search(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 Client " << h[i].clnt_name;
            return i;
        }
    }
    if (i == 10)
    {

```

```

        return -1;
    }
}

void Dictionary::show()
{
    int i;
    cout << "\n\tkey\t\tClnt_name ";
    for (i = 0; i < 10; i++)
    {
        cout << "\n\tth[" << i << "]\t" << h[i].key << "\t\t" << h[i].clnt_name;
    }
}

void Dictionary::add_record()
{
    char ans, n[10], ntemp[10];
    long k, temp;
    int v, hi, ent = 0, flag = 0, i;
    do
    {
        if (ent >= 10)
        {
            cout << "\nHash table is full !";
            break;
        }
        cout << "\n -Enter telephone No. ";
        cin >> k;
        cout << "\n -Enter client name: ";
        cin >> n;
        hi = k % 10;
        if (h[hi].key == -1)
        {
            h[hi].key = k;
            strcpy(h[hi].clnt_name, n);
        }
        else
        {
            if (h[hi].key % 10 != hi)
            {
                temp = h[hi].key;
                strcpy(ntemp, h[hi].clnt_name);
                h[hi].key = k;
                strcpy(h[hi].clnt_name, n);
                for (i = hi + 1; i < 10; i++)
                {
                    if (h[i].key == -1)
                    {
                        h[i].key = temp;
                        strcpy(h[i].clnt_name, ntemp);
                        flag = 1;
                        break;
                    }
                }
            }
        }
    }
    else
    {

```

```

        for (i = hi + 1; i < 10; i++)
        {
            if (h[i].key == -1)
            {
                h[i].key = k;
                strcpy(h[i].clnt_name, n);
                flag = 1;
                break;
            }
        }
        for (i = 0; i < hi && flag == 0; i++)
        {
            if (h[i].key == -1)
            {
                h[i].key = k;
                strcpy(h[i].clnt_name, n);
                break;
            }
        }
    }
    flag = 0;
    ent++;
    cout << "\n*Do you want to insert more keys(to insert enter y) ";
    cin >> ans;
} while (ans == 'y' || ans == 'Y');
}

int main()
{
    long k;
    int ch, index;
    char ans;
    Dictionary obj;
    do
    {
        cout << "\n\t***Telephone(adtt)***";
        cout << "\n1.Insert\n2.Display\n3.Find\n4.Delete\n5.Exit";
        cout << "\nSelect your choice ";
        cin >> ch;
        switch (ch)
        {
            case 1:
                obj.add_record();
                break;
            case 2:
                obj.show();
                break;
            case 3:
                cout << "\nEnter key to be searched : ";
                cin >> k;
                index = obj.search(k);
                if (index == -1)
                {
                    cout << "\n---Key not found--- ";
                }
            }
        }
    } while (ans != 'n');
}

```

```
        break;
    case 4:
        cout << "\nEnter element to be deleted - ";
        cin >> k;
        obj.Delete_rec(k);
        break;
    case 5:
        break;
    }
    cout << "\n*Do you want to continue Menu:y/n ";
    cin >> ans;
} while (ans == 'y' || ans == 'Y');
}
```