Harsh Shankar
IBM18CS032

```
#include<stdio
#include<bits/stdc++.h>
using namespace std;

typedef struct list
{
    int data;
    struct list *next;
} node_type;

node_type *ptr[max], *root[max], *temp[max];

class Dictionary
{

public:
        int index;
        Dictionary();
        void insert(int);
        void search(int);
        void delete_de(int);
};

void Dictionary::insert(int key)
{
    index = int(key % max);
    ptr[index] = (node_type*) malloc(sizeof(node_type));
    ptr[index]->data = key;
```

Date _____

Expt. No. _____ Page No. _____

```
if (root[index] == NULL)
{
    root[index] = ptr[index];
    root[index] -> next = NULL;
    temp[index] = ptr[index];
}
else
{
    temp[index] = root[index];
    while (temp[index] -> next != NULL)
        temp[index] = temp[index] -> next;
    temp[index] -> next = ptr[index]
}
}

void Dictionary :: search (int key)
{
    int flag = 0;
    index = int (key % omax);
    temp[index] = root[index];
    while (temp[index] != NULL)
    {
        if (temp[index] -> data == key)
        {
            cout << "\n Search found!";
            flag = 1;
            break;
        }
```

Teacher's Signature : _____

```
        else temp[index] = temp[index]->next;
    }
    if (flag == 0)
        cout << "\n search key not found...";
}

void Dictionary::delete_el(int key)
{
    index = int(key%max);
    temp[index] = root[index];
    while (temp[index]->data != key && temp[index] != NULL)
    {
        ptr[index] = temp[index];
        temp[index] = temp[index]->next;
        cout << "\n" << temp[index] << "has been deleted";
        temp[index]->data = -1;
        temp[index] = NULL;
        free(temp[index]);
}
```