Snehita.J
1BM18CS109

```
#include <iostream>
#include <stdlib.h>
using namespace std; #define max 10
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_ele(int);
};
Dictionary::Dictionary()
{   index = -1;
    for(int i=0; i<max; i++)
    {   root[i] = NULL;
        ptr[i] = NULL;
        temp[i] = NULL;
    }
}
void Dictionary::insert(int key)
{   index = int(key%max);
    ptr[index] = (node_type*) malloc(sizeof(node_type));
    if(root[index]==NULL)
    {   root[index] = ptr[index];
        root[index]->next = NULL;
        temp[index] = ptr[index];
    }
    else
    {   temp[index]->root = index
    }   temp[index]->next = ptr[index];
}
```

Snehita-1
1BM18CS109

```
void Dictionary :: search (int key)
{
    int flag=0;
    index = int (key % max);
    temp[index] = root[index];
    while (temp[index] != NULL)
    {
        if (temp[index]->data == key)
        {
            cout << "Found";
            flag=1; break;
        }
        else temp[index] = temp[index]->next;
    }
    if(flag==0)
        cout << " Not found";
}
void Dictionary :: delete_ele (int key)
{
    index = int (key % max);
    temp[index] = root[index];
    while (temp[index]->data != key && temp[index] != NULL)
    {
        pt[index] = temp[index];
        temp[index] = temp[index]->next;
    }
    pt[index]->next = temp[index]->next;
    cout << temp[index]->data << " deleted";
    temp[index]->data = -1;
    temp[index] = NULL; free(temp[index]);
}
int main()
{
    int val, ch, x, num;
    Dictionary d;
    do
    {
        cout<<"1: Create, 2: Search 3 Delete";
        cin >> ch;
        switch(ch)
        {
```

Snehita. S
1BM18CS109

```
vo
{
        case 1 : cout<<"Enter no of elements to be inserted";
                 cin>>n;
                 cout<<"Enter elements";
                 for(int i=0;i<n;i++)
                 {   cin>>num;
                     d.insert(num);
                 } break;
        case 2 : cout<<"Enter search ele";
                 cin>>n;
                 d.search(n);
        case 3 : cout<<"Enter element to be deleted";
                 cin>>n;
                 d.delete_ele(n);
                 break;
        default : cout<<"Invalid choice";
    } cout<<"Enter y to continue";
      cin>>c;
    } while(c=='y');
    return 0;
}
```