Prathibha.R
1BM18CS074

```
#include<iostream>
#include<conio.h>
#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));
```

```
ptr[index] -> data = key;
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 % max);
    temp[index] = root[index];
    while (temp[index] != NULL)
    {
        if (temp[index] -> data == key)
        {
            cout << "\n search key is found!";
            flag = 1
            break;
        }
        else temp[index] = temp[index] -> next;
    }
    if (flag == 0)
        cout << "\n search key not found---";
};
```

```
void Dictionary :: delete_de (int key)
{
  int 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;
  }
  ptr[index] -> next = temp[index] -> next;
  cout << "\n" << temp[index] -> data << " has been deleted.";
  temp[index] -> data = -1;
  temp[index] = NULL;
  free (temp[index]);
}
main()
{
  int val, ch, n, num;
  char c;
  Dictionary d;
  do
  {
    cout << "\n Menu: \n1. Create";
    cout << "\n2. Search for a value \n3. Delete an value";
    cout << "\n Enter your choice:";
    cin >> ch;
    switch (ch)
    {
      case 1:
        cout << "\n Enter the number of elements to be
                inserted: ";
        cin >> n;
```

```
cout<< "\n Enter the elements to be inserted:";
for(int i=0;i<n;i++)
{
  cin>>num;
  d.insert(num);
}
 break;
case 2:
  cout<<"\n Enter the element to be searched:";
  cin>>n;
  d.search(n);
  break;
case 3;
  cout<<"\n Enter the element to be deleted:";
  cin>>n;
  d.delete_ele(n);
  break;
default:
  cout<<"\n Invalid choice";
}
cout<<"\n Enter y to continue:";
  cin>>c;
}
 while(c=='y');
 getch();
}
```

Prathsl