Nikhil. A.S
1BM18CS061

Implement Function of Dictionary Using Hashing.

```
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 Success";
            flag = 1;
            break;
        }
        else
            temp[index] = temp[index]->next;
    }
    if (flag == 0)
        cout << "\n search unsuccessful";
}

~~Destructor~~
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 (size of node);
  ptr[index] -> data = key;
  if (root [index] == NULL)
      root[index] = ptr[index];
      root[index] -> ptr next = NULL;
      temp[index] = ptr[index];
    }
    else {
      temp[index] = root[index];
      while (temp[index] * temp[index] -> next == NULL)
        temp[index] = temp[index] -> next;
      temp[index] -> next = ptr[index];
    }
  }
Void Dictionary :: delete_ele (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;
    }
    ptr[index] -> next = temp[index] -> next;
    cout << "temp[index] -> data << "deleted";
      temp[index] -> data = -1;
      temp[index] = NULL;
      free(temp[index]);
  }
```