```
struct list{
  int data;
  struct list * next;
} node_type;
node_type *ptr[max], *root[max], *temp[max];

class Dictionary{
  public:
    int index;
    Dictionary();
    void search (int);
    void insert (int);
    void delete (int);
};

Dictionary :: Dictionary(){
    index = -1;
    for(int i=0; i<max; i++){
        root[i] = NULL;
        ptr[i] = NULL;
        temp[i] = NULL;
    }
}

Dictionary :: search (int key){
    int flag = 0;
    index = int (key % max);
    temp[index] = root[index];
    while (temp[index] != NULL){
        if (temp[index]->data == key){
            // found    flag=1;
                        break;
```

```
        else temp[index] = temp[index]->next;
    }
    if (flag == 0) cout << "Not found";
}

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 :: delete (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;
    temp[index]->data = -1;
    temp[index] = NULL;
    free(temp[index]);
}
```