```
#include<iostream>
#include<stdlib.h>
using namespace std;

# define max 10

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

class Dictionary {
public:
    int index;
    Dictionary();
    void insert(int);
    void search(int);
    void del(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*) malloc(sizeof(node));
    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 << "\nSearch key found";
            flag = 1;
            break;
        } else temp[index] = temp[index] -> next;
    }
    if (flag == 0) cout<<"\nSearch key not found";
}
void Dictionary::del(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 << "\n" << temp[index] -> data << " has been deleted.";
    temp[index] -> data = -1;
    temp[index] = NULL;
    free(temp[index]);
}
int main() {
    int val, ch, n, num;
    char c;
    Dictionary d;
    do {
        cout << "\nMENU:\n1.Create";
        cout << "\n2.Search for a value\n3.Delete an value";
        cout << "\nEnter your choice:";
        cin >> ch;
        switch(ch) {
        case 1:
            cout << "\nEnter the number of elements to be inserted:";
            cin >> n;
            cout << "\nEnter the elements to be inserted:";
            for (int i = 0; i < n; i++) {
                cin >> num;
                d.insert(num);
            }
            break;
        case 2:
            cout << "\nEnter the element to be searched:";
            cin >> n;
            d.search(n);
            break;
        case 3:
            cout << "\nEnter the element to be deleted:";
```

```
            cin >> n;
            d.del(n);
            break;
        default:
            cout << "\nInvalid Choice.";
        }
        cout << "\nEnter y to Continue:";
        cin >> c;
    }
    while(c == 'y');
    return 0;



}
```

OUTPUT:

```
MENU:
1.Create
2.Search for a value
3.Delete an value
Enter your choice:1

Enter the number of elements to be inserted:5

Enter the elements to be inserted:1
2
5
2
1

Enter y to Continue:y

MENU:
1.Create
2.Search for a value
3.Delete an value
Enter your choice:2

Enter the element to be searched:2
```