

# Rajalakshmi Engineering College

Name: sri sidharthan .T

Email: 241501210@rajalakshmi.edu.in

Roll no: 241501210

Phone: 9344823470

Branch: REC

Department: I AI & ML FC

Batch: 2028

Degree: B.E - AI & ML

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

### REC\_DS using C\_Week 7\_COD\_Question 3

Attempt : 1

Total Mark : 10

Marks Obtained : 10

### Section 1 : Coding

#### 1. Problem Statement

In a messaging application, users maintain a contact list with names and corresponding phone numbers. Develop a program to manage this contact list using a dictionary implemented with hashing.

The program allows users to add contacts, delete contacts, and check if a specific contact exists. Additionally, it provides an option to print the contact list in the order of insertion.

#### *Input Format*

The first line consists of an integer  $n$ , representing the number of contact pairs to be inserted.

Each of the next  $n$  lines consists of two strings separated by a space: the name of the contact (key) and the corresponding phone number (value).

The last line contains a string *k*, representing the contact to be checked or removed.

### ***Output Format***

If the given contact exists in the dictionary:

1. The first line prints "The given key is removed!" after removing it.
2. The next *n* - 1 lines print the updated contact list in the format: "Key: *X*; Value: *Y*" where *X* represents the contact's name and *Y* represents the phone number.

If the given contact does not exist in the dictionary:

1. The first line prints "The given key is not found!".
2. The next *n* lines print the original contact list in the format: "Key: *X*; Value: *Y*" where *X* represents the contact's name and *Y* represents the phone number.

Refer to the sample outputs for the formatting specifications.

### ***Sample Test Case***

Input: 3

Alice 1234567890

Bob 9876543210

Charlie 4567890123

Bob

Output: The given key is removed!

Key: Alice; Value: 1234567890

Key: Charlie; Value: 4567890123

### ***Answer***

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#define MAX_CONTACTS 50
```

```
struct Contact {
```

```
    char name[11];
    char phone[15];
};

void printContacts(struct Contact contacts[], int size) {
    for (int i = 0; i < size; i++) {
        printf("Key: %s; Value: %s\n", contacts[i].name, contacts[i].phone);
    }
}
```

```
int removeContact(struct Contact contacts[], int *size, char *key) {
    for (int i = 0; i < *size; i++) {
        if (strcmp(contacts[i].name, key) == 0) {
            for (int j = i; j < *size - 1; j++) {
                contacts[j] = contacts[j + 1];
            }
            (*size)--;
            printf("The given key is removed!\n");
            return 1;
        }
    }
    printf("The given key is not found!\n");
    return 0;
}
```

```
int main() {
    int n;
    scanf("%d", &n);

    if (n < 2 || n > MAX_CONTACTS) {
        printf("Invalid number of contacts.\n");
        return 1;
    }

    struct Contact contacts[MAX_CONTACTS];

    for (int i = 0; i < n; i++) {
        scanf("%s %s", contacts[i].name, contacts[i].phone);
    }

    char key[11];
```

```
scanf("%s", key);  
removeContact(contacts, &n, key);  
  
printContacts(contacts, n);  
  
return 0;  
}
```

**Status :** Correct

**Marks :** 10/10