

Rajalakshmi Engineering College

Name: Haripreeth CJ
Email: 241501065@rajalakshmi.edu.in
Roll no: 241501065
Phone: 9445359004
Branch: REC
Department: I AI & ML FA
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

```
// You are using GCC
```

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

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

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

```
typedef struct contact{
```

```
    char name[20],phone[20];
```

```

    struct contact*next;
} contact;
contact*head = NULL,*tail = NULL;
void insert(char *name, char *phone){
    contact *newNode = (contact*)malloc(sizeof(contact));
    strcpy(newNode->name,name);
    strcpy(newNode->phone,phone);
    newNode->next=NULL;
    if(!head) head = tail = newNode;
    else tail = tail->next = newNode;
}
int delete1(char *key){
    contact *cur = head, *prev = NULL;
    while(cur){
        if(!strcmp(cur->name,key)){
            if(prev) prev->next=cur->next;
            else head = cur->next;
            if(cur == tail) tail = prev;
            free(cur);
            return 1;
        }
        prev = cur;
        cur= cur->next;
    }
    return 0;}
void printlist(){
    for(contact *cur = head; cur;cur=cur->next)
        printf("Key: %s; Value: %s\n",cur->name,cur->phone);
}
int main(){
    int n;
    scanf("%d",&n);
    char name[20],phone[20],key[20];
    for(int i =0; i<n;i++){
        scanf("%s %s",name,phone);
        insert(name,phone);
        scanf("%s",key);
        if(delete1(key)) printf("The given key is removed!\n");
        else printf("The given key is not found!\n");
        printlist();
        return 0;
    }
}

```

Status : Correct

Marks : 10/10