

Rajalakshmi Engineering College

Name: JAGADISH S A

Email: 241501071@rajalakshmi.edu.in

Roll no: 241501071

Phone: 9245831133

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

```
#include <iostream>
```

```
#include <unordered_map>
```

```
#include <list>
```

```
#include <algorithm>
```

```
using namespace std;
```

```

class Dictionary {
private:
    unordered_map<string, string> dictionary;
    list<string> insertionOrder;

public:
    void insertKeyValuePair(const string& key, const string& value) {
        dictionary[key] = value;
        insertionOrder.push_back(key);
    }

    void removeKeyValuePair(const string& key) {
        dictionary.erase(key);
        insertionOrder.remove(key);
    }

    bool doesKeyExist(const string& key) {
        return dictionary.count(key) > 0;
    }

    void printDictionary() {
        for (const auto& key : insertionOrder) {
            cout << "Key: " << key << "; Value: " << dictionary[key] << endl;
        }
    }
};

```

```

int main() {
    Dictionary dict;

    int numPairs;
    cin >> numPairs;

    string key, value;
    for (int i = 0; i < numPairs; i++) {
        cin >> key >> value;
        dict.insertKeyValuePair(key, value);
    }

    cin >> key;

```

```
bool keyExists = dict.containsKey(key);  
if (keyExists) {  
    cout << "The given key is removed!" << endl;  
    dict.removeKeyValuePair(key);  
    dict.printDictionary();  
} else {  
    cout << "The given key is not found!" << endl;  
    dict.printDictionary();  
}  
  
return 0;  
}
```

Status : Correct

Marks : 10/10