Rajalakshmi Engineering College

Name: SANJAY V
Email: 241801247@rajalakshmi.edu.in
Roll no: 241801247
Phone: 7397492247
Branch: REC
Department: I AI & DS FD
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 7_COD_Question 4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Develop a program using hashing to manage a fruit contest where each fruit is assigned a unique name and a corresponding score. The program should allow the organizer to input the number of fruits and their names with scores.

Then, it should enable them to check if a specific fruit, identified by its name, is part of the contest. If the fruit is registered, the program should display its score; otherwise, it should indicate that it is not included in the contest.

Input Format

The first line consists of an integer N, representing the number of fruits in the contest.

The following N lines contain a string K and an integer V, separated by a space, representing the name and score of each fruit in the contest.

The last line consists of a string T, representing the name of the fruit to search for.

Output Format

If T exists in the dictionary, print "Key "T" exists in the dictionary.".

If T does not exist in the dictionary, print "Key "T" does not exist in the dictionary.".

Refer to the sample outputs for the formatting specifications.

Sample Test Case

Input: 2
banana 2
apple 1
Banana

Output: Key "Banana" does not exist in the dictionary.

Answer

```
#include <stdio.h>
#include <string.h>
#define MAX 15

typedef struct {
   char key[21];
   int value;
   int occupied;
} HashEntry;

int hashFunc(const char *key, int size) {
   int sum = 0;
   for (int i = 0; key[i] != '\0'; i++) {
      sum += key[i];
   }
   return sum % size;
```

```
}

void insert(HashEntry table[], int size, const char *key, int value) {
    int idx = hashFunc(key, size);
    while (table[idx].occupied == 1) {
        if (strcmp(table[idx].key, key) == 0) {
            table[idx].value = value;
            return;
        }
        idx = (idx + 1) % size;
    }
    strcpy(table[idx].key, key);
    table[idx].value = value;
    table[idx].occupied = 1;
}

int search(HashEntry table[], int size, const char *key) {
    int idx = hashFunc(key, size);
    int start = idx;
    while (table[idx].occupied != 0) {
        if (table[idx].occupied == 1 && strcmp(table[idx].key, key) == 0) {
            return idx;
        }
        idx = (idx + 1) % size;
        if (idx == start)
            break;
    }
    return -1;
}

int main() {
    int n;
    scanf("%d", &n);
    HashEntry table[MAX];
    for (int i = 0; i < MAX; i++) {
        table[i].occupied = 0;
    }
    char key[21];
    int value;

    for (int i = 0; i < n; i++) {
        scanf("%s %d", key, &value);
        insert(table, MAX, key, value);
    }

    char T[21];
    scanf("%s", T);
    int pos = search(table, MAX, T);

    if (pos != -1) {
        printf("Key \"%s\" exists in the dictionary.\n", T);
    } else {
        printf("Key \"%s\" does not exist in the dictionary.\n", T);
    }

    return 0;
}
```

Status : Correct ***Marks : 10/10***