Rajalakshmi Engineering College

Name: Shremathi K
Email: 240701504@rajalakshmi.edu.in
Roll no:
Phone: 8870649491
Branch: REC
Department: I CSE FE
Batch: 2028
Degree: B.E - CSE

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 SIZE 101

typedef struct {
    char name[20];
    int score;
    int is_active;
} Fruit;

Fruit table[SIZE];

int hash(char *key) {
    int sum = 0;
    for (int i = 0; key[i]; i++)
```

```
        sum += key[i];
    return sum % SIZE;
}

void insert(char *name, int score) {
    int index = hash(name);
    while (table[index].is_active && strcmp(table[index].name, name) != 0)
        index = (index + 1) % SIZE;
    strcpy(table[index].name, name);
    table[index].score = score;
    table[index].is_active = 1;
}

int search(char *name) {
    int index = hash(name);
    int start = index;
    while (table[index].is_active || strlen(table[index].name) != 0) {
        if (table[index].is_active && strcmp(table[index].name, name) == 0)
            return 1;
        index = (index + 1) % SIZE;
        if (index == start)
            break;
    }
    return 0;
}

int main() {
    int n, score;
    char name[20], target[20];
    scanf("%d", &n);
    for (int i = 0; i < SIZE; i++)
        table[i].is_active = 0;
    for (int i = 0; i < n; i++) {
        scanf("%s %d", name, &score);
        insert(name, score);
    }
    scanf("%s", target);
    if (search(target))
        printf("Key \"%s\" exists in the dictionary.\n", target);
    else
        printf("Key \"%s\" does not exist in the dictionary.\n", target);
    return 0;
```

}

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