Rajalakshmi Engineering College

Name: Parameswari P
Email: 240701378@rajalakshmi.edu.in
Roll no: 240701378
Phone: 9500133836
Branch: REC
Department: I CSE FD
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

```
// You are using GCC
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#define SIZE 31
typedef struct {
    char name[20];
    int score;
    int occupied;
} Fruit;

Fruit hash_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 (hash_table[index].occupied) {
        index = (index + 1) % SIZE;
    }
    strcpy(hash_table[index].name, name);
    hash_table[index].score = score;
    hash_table[index].occupied = 1;
}
int search(char *name, int *score) {
    int index = hash(name);
    int start = index;
    while (hash_table[index].occupied) {
        if (strcmp(hash_table[index].name, name) == 0) {
            *score = hash_table[index].score;
            return 1;
        }
        index = (index + 1) % SIZE;
        if (index == start) break;
    }
    return 0;
}

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

    char name[20];
    int score;

    for (int i = 0; i < N; i++) {
        scanf("%s %d", name, &score);
        insert(name, score);
    }

    char search_name[20];
    scanf("%s", search_name);
    int found_score;
```

```
    if (search(search_name, &found_score)) {
        printf("Key \"%s\" exists in the dictionary.\n", search_name);
    } else {
        printf("Key \"%s\" does not exist in the dictionary.\n", search_name);
    }

    return 0;
}
```

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