Rajalakshmi Engineering College

Name: DHARINI BALA MURUGAN .
Email: 241501044@rajalakshmi.edu.in
Roll no: 241501044
Phone: 8754111345
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 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 <stdlib.h>
#include <string.h>

#define MAX_FRUITS 15
#define MAX_NAME_LENGTH 20

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

typedef struct {
    Fruit fruits[MAX_FRUITS];
    int count;
} FruitContest;
```

```
void initializeFruitContest(FruitContest *contest) {
    contest->count = 0; // Initialize the count of fruits to 0
}

int addFruit(FruitContest *contest, const char *name, int score) {
    if (contest->count >= MAX_FRUITS) {
        return -1; // Contest is full
    }
    strcpy(contest->fruits[contest->count].name, name);
    contest->fruits[contest->count].score = score;
    contest->count++;
    return 0; // Success
}

int findFruit(FruitContest *contest, const char *name) {
    for (int i = 0; i < contest->count; i++) {
        if (strcmp(contest->fruits[i].name, name) == 0) {
            return i; // Return the index if found
        }
    }
    return -1; // Not found
}

int main() {
    FruitContest contest;
    initializeFruitContest(&contest);

    int N;
    scanf("%d", &N);
    for (int i = 0; i < N; i++) {
        char name[MAX_NAME_LENGTH];
        int score;
        scanf("%s %d", name, &score);
        addFruit(&contest, name, score);
    }

    char searchName[MAX_NAME_LENGTH];
    scanf("%s", searchName);

    int index = findFruit(&contest, searchName);
    if (index != -1) {
```

```
        printf("Key \"%s\" exists in the dictionary.\n", searchName);
    } else {
        printf("Key \"%s\" does not exist in the dictionary.\n", searchName);
    }

    return 0;
}
```

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