Rajalakshmi Engineering College

Name: janane jaipratha
Email: 241501072@rajalakshmi.edu.in
Roll no: 241501072
Phone: 7548851756
Branch: REC
Department: I AIML AD
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 TABLE_SIZE 15

typedef struct Fruit {
    char name[21];
    int score;
    struct Fruit* next;
} Fruit;

Fruit* hashTable[TABLE_SIZE];

unsigned int hash(char* key) {
    unsigned int hashValue = 0;
```

```
    for (int i = 0; key[i] != '\0'; i++) {
        hashValue = (hashValue * 31 + key[i]) % TABLE_SIZE;
    }
    return hashValue;
}

void insert(char* name, int score) {
    unsigned int index = hash(name);
    Fruit* newFruit = (Fruit*)malloc(sizeof(Fruit));
    strncpy(newFruit->name, name, 20);
    newFruit->name[20] = '\0';
    newFruit->score = score;
    newFruit->next = hashTable[index];
    hashTable[index] = newFruit;
}

Fruit* search(char* name) {
    unsigned int index = hash(name);
    Fruit* current = hashTable[index];
    while (current) {
        if (strcmp(current->name, name) == 0) {
            return current;
        }
        current = current->next;
    }
    return NULL;
}

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

    for (int i = 0; i < N; i++) {
        char name[21];
        int score;
        scanf("%20s %d", name, &score);
        insert(name, score);
    }

    char target[21];
    scanf("%20s", target);
```

```
    Fruit* result = search(target);
    if (result) {
        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**