Rajalakshmi Engineering College

Name: Divya darshini S
Email: 241501051@rajalakshmi.edu.in
Roll no: 241501051
Phone: 6383045036
Branch: REC
Department: I AIML 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 TABLE_SIZE 31
#define MAX_LEN 20


typedef struct Node {
    char fruit[MAX_LEN];
    int score;
    struct Node* next;
} Node;

Node* hash_table[TABLE_SIZE] = {NULL};
```

```
unsigned int hash(char* str) {
    unsigned int h = 0;
    while (*str) {
        h = (h * 31 + *str) % TABLE_SIZE;
        str++;
    }
    return h;
}


void insert(char* fruit, int score) {
    unsigned int index = hash(fruit);
    Node* new_node = (Node*)malloc(sizeof(Node));
    strcpy(new_node->fruit, fruit);
    new_node->score = score;
    new_node->next = hash_table[index];
    hash_table[index] = new_node;
}


int search(char* fruit) {
    unsigned int index = hash(fruit);
    Node* curr = hash_table[index];
    while (curr) {
        if (strcmp(curr->fruit, fruit) == 0) {
            return 1;
        }
        curr = curr->next;
    }
    return 0;
}

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

    char name[MAX_LEN];
    int score;

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

```
        insert(name, score);
    }

    char query[MAX_LEN];
    scanf("%s", query);

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

    return 0;
}
```

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