Rajalakshmi Engineering College

Name: swetha veeramani
Email: 241501261@rajalakshmi.edu.in
Roll no: 241501261
Phone: 9790907713
Branch: REC
Department: I AI & ML FC
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

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

#define MAX 20
#define TABLE_SIZE 31
#define KEY_LEN 20
typedef struct {
   char key[KEY_LEN];
   int value;
   int occupied;
} HashEntry;
unsigned long hash(char *str) {
   unsigned long hash = 5381;
   int c;
   while ((c = *str++))
```

```
        hash = ((hash << 5) + hash) + c;
    return hash % TABLE_SIZE;
}
void insert(HashEntry table[], char key[], int value) {
    unsigned long idx = hash(key);
    while (table[idx].occupied) {
        idx = (idx + 1) % TABLE_SIZE;
    }
    strcpy(table[idx].key, key);
    table[idx].value = value;
    table[idx].occupied = 1;
}
int search(HashEntry table[], char key[], int *value) {
    unsigned long idx = hash(key);
    unsigned long start = idx;
    while (table[idx].occupied) {
        if (strcmp(table[idx].key, key) == 0) {
            *value = table[idx].value;
            return 1;
        }
        idx = (idx + 1) % TABLE_SIZE;
        if (idx == start) break;
    }
    return 0;
}

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

    HashEntry table[TABLE_SIZE] = {0};

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

    char target[KEY_LEN];
    scanf("%s", target);
    int found_score;
    if (search(table, target, &found_score)) {
        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***