Rajalakshmi Engineering College

Name: Kirubashini R
Email: 241501087@rajalakshmi.edu.in
Roll no: 241501087
Phone: 9843749339
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 17
#define MAX_NAME_LEN 21

typedef struct Node {
    char key[MAX_NAME_LEN];
    int value;
    struct Node* next;
} Node;

Node* hashTable[TABLE_SIZE];

int hash(char* key) {
```

```
    int sum = 0;
    for (int i = 0; key[i]; i++) {
        sum += key[i];
    }
    return sum % TABLE_SIZE;
}

void insert(char* key, int value) {
    int idx = hash(key);
    Node* newNode = (Node*)malloc(sizeof(Node));
    strcpy(newNode->key, key);
    newNode->value = value;
    newNode->next = hashTable[idx];
    hashTable[idx] = newNode;
}

Node* search(char* key) {
    int idx = hash(key);
    Node* curr = hashTable[idx];
    while (curr) {
        if (strcmp(curr->key, key) == 0)
            return curr;
        curr = curr->next;
    }
    return NULL;
}

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

    char key[MAX_NAME_LEN];
    int value;

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

    char searchKey[MAX_NAME_LEN];
    scanf("%s", searchKey);
```

```
    Node* result = search(searchKey);
    if (result) {
        printf("Key \"%s\" exists in the dictionary.\n", searchKey);
    } else {
        printf("Key \"%s\" does not exist in the dictionary.\n", searchKey);
    }

    return 0;
}
```

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