KEGIATAN PRAKTIKUM 7
HASH TABLE

Impelementasikan program hash table berikut:

1. Hash Table dengan Chaining

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>

#define size 4
#define h(k) k%size

struct node
{
    char* key;
    char* val;
    struct node *next;
};

struct node *chain[size];

void init()
{
    int i;
    for(i = 0; i < size; i++)
        chain[i] = NULL;
}

int keyToValue(char* key){
    int n = strlen(key);
    int i;
    int radix128 = 0;

    for(i=0; i<n; i++){
        radix128 += key[n-1-i]*pow(128,i);
    }
    return radix128;
}

void insert(char* key, char* value)
{
    //create a newnode with value
    struct node *newNode = malloc(sizeof(struct node));

    newNode->next = NULL;
    newNode->key = key;
    newNode->val = value;

    //calculate hash key
    int idx = h(keyToValue(key));
```

```
    newNode->next = chain[idx];
    chain[idx] = newNode;
}

void delete(char* key){
    int idx = h(keyToValue(key));
    struct node* p = chain[idx];
    struct node* prev = p;

    while(p && strcmp(key, p->key)){
        prev = p;
        p = p->next;
    }

    if(p == NULL){
        printf("\"%s\" is not found. Nothing to delete.\n", key);
    }else if(p == prev){
        chain[idx] = p->next;
        printf("\"%s\" is successfully removed\n", key);
        free(p);
    }else{
        prev->next = p->next;
        printf("\"%s\" is successfully removed\n", key);
        free(p);
    }
};

struct node* search(char* key){
    int idx = h(keyToValue(key));
    struct node* p = chain[idx];

    while(p && strcmp(key, p->key)){
        p = p->next;
    }
    return p;
};

void print()
{
    int i;

    for(i = 0; i < size; i++)
    {
        struct node *temp = chain[i];
        printf("chain[%d]-->",i);
        while(temp)
        {
            printf("%s:%s -->",temp->key,temp->val);
            temp = temp->next;
        }
        printf("NULL\n");
    }
}

int main()
{
```

```
    //init array of list to NULL
    init();
    insert("ab","red");
    insert("bc","black");
    insert("cd","white");
    insert("12","night");
    insert("xyz","kids");

    print();

    struct node* res = search("ab");
    if(res == NULL){
        printf("Data is not found!\n");
    }else{
        printf("%s:%s\n", res->key, res->val);
    }

    delete("12");
    delete("cd");
    delete("tx");
    print();
    return 0;
}
```

2. Hash Table dengan Open Addressing Linear Probing

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>

#define SIZE 20

#define h(k) k%SIZE
#define h_probe(k,i) (h(k)+i)%SIZE

struct node{
    char* key;
    char* val;
};

struct node T[SIZE];

void init(){
    int i;
    for(i=0; i< SIZE; i++){
        T[i].key = NULL;
    }
}

int keyToValue(char* key){
    int n = strlen(key);
    int i;
    int radix128 = 0;

    for(i=0; i<n; i++){
```

```
        radix128 += key[n-1-i]*pow(128,i);
    }
    return radix128;
}

void insert(char* key, char* val){
    int idx, i;
    i = 0;
    do{
        idx = h_probe(keyToValue(key), i);
        if(T[idx].key == NULL){
            T[idx].key = key;
            T[idx].val = val;
            return;
        }
        else{
            i++;
        }
    }while(i != idx);
    fprintf(stderr, "hash table overflow!");
}

int search(char* key){
    int idx, i;
    i = 0;
    do{
        idx = h_probe(keyToValue(key), i);
        if(!strcmp(T[idx].key, key)){
            return idx;
        }
        else{
            i++;
        }
    }while(i != idx && T[idx].key != NULL);
    return -1;
}

void print(){
    int i;
    for(i=0; i<SIZE; i++){
        printf("%s:%s\n", T[i].key, T[i].val);
    }
}

int main(){
    init();
    insert("ab","red");
    insert("bc","black");
    insert("cd","white");
    insert("12","night");
    insert("xyz","kids");

    print();

    int idx = search("cd");
    if(idx != -1){
```

```
        printf("\n%s:%s\n", T[idx].key, T[idx].val);
    }
    else{
        printf("Data is not found\n");
    }
    return 0;
}
```

Sesuai dengan kelompok Tugas Akhir yang telah terbentuk sebelumnya, kerjakan tugas-tugas berikut!

1. Jelaskan bagaimana fungsi insert pada program 1 bekerja!
2. Jelaskan bagaimana fungsi search pada program 1 bekerja!
3. Jelaskan bagaimana fungsi delete pada program 1 bekerja!
4. Jelaskan bagaimana fungsi keyToValue bekerja! Apa kegunaan dari fungsi tersebut?
5. Jelaskan mengenai fungsi hash yang digunakan pada program 1!
6. Jelaskan bagaimana fungsi insert pada program 2 bekerja!
7. Jelaskan bagaimana fungsi search pada program 2 bekerja!
8. Jelaskan mengenai fungsi hash dan probe yang digunakan pada program 2!
9. Jelaskan bagaimanakah cara untuk mengimplementasikan fungsi delete pada program 2!
10. Tuliskan jawaban Anda pada fail dengan nama: **NamaKelompok_laporan_Praktikum_7.pdf**. Jangan lupa tuliskan daftar nama dan NIM anggota kelompok Anda dalam fail tersebut!
11. Jawaban dikumpulkan selambatnya pada hari **Minggu malam**, **12 Maret 2023, pukul 23.59.**

===SELAMAT BEKERJA===