**Lab 9**

**Given a File of N employee records with a set K of Keys(4-digit) which uniquely determine the records in file F. Assume that file F is maintained in memory by a Hash Table (HT) of m memory locations with L as the set of memory addresses (2-digit) of locations in HT. Let the keys in K and addresses in L are integers. Design and develop a Program in C that uses Hash function H: K -> L as H(K)=K mod m (remainder method), and implement hashing technique to map a given key K to the address space L.  
Resolve the collision (if any) using linear probing.**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define TABLE\_SIZE 100

#define KEY\_LENGTH 5

#define MAX\_NAME\_LENGTH 50

#define MAX\_DESIGNATION\_LENGTH 50

struct Employee {

char key[KEY\_LENGTH];

char name[MAX\_NAME\_LENGTH];

char designation[MAX\_DESIGNATION\_LENGTH];

float salary;

};

struct HashTable {

struct Employee\* table[TABLE\_SIZE];

};

int hash\_function(const char\* key, int m) {

int sum = 0;

for (int i = 0; key[i] != '\0'; i++) {

sum += key[i];

}

return sum % m;

}

void insert(struct HashTable\* ht, struct Employee\* emp) {

int index = hash\_function(emp->key, TABLE\_SIZE);

while (ht->table[index] != NULL) {

index = (index + 1) % TABLE\_SIZE;

}

ht->table[index] = emp;

}

struct Employee\* search(struct HashTable\* ht, const char\* key) {

int index = hash\_function(key, TABLE\_SIZE);

while (ht->table[index] != NULL) {

if (strcmp(ht->table[index]->key, key) == 0) {

return ht->table[index];

}

index = (index + 1) % TABLE\_SIZE;

}

return NULL;

}

int main() {

struct HashTable ht;

struct Employee\* emp;

char key[KEY\_LENGTH];

FILE\* file;

char filename[100];

char line[100];

for (int i = 0; i < TABLE\_SIZE; i++) {

ht.table[i] = NULL;

}

printf("Enter the filename containing employee records: ");

scanf("%s", filename);

file = fopen(filename, "r");

if (file == NULL) {

printf("Error opening file.\n");

return 1;

}

while (fgets(line, sizeof(line), file)) {

emp = (struct Employee\*)malloc(sizeof(struct Employee));

sscanf(line, "%s %s %s %f", emp->key, emp->name, emp->designation, &emp->salary);

insert(&ht, emp);

}

fclose(file);

printf("Enter the key to search: ");

scanf("%s", key);

emp = search(&ht, key);

if (emp != NULL) {

printf("Employee record found with key %s:\n", emp->key);

printf("Name: %s\n", emp->name);

printf("Designation: %s\n", emp->designation);

printf("Salary: %.2f\n", emp->salary);

} else {

printf("Employee record not found for key %s\n", key);

}

for (int i = 0; i < TABLE\_SIZE; i++) {

if (ht.table[i] != NULL) {

free(ht.table[i]);

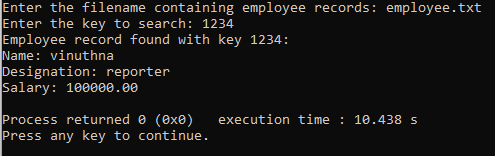
}

}

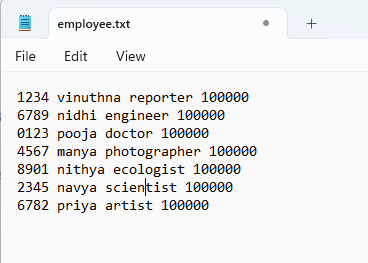
return 0;

}

**Output:**

****

**File:**

****