LAB-10

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: $K \rightarrow L$ 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.

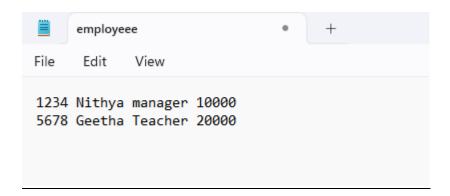
SOURCE CODE:

```
#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);
    // Print other details as needed
  } 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;
}</pre>
```

OUTPUT:



C:\Users\Admin\Desktop\hashing1.exe

```
Enter the filename containing employee records: employeee.txt
Enter the key to search: 1234
Employee record found with key 1234:
Name: Nithya
Designation: manager
Salary: 10000.00
```

C:\Users\Admin\Desktop\hashing1.exe

```
Enter the filename containing employee records: employeee.txt
Enter the key to search: 5678
Employee record found with key 5678:
Name: Geetha
Designation: Teacher
Salary: 20000.00
```