

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 → L** as $H(K) = K \bmod m$ (**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 SIZE 10
typedef struct
{
    int id;
    char name[20];
}EMPLOYEE;
EMPLOYEE e[SIZE];

void initialize_table()
{
    for(int i=0; i<SIZE; i++)
    {
        e[i].id=0;
    }
}

void insert_table()
{
    int i, id, index, hvalue;
    char name[26];
    printf("Enter the employee id and name: ");
    scanf("%d %s", &id, name);
    hvalue= id % SIZE;
    for(i=0; i<SIZE; i++)
    {
        index=(hvalue+i) % SIZE;
        if(e[index].id==0)
        {
            e[index].id=id;
            strcpy(e[index].name,name);
            break;
        }
    }
    if(i==SIZE)
    {
        printf("Hash table full\n");
    }
}
```

void display_table()

```
{
    printf("H\t Id\t Name\n");
    for(int i=0; i<SIZE; i++)
    {
        printf("%d\t %d\t %s\n", i, e[i].id, e[i].name);
    }
}
```

void main()

```
{
    int ch=0, id;
    char name[26];
    initialize_table();
    while(ch<4)
    {
        printf("1:Insert\t 2:Display\t 3:Exit\n");
        printf("Enter the choice:");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:insert_table( );
            break;
            case 2:display_table();
            break;
            case 3: exit(0);
            break;
            default: printf("Enter valid choice\n");
            break;
        }
    }
}
```