0 1 3 2

L.

LAB-10

Given a file of N employee records with a set K of keys (10-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 H.T. 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>
#define MAX 20
int hashTable[MAX];
int m;
void insert(int key)
{ int index = key % m;
  if (hasTable[index] == -1)
  {
    hasTable[index] = key
  }
  else
  {
    int i = 1;
    while (hashTable[(index+i) % m] != -1)
    {
      i++;
    }
    hashTable[(index+i) % m] = key;
  }
}
```

~~LAB 10~~

```
void display()
{   printf("\n HashTable:\n");
    for(int i=0; i<m; i++)
    {   if (hashTable[i] != -1)
            printf("Address %d : %d\n", i, hashTable[i]);
        else printf("Address %d : Empty\n", i);
    }
}
int main()
{   int n, key;
    printf("Enter size of hashtable(m): ");
    scanf("%d", &m);
    printf("Enter number of employee records: ");
    scanf("%d", &n);
    for(int i=0; i<m; i++)
        hashTable[i] = -1;
    printf("Enter %d employee keys (4 digits):\n", n);
    for(int i=0; i<n; i++)
    {   scanf("%d", &key);
        insert(key);
    }
    display();
    return 0;
}
```

O/P

```
Enter size of hashtable(m): 10
Enter number of employee records: 6
Enter 6 employee keys (4-digit):
1230
1267
1357
1789
1999
1555
HashTable:
Address 0 : 1230
Address 1 : 1999
Address 2 : Empty
Address 3 : Empty
Address 4 : Empty
Address 5 : 1555
Address 6 : Empty
Address 7 : 1267
Address 8 : 1357
Address 9 : 1789
```

MG 22/12/22