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 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 (hashTable [index] == -1 )
{
    hashTable [index] = key;
}
else
{
    int i = 1;
    while (hashTable [(index+i) %m] != -1)
    {
        i++;
    }
    hashTable [ (index +i ) %m] = key;
}
}
void display ()
{
 printf ("\n Hash Table : \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 hash table (m);");
 scanf (" %d ", &m);
```

```
printf ("Entre number of employee
           record :");
scanf (" %d", &n);

for (int i = 0 ; i<m ; i++)
    hash Table [i] = -1;

printf (" Enter %d employee keys
          (4-digit) : \n", n);
for (int i = 0 ; i<n ; i++)
 {
    scanf (" %d ", &key);
    insert (key);
  }
  display ();
  return 0;
 }.
```

output:

```
Enter size of hash table (m): 10

Enter number of employee records : 5
Enter 5 employee keys (4-digit) :
1234
92345
 3456
 4567
  5678
```

Hash Table

```
Address 0: Empty
Address 1: Empty
Address 2: Empty
Address 3: Empty
Address 4: 1234
Address 5: 2345
Address 6: 3456
Address 7: 4567
Address 8: 5678
Address 9: Empty
```

MG
22/12/25