12…#include <stdio.h>

```
#define MAX 20
int ht[MAX];
int m;

/* Hash Function */
int hash(int key)
{
    return key % m;
}

/* Insert using Linear Probing */
void insert(int key)
{
    int index = hash(key);
    int startIndex = index;

    while (ht[index] != -1)
    {
        index = (index + 1) % m;

        if (index == startIndex)
        {
            printf("Hash Table is Full. Cannot insert %d\n", key);
            return;
        }
    }
    ht[index] = key;
}

/* Display Hash Table */
void display()
{
    int i;
    printf("\nHash Table Contents:\n");
    for (i = 0; i < m; i++)
    {
        if (ht[i] == -1)
            printf("HT[%d] --> EMPTY\n", i);
        else
            printf("HT[%d] --> %d\n", i, ht[i]);
    }
}
int main()
```

```
{
    int n, key, i;
    printf("Enter size of Hash Table (m): ");
    scanf("%d", &m);

    /* Initialize hash table */
    for (i = 0; i < m; i++)
        ht[i] = -1;

    printf("Enter number of employee records (N): ");
    scanf("%d", &n);

    for (i = 0; i < n; i++)
    {
        printf("Enter 4-digit employee key: ");
        scanf("%d", &key);
        insert(key);
    }
    display();
    return 0;
}
/*output
Enter size of Hash Table (m): 10
Enter number of employee records (N): 5
Enter 4-digit employee key: 1234
Enter 4-digit employee key: 2345
Enter 4-digit employee key: 3456
Enter 4-digit employee key: 4567
Enter 4-digit employee key: 1244

Hash Table Contents:
HT[0] --> EMPTY
HT[1] --> EMPTY
HT[2] --> EMPTY
HT[3] --> EMPTY
HT[4] --> 1234
HT[5] --> 2345
HT[6] --> 3456
HT[7] --> 4567
HT[8] --> 1244
HT[9] --> EMPTY
```