

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  $H(K)=K \text{ mod } m$  (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. simple c program.

```
#include <stdio.h>
```

```
#define MAX 100
```

```
int hashTable[MAX];
```

```
int m;
```

```
/* Hash function */
```

```
int hashFunction(int key) {
```

```
    return key % m;
```

```
}
```

```
/* Insert key into hash table using linear probing */
```

```
void insert(int key) {
```

```
    int index = hashFunction(key);
```

```
    /* Linear probing for collision resolution */
```

```
    while (hashTable[index] != -1) {
```

```
        index = (index + 1) % m;
```

```
}
```

```
    hashTable[index] = key;
```

```
}
```

```
/* Display hash table */
```

```
void display() {
```

```
    int i;
```

```
    printf("\nHash Table:\n");
```

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

OUTPUT:

```
Enter size of hash table (m): 4
Enter number of employee records: 4
Enter 4 employee keys (4-digit):
22
18
45
50

Hash Table:
Address 0 : 50
Address 1 : 45
Address 2 : 22
Address 3 : 18

Process returned 0 (0x0)  execution time : 27.381 s
Press any key to continue.
```