15. Hashing using Linear Probing

Aim:

To implement hashing using linear probing.

Algorithm:

1. Define hash table with size m.
2. Use hash function key % m.
3. If collision → move linearly until empty slot found.
4. Insert/search elements accordingly.

Code:

```
#include <stdio.h>
#define SIZE 10

int hashTable[SIZE];

void insert(int value) {
    int key = value % SIZE;
    int i = 0;
    while (hashTable[(key + i) % SIZE] != 0)
        i++;
    hashTable[(key + i) % SIZE] = value;
}

void display() {
    for (int i = 0; i < SIZE; i++)
        printf("%d ", hashTable[i]);
    printf("\n");
}

int main() {
    insert(5);
```

```
    insert(15);
    insert(25);
    insert(35);
    printf("Hash Table: ");
    display();
    return 0;
}
```

Sample Output:

```
Hash Table: 0 0 0 0 0 5 15 25 35 0


=== Code Execution Successful ===
```

Result:

Hashing with linear probing was successfully implemented.