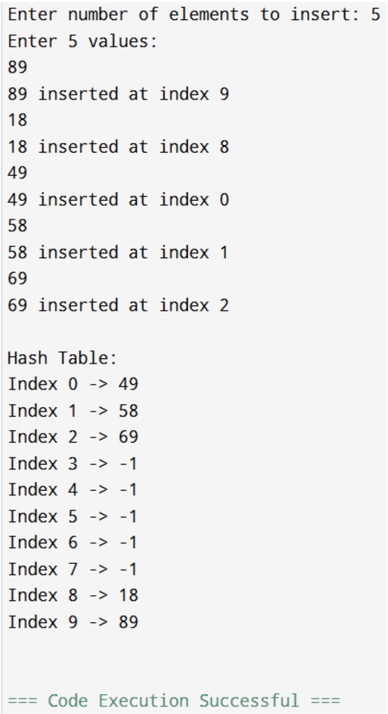
Experiment 23: Hashing – Linear Probing

Code:

```
#include <stdio.h>
#define SIZE 10
int hashTable[SIZE];
void init() {
    for (int i = 0; i < SIZE; i++)
        hashTable[i] = -1;
}
int hash(int key) {
    return key % SIZE;
}
void insert(int key) {
    int index = hash(key);
    int i = 0;
    while (i < SIZE) {
        int newIndex = (index + i) % SIZE;
         if (hashTable[newIndex] == -1) {
            hashTable[newIndex] = key;
            printf("%d inserted at index %d\n", key, newIndex);
            return;
        }
        i++;
    }
    printf("Hash Table Full! Cannot insert %d\n", key);
}
void display() {
    printf("\nHash Table:\n");
    for (int i = 0; i < SIZE; i++)
        printf("Index %d -> %d\n", i, hashTable[i]);
}
int main() {
    int n, key;
    init();
    printf("Enter number of elements to insert: ");
    scanf("%d", &n);
    printf("Enter %d values:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &key);
        insert(key);
    }
    display();

    return 0;
}
```

Output:



```
Enter number of elements to insert: 5
Enter 5 values:
89
89 inserted at index 9
18
18 inserted at index 8
49
49 inserted at index 0
58
58 inserted at index 1
69
69 inserted at index 2

Hash Table:
Index 0 -> 49
Index 1 -> 58
Index 2 -> 69
Index 3 -> -1
Index 4 -> -1
Index 5 -> -1
Index 6 -> -1
Index 7 -> -1
Index 8 -> 18
Index 9 -> 89


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