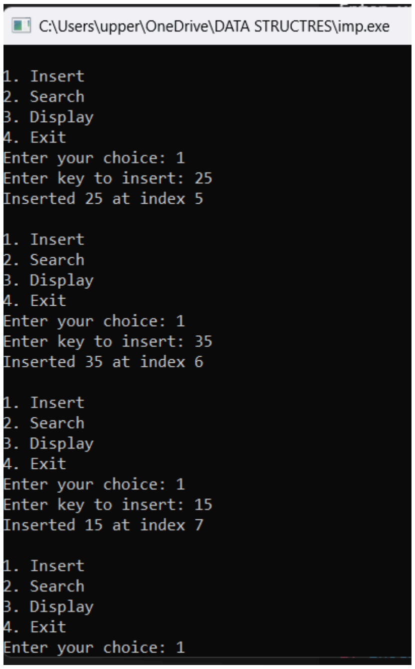
```
#include <stdio.h>
#define SIZE 10
int hashTable[SIZE];
void initialize() {
    for (int i = 0; i < SIZE; i++) {
        hashTable[i] = -1;
    }
}
int hash(int key) {
    return key % SIZE;
}
void insert(int key) {
    int index = hash(key);
    for (int i = 0; i < SIZE; i++) {
        int newIndex = (index + i) % SIZE;
        if (hashTable[newIndex] == -1) {
            hashTable[newIndex] = key;
            printf("Inserted %d at index %d\n", key, newIndex);
            return;
        }
    }
    printf("Hash table is full. Cannot insert %d\n", key);
}
int search(int key) {
    int index = hash(key);
    for (int i = 0; i < SIZE; i++) {
        int newIndex = (index + i) % SIZE;
        if (hashTable[newIndex] == key) {
            return newIndex;
        }
        if (hashTable[newIndex] == -1) {
            return -1;
        }
    }
    return -1;
}
void display() {
    printf("\nHash Table:\n");
    for (int i = 0; i < SIZE; i++) {
        printf("Index %d: %d\n", i, hashTable[i]);
    }
}
int main() {
    int choice, key;
    initialize();
    while (1) {
        printf("\n1. Insert\n2. Search\n3. Display\n4. Exit\nEnter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter key to insert: ");
                scanf("%d", &key);
                insert(key);
                break;
            case 2:
                printf("Enter key to search: ");
                scanf("%d", &key);
                int result = search(key);
                if (result != -1)
                    printf("Key %d found at index %d\n", key, result);
                else
                    printf("Key %d not found in hash table\n", key);
                break;
            case 3:
                display();
                break;
            case 4:
                return 0;
            default:
                printf("Invalid choice!\n");
        }
    }
    return 0;
}
```

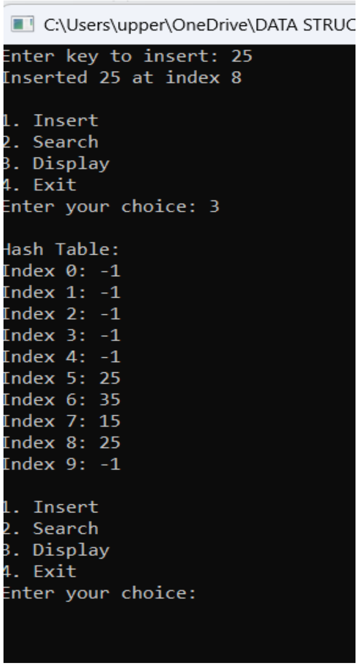


```
1. Insert
2. Search
3. Display
4. Exit
Enter your choice: 1
Enter key to insert: 25
Inserted 25 at index 5

1. Insert
2. Search
3. Display
4. Exit
Enter your choice: 1
Enter key to insert: 35
Inserted 35 at index 6

1. Insert
2. Search
3. Display
4. Exit
Enter your choice: 1
Enter key to insert: 15
Inserted 15 at index 7

1. Insert
2. Search
3. Display
4. Exit
Enter your choice: 1
```



```
Enter key to insert: 25
Inserted 25 at index 8

1. Insert
2. Search
3. Display
4. Exit
Enter your choice: 3

Hash Table:
Index 0: -1
Index 1: -1
Index 2: -1
Index 3: -1
Index 4: -1
Index 5: 25
Index 6: 35
Index 7: 15
Index 8: 25
Index 9: -1

1. Insert
2. Search
3. Display
4. Exit
Enter your choice:
```