#include <stdio.h>

#include <stdlib.h>

#define TABLE\_SIZE 10

#define EMPTY -1

int hashTable[TABLE\_SIZE];

// Initialize hash table

void initTable() {

for (int i = 0; i < TABLE\_SIZE; i++) {

hashTable[i] = EMPTY;

}

}

// Hash function

int hash(int key) {

return key % TABLE\_SIZE;

}

// Insert using linear probing

void insert(int key) {

int index = hash(key);

int startIndex = index;

while (hashTable[index] != EMPTY) {

index = (index + 1) % TABLE\_SIZE;

if (index == startIndex) { // table full

printf("Hash Table Overflow! Cannot insert %d\n", key);

return;

}

}

hashTable[index] = key;

printf("Inserted %d at index %d\n", key, index);

}

// Search using linear probing

void search(int key) {

int index = hash(key);

int startIndex = index;

while (hashTable[index] != EMPTY) {

if (hashTable[index] == key) {

printf("Key %d found at index %d\n", key, index);

return;

}

index = (index + 1) % TABLE\_SIZE;

if (index == startIndex) break; // searched whole table

}

printf("Key %d not found!\n", key);

}

// Display hash table

void display() {

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

for (int i = 0; i < TABLE\_SIZE; i++) {

if (hashTable[i] != EMPTY)

printf("%d → %d\n", i, hashTable[i]);

else

printf("%d → EMPTY\n", i);

}

}

int main() {

int choice, key;

initTable();

while (1) {

printf("\n--- Hashing Menu ---\n");

printf("1. Insert\n2. Search\n3. Display\n4. Exit\n");

printf("Enter 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);

search(key);

break;

case 3:

display();

break;

case 4:

exit(0);

default:

printf("Invalid choice! Try again.\n");

}

}

return 0;

}