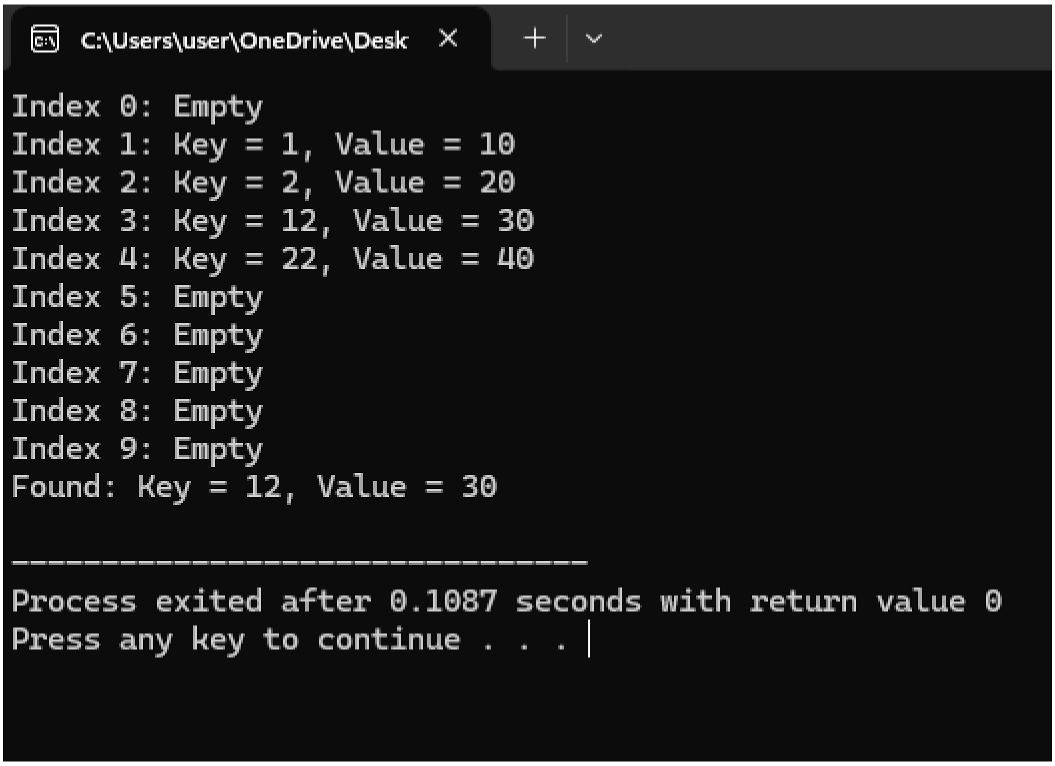
```c
#include <stdio.h>
#include <stdlib.h>

#define TABLE_SIZE 10

typedef struct {
    int key;
    int value;
} HashEntry;

HashEntry* hashTable[TABLE_SIZE];

int hashFunction(int key) {
    return key % TABLE_SIZE;
}

void insert(int key, int value) {
    int index = hashFunction(key);
    while (hashTable[index] != NULL) {
        index = (index + 1) % TABLE_SIZE;
    }
    hashTable[index] = (HashEntry*)malloc(sizeof(HashEntry));
    hashTable[index]->key = key;
    hashTable[index]->value = value;
}

HashEntry* search(int key) {
    int index = hashFunction(key);
    while (hashTable[index] != NULL) {
        if (hashTable[index]->key == key) {
            return hashTable[index];
        }
        index = (index + 1) % TABLE_SIZE;
    }
    return NULL;
}

void display() {
    for (int i = 0; i < TABLE_SIZE; i++) {
        if (hashTable[i] != NULL) {
            printf("Index %d: Key = %d, Value = %d\n", i, hashTable[i]->key, hashTable[i]->value);
        } else {
            printf("Index %d: Empty\n", i);
        }
    }
}

int main() {
    insert(1, 10);
    insert(2, 20);
    insert(12, 30);
    insert(22, 40);

    display();

    HashEntry* entry = search(12);
    if (entry != NULL) {
        printf("Found: Key = %d, Value = %d\n", entry->key, entry->value);
    } else {
        printf("Key not found.\n");
    }

    return 0;
}
```



```
Index 0: Empty
Index 1: Key = 1, Value = 10
Index 2: Key = 2, Value = 20
Index 3: Key = 12, Value = 30
Index 4: Key = 22, Value = 40
Index 5: Empty
Index 6: Empty
Index 7: Empty
Index 8: Empty
Index 9: Empty
Found: Key = 12, Value = 30

--------------------------------
Process exited after 0.1087 seconds with return value 0
Press any key to continue . . .
```