—--

CODE:

```c
#include <stdio.h>

#include <stdlib.h>

#define TABLE_SIZE 11

struct KeyValue {

    int key;

    int value;

};

int hash(int key) {

    return key % TABLE_SIZE;

}

struct KeyValue* createHashTable() {

    struct KeyValue* table = (struct KeyValue*)malloc(TABLE_SIZE * sizeof(struct KeyValue));

    for (int i = 0; i < TABLE_SIZE; ++i) {

        table[i].key = -1;

        table[i].value = 0;

    }

    return table;

}
```

```c
void insert(struct KeyValue* table, int key, int value) {

    int index = hash(key);

    while (table[index].key != -1) {

        index = (index + 1) % TABLE_SIZE;

    }

    table[index].key = key;

    table[index].value = value;

}
int search(struct KeyValue* table, int key) {

    int index = hash(key);

    while (table[index].key != -1) {

        if (table[index].key == key) {

            return table[index].value;

        }

        index = (index + 1) % TABLE_SIZE;

    }

    return -1;

}
int main() {

    struct KeyValue* hashTable = createHashTable();

    insert(hashTable, 42, 100);

    printf("Value for key 42: %d\n", search(hashTable, 42));

    free(hashTable);

    return 0;}
```