

Your tests should be written as a single .c file separate from the body of text containing your functionality to be tested. A simple example might look something like this:

---

```
1  /* keep this include at the very top of the file */
2  #include "simple-test.h"
3
4  /* any global variables, functions, other inclusions, etc.
5   * should be declared here */
6  #include "header_with_stuff_to_be_tested.h"
7
8  BEGIN_TEST
9
10 /* the string is a description of the test being run */
11 TEST("check add()'s return value")
12 {
13     int var1=2;
14     int var2=4;
15
16     /* add is a function included from our hypothetical
17      * header_with_stuff_to_be_tested */
18     EXPECT_INT(var1+var2, add(var1, var2));
19 }
20
21 TEST("compare two arrays of strings")
22 {
23     int i;
24     char array1[][10] = {
25         "str1",
26         "str2",
27         "str3",
28     };
29     char array2[][10] = {
30         "str1",
31         "str2",
32         /* matching will fail here */
33         "different",
34     };
35
36     for(i = 0; i < sizeof(array1) / sizeof(char[10]); i++) {
37         /* ECHO can be used to print (with pretty
38          * formatting) the current state within the
39          * test */
40         ECHO("checking strs at i == %i", i);
41         EXPECT_STR_EQ(array1[i], array2[i]);
42     }
43 }
44
45 END_TEST
```

---

If the second test was omitted, the output would look like this:

---

```
a_simple_example.c
1 :: check add()'s return value
   :: success!
```

---

If the second test was included, it would look like this:

---

```
a_simple_example.c
1 :: check add()'s return value
2 :: compare two arrays of strings
   :: checking strs at i == 0...
   :: checking strs at i == 1...
   :: checking strs at i == 2...
   :: a_simple_example.c:41: fail: strings unequal
   :: array1[i] == 'str3'
   :: array2[i] == 'different'
```

---

For detailed descriptions of all available tests, see the included man page or README.md .