String Equality

This is a **pair** exercise and must be competed in your **tutorial** or **lab** with your partner.

You've probably noticed that you can't use the comparison operator, `==`, to compare strings. So, how do we compare two strings? For this activity, you'll be writing a function to do exactly that:

```
int stringsEqual (char *stringA, char *stringB);
```

It takes two strings, `stringA` and `stringB`, and, if they are element-for-element the same, it returns *TRUE*, and *FALSE* otherwise. You shouldn't ever read beyond the null-terminator of either string.

Download `stringsEqual.c`, or copy it into your current directory on a CSE system by running

```
$ cp /web/cs1511/17s2/week05/files/stringsEqual.c .
```

We've also provided some simple, `assert`-based tests to help you build your solution:

```
    assert (stringsEqual ("", "") == TRUE);
    assert (stringsEqual (" ", "") == FALSE);
    assert (stringsEqual ("", " ") == FALSE);
    assert (stringsEqual (" ", " ") == TRUE);
    assert (stringsEqual ("\n", "\n") == TRUE);
    assert (stringsEqual ("This is 17 bytes.", "") == FALSE);
    assert (stringsEqual ("", "This is 17 bytes.") == FALSE);
    assert (stringsEqual ("This is 17 bytes.", "This is 17 bytes.") == TRUE);
    assert (stringsEqual ("Here are 18 bytes!", "This is 17 bytes.") == FALSE);
```

You should write more tests to demonstrate your solution works.

To run some simple automated tests:

```
$ 1511 autotest stringsEqual
```

To run Styl-o-matic:

```
$ 1511 stylomatic stringsEqual.c
Looks good!
```

You'll get advice if you need to make changes to your code.

Submit your work with the *give* command, like so:

```
$ give cs1511 wk05_stringsEqual
```

Or, if you are working from home, upload the relevant file(s) to the `wk05_stringsEqual` activity on Give Online.