7.4 - A Little Bit of Lab6 (5 points)

1. Be sure you have completed 7.1 - Reading and Writing Files in C. You also want to have read through the 7.3 slides about `grep` - the Linux command

2. Write a C program named `grep_one.c`. Using a technique similar to the 7.1 and the examples given in the 7.1 slides, modify your code to take a **string** as the first command line argument and a **file** as the second. At the basic level, your program should **print a message if the string in the first argument appears in the file specified by the second argument**.

   Example:
   ```
   grep_one aargh lab6.txt
   The string 'aargh' appears in lab6.txt
   ```
   or
   ```
   grep_one blergh lab6.txt
   The string 'blergh' does not appear in lab6.txt
   ```

   You can take this a little farther now if you want – another assignment in this unit will ask you to modify your grep_one.c so that it prints at least one line number and some context within the line, like the actual Linux grep function. But for this assignment, simply identifying if the word is in the file or not is sufficient.

   Example:
   ```
   grep_one aargh lab6.txt
   line 6: buf[bufp++] = *p  // aargh this isn't working
   ```

   A `grep_one.c` with a helper function `showstring(char* buf, int size)` is provided for you in `/home/msarris/csci235/lab6/grep_one.c` – this will start you off and give you at least one function to help debug what is going on.

   `showstring()` will print a message on the first line, your string on the second, and numbers on the third. It will also replace `'\0'` characters (that you cannot see) with `'0'` characters that you can. (Yes, technically, that means you can't tell `\0` and `0` apart, but in practice, this shouldn't make much of a difference. It's a *debug* function.)

   You can use `strcmp`, if you wish, which is mentioned in K&R Section 7.8
   Some things to consider:
   a. Does your solution work if the string contains a substring of itself? For example, if the string is `"abac"`, does your solution find it in the string `"ababac"`?
   b. Does your solution work if a line is *very* long?
   c. Modify it to print the line(s) on which the string occurs.

Navigate to your `csci235` directory. Then `tar` and compress the lab directory with the linux command (from your `csci235` directory):

```
      tar -cvzf lab6.1.tgz lab6
```

Turn in your code to moodle in `lab6.1.tgz`
It should have:

- `grep_one.c`
- `lab6.txt` (that has your name as the driver)

There is no test or autograder for this lab – you should be able to test it on your own files and determine if it is working or not.