**Homework 0914: Shell Scripts**

Due: 14 September, noon

**Homework Assignment**

This assignment must be completed entirely in a Unix environment, either Linux or Mac OS.

Write bash programs to address the following requirements. Each bash program should have a verbose header comment block including your name and the purpose of the script. Where there is any ambiguity whatsoever about how the script works or its limitations, explain. Use meaningful, self-descriptive variable names.

1. Write a shell script that accepts a filename as its argument. The named file is assumed to be a text file. The script should save the original content to a file with the original name with “.orig” appended (without the quotes). The file should be replaced with one that has all trailing spaces on each line removed. Do not use either sed or awk in your solution. Your script should do something intelligent if the named file does not exist, and should have reasonable exit codes.
2. Write a shell script that accepts a filename as its first argument. The named file is assumed to be a csv file with fields delimited by tabs. The second argument is a column number. The purpose of the script is to calculate the sum of the values in that column and print it to standard output. You may assume that all the values, if each is a value, will be integers as opposed to floats. Do not use either sed or awk in your solution. Your script should do something intelligent if the named file does not exist, if the column does not exist, or if any line in the column does not contain a value, and should have reasonable exit codes. In all cases, consider the column count to start at 0.
3. MD5 is a hash function producing a 128-bit checksum of a collection of bytes. You can read about it [here](https://en.wikipedia.org/wiki/MD5). On Linux, the command to produce an md5 is /usr/bin/md5sum. On Macs, it is /sbin/md5. For example, on Linux the command  
   $ md5sum foo  
   prints  
   fceab221011657b8f7453d10009485f0  
   to the screen. If the contents of two files (text or binary) are identical, then the md5 hash of the two files will be identical. Thus an easy way to detect identical files is to compare their md5 hashes.  
     
   Write a shell script that accepts a directory pathname as its argument that prints the names of all files that are duplicates within that directory. If there is no argument, the current working directory is assumed. A hint that may or may not be useful: if two files have different sizes, they cannot be identical. Do not use either sed or awk in your solution. Your script should do something intelligent if the named directory does not exist, and should have reasonable exit codes.

By noon on Tuesday, 14 September, submit your script files to the [homework submission](https://borax.truman.edu/250/submit.php) page.