CS 97 Discussion 1 Practice Problems

1. What text editors/IDEs have you used before Emacs? What ideas are similar between them and Emacs? What's different? What's the most difficult part of learning Emacs so far?
    a. *Homework hint!* Struggling to remember all the commands? Print this cheatsheet out (or have it handy on a screen nearby) as you do the lab.
    b. Don't forget to run `M-x open-dribble-file` for each editing session!

2. The command `env` lists all the set environment variables in your shell. What are some environment variables you see?

3. The "Unix Philosophy" is the idea that programs should be small and do one thing well.
    a. What are some examples of this that we've seen in class?
    b. Have we seen any programs that don't follow the Unix philosophy?
    c. Dan Luu argues that increasing complexity is inevitable and ultimately better for users, as it gives them easier ways to accomplish tasks. Rob Pike and Brian Kernighan disagree, using the -v option of cat as an example.
    d. Who do you agree with? Why?

4. Can you ssh into `lnxsrv06.seas.ucla.edu?`
    a. You need to be connected to the UCLA VPN.

5. Run the command `man man`. What does it say?
    a. You can navigate the man page by using the arrow keys to move up/down a line. 'space' goes forward one page, 'b' goes back one page,'q' quits
    b. *Homework hint!* Programs you'll want to man include:
        i. `which`
        ii. `wget`
        iii. `cp`
        iv. `ls`
        v. `chmod`
        vi. `find`
        vii. `locale`
    c. Are man pages too long for you? There are also TL;DR pages.

6. Execute each of the following commands, and then describe what each one did (Hint: `ls` is your friend):
    a. `cd ~/Desktop`
    b. `mkdir foobar`
    c. `cd foobar`
    d. `touch silent.txt`

e. `echo “woof” > dog`

f. `echo “oink” > pig.animal`

g. `cat silent.txt`

h. `cat dog`

i. `cat pig.animal`

j. Did all of the cat commands succeed? What does this tell you about the meaning/purpose of file extensions (.txt, .pdf, .docx, etc.)?

k. `cd ..`

l. `rmdir foobar`

m. **(Be careful you type this command exactly!)** `rm -rf foobar`

Shell Scripting

Shell Scripting Hints:

a. Remember to always start your scripts with the line “#!/bin/bash”.

b. If you try to run a shell script but you get the error “Permission denied”, make sure that you have given that file execution permissions by running:
`chmod +x <file-name>`
BN: Don’t just give execution permissions to any file! Before you execute any script, always make sure that you know who wrote it and what it does!

7. Write a shell script named “hello.sh” that, when run, outputs “Hello, world!”

8. Write a shell script named “first.sh” that outputs whatever its first argument is (empty if no arguments are given).

9. The following shell script prints out the number of visible files in the current directory:

   count-files.sh:

```
#!/bin/bash

FILES=`ls`
COUNT=0
for FILE in $FILES
do
  # echo "Found file: $FILE"
  let COUNT++
done

echo "There are $COUNT visible files in this directory."
```

   Can you use the above script to write a new one named “list-files.sh” that behaves exactly as the `ls` command (invoked with no arguments), except it prints out files separated by spaces instead of tabs, and terminates its output with a newline?

Hint: By default, echo always adds a newline character to the end of its output. You’ll need to figure out how to suppress this functionality.