CS 2750 **System Programming and Tools** Test 1
Name: Spring 2018 Max Pts: 56

Important: This is an open book test. You can use any books, notes, or paper but no electronic device. *Do not log into the computer during the test, or use any electronic or communications device. Change your cell phones to silent mode.* Any calculations and rough work can be done on the back side of the test pages. If there is a syntax error in any program segment, just write it down and you will get full credit for the problem. Please write legibly; if I cannot read what you wrote, I'll give you a zero. You will lose five points for not writing your name.

1. [6 pt] What is the difference between a system call and a library function?

   A system call is specific to an operating system and is used to make requests to kernel. They allow a user to modify the kernel data structures by assuming privileged mode of execution. A library function is a wrapper for a system call and enables portability of code. System calls are exemplified by `open` and `close` while the corresponding library functions are `fopen` and `fclose`.

2. [6 pt] In bash, can you issue two separate commands on one line? Can you extend a single command into more than one line? Give examples of how to do it?

   You can issue multiple commands by using command separator semicolon; for example

   ```
   cmd1 ; cmd2
   ```

   You can extend a single command into multiple lines by using backslash at the end of the line, such as

   ```
   cmd \
   parameter
   ```

3. [6 pt] You are given two files: `file1` and `file2`. How will you copy lines 11 to 15 from `file2` just after line 20 in `file1` using named buffers in vi/vim?

   ```
   vi file2 file1
   11G
   "a5yy
   :n
   20G
   "ap
   ```

4. [6 pt] While working in bash, we can generate a sequence of numbers by using a brace expansion or by using the `seq` command. What is the difference between those two? Why will you prefer one over the other?

   Braces are a shell builtin and are very efficient. They are limited in the sense that they do not allow the use of variables. `seq` is an external utility and invoked by the shell as an application making it slower. `seq` allows the use of variables to specify the sequence.

5. [6+4 pt] When I use the `find` command to look for a file in the system, I get a bunch of errors on the directories for which I do not have permission to search. What will you do to remove those error messages? Illustrate the command for a search for file `foobar` starting at the root directory such that no error messages are seen.

   You can get rid of the errors by sending the standard output to `/dev/null`. Example is given by

   ```
   find / -name foobar 2> /dev/null
   ```

6. [6 pt] What is the difference between creating a subshell of commands using parenthesis, such as `( cd xyz; ls )` and just grouping together the commands by curly braces, such as `{ cd xyz; ls; }`?

   The parenthesis create a sub shell; any changes made inside the sub shell are not carried back into the calling shell. Braces execute the command within the same shell and any changes (such as variable value change) are maintained as the groups of commands is done.

7. [8 pt] Write a set of commands to ask user for a number between 1 and 1000. If the input number is larger than 500, print a message saying that the number is more than 500; else print that the number is less than 500, unless the number is equal to 500. In the last case, print `You hit it.`

```
read -p "Enter a number between 1 and 1000: " number
if [ $number -gt 500 ]
then
    echo "The number is more than 500"
elif [ $number -lt 500 ]
then
    echo "The number is less than 500"
else
    echo "You hit it."
fi
```

8. [8 pt] Write a regular expression to search for a dollar amount between `$0.00` and `$999.99`. Thus, your expression should match `$0.05`, `$42.84`, `$234.87`, but should not match `$1,234.76` or `$12.345` or `123.45`.

```
/\$[[:digit:]]\{1,3\}\.[[:digit:]]\{2\}\>/
```