

Subject: ECE 3822: Exam No. 1

From: Joseph Picone <joseph.picone@gmail.com>

Date: 2/13/19, 8:29 AM

To: "ECE 3822: Facebook" <temple.engineering.ece3822@groups.facebook.com>, "ECE 3822: Google" <temple-engineering-ece3822@googlegroups.com>

To submit your solution to Exam No. 1, reply to this email, change the addressee to the class instructor – joseph.picone@gmail.com – and attach three files:

- (1) p01.sh: Your solution to problem no. 1
- (2) p02.txt: Your solution to problem no. 2
- (3) p03.txt: Your solution to problem no. 3

Any deviations from this will result in a grade of 0.

=====

(40 pts) Problem No. 1:

The find command, when run using the following arguments, produces a list of files:

```
nedc_999_[1]: find /data/courses/ece_3822/current/eeg_reports/ | head -10
/data/courses/ece_3822/current/eeg_reports/
/data/courses/ece_3822/current/eeg_reports/04_tcp_le_a
/data/courses/ece_3822/current/eeg_reports/04_tcp_le_a/031
/data/courses/ece_3822/current/eeg_reports/04_tcp_le_a/031/00003100
/data/courses/ece_3822/current/eeg_reports/04_tcp_le_a/031/00003100/s001_2006_07_25
/data/courses/ece_3822/current/eeg_reports/04_tcp_le_a/031/00003100/s001_2006_07_25
/00003100_s001.txt
/data/courses/ece_3822/current/eeg_reports/04_tcp_le_a/030
/data/courses/ece_3822/current/eeg_reports/04_tcp_le_a/030/00003010
/data/courses/ece_3822/current/eeg_reports/04_tcp_le_a/030/00003010/s003_2006_07_22
/data/courses/ece_3822/current/eeg_reports/04_tcp_le_a/030/00003010/s003_2006_07_22
/00003010_s003.txt
```

This output in some cases is a directory specification and in other cases includes a directory and a filename – called a full pathname. For this data, the pathname includes various fields:

- /data/courses/ece_3822/current/eeg_reports
- a directory describing the type of data (e.g., "04_tcp_le_a")
- an index (e.g., "030")
- a patient number (e.g., "00003010")
- a session number and date ("s003_2006_07_22")
- a filename ("00003010_s003.txt")

Write a bash shell script that reads data from stdin and counts the number of lines that meet these conditions:

- (1) The index is specified as the first argument in the script
- (2) The year of the session is specified by the second command line argument.
- (3) The filename extension is specified as the third argument.

You should number each line and print out the total number of lines following this format:

```
1: ...first match...
2: ...second match...
...
total number of matches = 27
```

For example:

```
cd /data/courses/ece_3822/current/eeg_reports/
find $PWD | p01.sh 034 2015 txt
```

would produce this output (each entry is on the same line):

```
1: /data/courses/ece_3822/current/eeg_reports/01_tcp_ar/034/00003466/s003_2015_07_10
/00003466_s003.txt
total number of matches = 1
```

Note that the index selector must only match the index and not other parts of the full pathname.

Your script should work for any combination of arguments. It cannot only work for the example above. Here are more interesting test cases:

```
find $PWD | p01.sh 9 2003 txt (113 matches)
find $PWD | p01.sh 2 2016 txt (0 matches)
find $PWD | p01.sh 02 2015 txt (5 matches)
```

The last command produces this output:

```
1: /data/courses/ece_3822/current/eeg_reports/01_tcp_ar/002/00000236/s003_2015_09_01
/00000236_s003.txt
2: /data/courses/ece_3822/current/eeg_reports/01_tcp_ar/002/00000262/s003_2015_04_21
/00000262_s003.txt
3: /data/courses/ece_3822/current/eeg_reports/01_tcp_ar/102/00010206/s002_2015_07_13
/00010206_s002.txt
4: /data/courses/ece_3822/current/eeg_reports/01_tcp_ar/102/00010231/s002_2015_06_18
/00010231_s002.txt
5: /data/courses/ece_3822/current/eeg_reports/01_tcp_ar/102/00010239/s002_2015_09_14
/00010239_s002.txt
total number of matches = 5
```

Submit your code in a file `p01.sh` that contains a bash script. I need to be able to execute this script using the command `"bash p01.sh"`.

=====

(40 pts) Problem No. 2: Consider these three pieces of information associated with a directory: permissions, owner, and size. We refer to this as an ntuple (3-tuple in this case). Write a one-line Unix command that displays the unique 3-tuples only for directories in a directory tree (ignore files). For example, consider this command:

```
mycommand /data/courses/ece_3822/current/eeg_reports/01*/*010*
```

The output should be:

```
drwxr-xr-x picone 4096
```

Your command must use standard Unix commands and work entirely from the command line. It must recurse down a directory tree and explore all branches. You can use pipes but you cannot use custom scripts.

Place your command in a file `p02.txt`. Do not write a script. Just insert the command into this file.

=====

(20 pts) Problem No. 3: When you log into our Amazon AWS server using ssh, a number of processes are executed before you reach the command line. For example, when I log in from my Mac, I start from a terminal window on my Mac, execute an ssh command, and ultimately obtain a command line prompt in a terminal window:

```
| isip_027_[1]: ssh picone@54.234.94.88  
| Last login: Wed Feb 13 07:57:30 2019 from  
| wirelessnat188.wireless.temple.edu  
  
|  
|      _/ | /_/ _   _    _____ / | / (_ ) _/  
|     /  | // - \// // ____/_\  | // / |//  
|    // | /  _// _// // // // // // > <  
|   // | \^__/\^_\,/ //   \___// // |_// // |_/  
  
| Please direct any reports to tug98850@temple.edu or help@necdata.org  
| The man command, as in man rsync or man qsub, answers all questions.  
  
| nedc_999_[1]:
```

Describe the sequence of processes that are executed to log you in. Use process IDs to trace the sequence of processes that are created, and explain the function of each one of these. Be sure to trace all the way back to the state of the machine before you log in.

Be as specific as possible by showing actual process IDs and program names for your current login session.

Place your answer in a file p03.txt. Use plain text.