

CSCI 3308 Software Development Methods and Tools Fall 2018

Homework # 1

Bash Shell Scripts & Regular Expressions

Due Date: Sunday September 23 11:59 pm

Objectives

- Write a bash shell script with bash shell commands to loop through a data file
- Write a bash shell script using UNIX commands like “awk”
- Practice using Regex commands to parse text
- Gain more experience in pair-programming collaboration [optional]

Part 1 Bash Shell Scripts

Step 1:

Place the below content in a file named AthleteTimes.txt :

```
983820459 Alejandro Bannan 7978 7834 7374
392740008 Peter Smith 7074 7190 8000
395794739 Tom Franklin 8734 9023 8900
032465922 Molly Johnson 9971 9001 8462
937562834 Anna Reid 11419 11844 10901
204868393 Rosie Reid 10991 9921 9463
297573932 Fred Reid 9987 9098 8880
592384772 Enrique Parker 8580 7923 8824
033409276 Julian Parker 9794 8889 8638
```

Step 2A:

Create a bash script file with the name Times.sh

Step 2B:

Create another AWK script file with the name TimesAwk.sh

Step 3:

The above 2 files must contain scripts to:

- a) Read the contents of AthleteTimes.txt
- b) Calculate the average of the times for each record
- c) Sort the output by last name and then first name
- d) Format the output as shown in the “Report” below

Report:

```
983820459 [2659] Bannan, Alejandro
395794739 [2911] Franklin, Tom
032465922 [3323] Johnson, Molly
592384772 [2860] Parker, Enrique
033409276 [3264] Parker, Julian
937562834 [3806] Reid, Anna
297573932 [3329] Reid, Fred
204868393 [3663] Reid, Rosie
392740008 [2358] Smith, Peter
```

The first column should be the Athlete ID. The second column is the average of the three times (rounded or truncated averages are accepted) within square brackets. The third column is the Athlete last name. This is followed by a comma, space, and the first name.

Output should be sorted, first based on the last name. If the last name is the same, sort then on the first name. If the person has the same last name and first name, then sort based on the ID. All IDs are unique in the file.

Your scripts will be tested against different test data files (not just the content in AthleteTimes.txt). However, the test data files used for evaluation will be in the same format as in AthleteTimes.txt, though it may contain more or less number of lines in the file. All athletes in the test data files will have 3 times.

The objective of writing two scripts is to see that there are multiple correct solutions to such problems. One solution should use the awk tool, and the other should use bash commands (bash scripting).

Part 2 Regex

Download the Regex Practice Data from Moodle. Create RegexAnswers.sh and for each of the questions listed below, write the regex expression necessary to calculate the answer.

Hints :

- The command grep and egrep are your friends (egrep treats { } differently than grep)
- Be sure to check for word boundaries in your answers ‘\b’ where appropriate
- Pipe answers to “wc -l” to get the count

1. How many lines end with a number?
2. How many lines start with a vowel?
3. How many 9 letter (alphabet only) lines?
4. How many phone numbers are in the dataset (format: ‘____-____-____’)?
5. How many city of Boulder phone numbers (starting with 303)?
6. How many lines begin with a number and end with a vowel?
7. How many email addresses are from UC Denver? (Eg: end with UCDenver.edu)?
8. How many email addresses are in ‘first.last’ name format *and* involve someone whose first name starts with a letter in the second half of the alphabet (n-z)?

Running RegexAnswers.sh script file should output 8 lines which is the result of 'wc -l' for each regex command. If unsure of any one of the answers, use echo "0" so that the rest of your answers align in the output.

Requirements:

1. Scripts must be bash files named

- Times.sh
- TimesAwk.sh
- RegexAnswers.sh

2. At the top of your scripts, include a comment with your name (and your partner's name if you pair program).

3. For all scripts, read in the name of the data file from command-line arguments. (The file names should not be hard coded in the scripts). We will test all the three scripts with additional data files that have different names.

4. If the program is run without the filename as the command-line argument, print out the usage statement:

Usage: Times.sh filename

Or

Usage: TimesAwk.sh filename

Or

Usage: RegexAnswers.sh filename

5. A single zipped file containing all three scripts should be uploaded on Homework1 Submission Link. Only One submission is expected if you pair-program.

NOTE: If you are working alone, name the zip file using the following template:

Lastname_HW1.zip

If you are pair programming, then name the file using this template:

Lastname1_Lastname2_HW1.zip

All 3 scripts should be runnable from command line where filename is given as an argument. If a script doesn't execute or doesn't provide the right output, then points will be deducted.