# CSCE 3600: Systems Programming

## Minor Assignment 3 – gawk, sed, and bash

### Due: Sunday, February 18, 2018 at 11:59 PM

**PROGRAM DESCRIPTION:**

In this assignment, you will write gawk and sed commands as well as a bash script to accomplish certain requested functionality. Given the many powerful features of gawk and sed, you are provided with a link to a tutorial for gawk as well as sed to assist you in completing this assignment.

**Using gawk**

For help using gawk, you may find the tutorial <http://www.grymoire.com/Unix/Awk.html> useful.

1. Consider the following file called **numbers.txt**:

Numbers

-------

4

92

18

14

75

28

12

34

60

8

82

22

Each line (i.e., record) contains only one field, a positive integer. Write a one-line gawk script that prints the sum and average of all integers contained in the file. Obviously, you need to skip the first two lines that contain header information. In this file, for example, the gawk script should print: 449 37.4167

1. Consider the following file called **elems.txt**:

1,H,Hydrogen

2,He,Helium

3,Li,Lithium

4,Be,Beryllium

. . .

115,Mc,Moscovium

116,Lv,Livermorium

117,Ts,Tennessine

118,Og,Oganesson

Since the shown file is incomplete, the **elems.txt** file is provided with this assignment. This file contains a list of the elements in the periodic table. Each field in this file is separated by a comma and each record is separated by a newline character. First, write a one-line gawk script that prints out just the record number and the full name (e.g., Hydrogen) of all the elements in the list that do not contain a vowel (i.e., ‘a’, ‘e’, ‘i’, ‘o’, or ‘u’) in the element symbol (e.g., H or Mc). Note that your results should include element symbols with one or two characters.

**Using sed**

For help using sed, you may find the tutorial <http://www.grymoire.com/Unix/Sed.html> useful.

1. Consider the following file called **phone.txt**: containing some DFW area phone numbers, except that the area code (i.e., the first three digits) are backwards and do not include the parentheses around the area code as follows:

279 680-0035

412 328-2197

964 432-8492

718 780-4972

Write a one-line sed command that reverses the order of the three digits in the area code and adds the needed parentheses around the area code so that after running the appropriate sed command, the following would be output to the terminal:

(972) 680-0035

(214) 328-2197

(469) 432-8492

(817) 780-4972

1. The // character sequence is often known as C++ style or single-line comments, while the /\* … \*/ character sequence is often known as C-style or multi-line comments. As an example, assume the following **myprog.c** file that has a mix of both C-style and C++ style comments:

// This is a test of how this works

#include <stdio.h>

int main()

{

// declare some variables here

int num1 = 4;

float num2 = 3.5;

// print the result

printf("The result is %f\n", num1 \* num2); // this does it

/\* does it work? \*/

return 0;

}

Write a one-line sed command that transforms all of the C++ style (i.e., single-line comments) to the C-style (i.e., multi-line comments) so that after running the appropriate sed command, the following would be output to the terminal:

/\* This is a test of how this works \*/

#include <stdio.h>

int main()

{

/\* declare some variables here \*/

int num1 = 4;

float num2 = 3.5;

/\* print the result \*/

printf("The result is %f\n", num1 \* num2); /\* this does it \*/

/\* does it work? \*/

return 0;

}

**Writing Bash Scripts**

Write a bash script called **minor3.sh** that accepts two command-line parameters: (1) a directory name, either absolute or relative, and (2) a size in bytes. For a valid directory, you will print out the list of files along with the size (in bytes) of each file only for those files whose size is greater than or equal to the size specified by the user in the command-line.

Your script must adhere to the following requirements:

* If the user enters too few or too many arguments, you are to display an appropriate usage statement and terminate the program.
* Since the directory given in the command-line may be absolute (e.g., $HOME/csce3600/sp18) or relative (e.g., ../sp18), you must check that it is a valid directory. If the directory is not valid, you will print out an error message and terminate the program.
* You must implement and use at least one non-trivial function in your bash script that accepts at least one parameter.
* You must use at least one non-trivial sed or gawk command within your script. For use with these commands, you may wish to redirect some output from your script to a file called minor3.tmp and apply your sed and/or gawk command.
* If no files match your criteria for the size of the file, you will print out some indicator that no files matched your criteria.
* Your script should be well documented in terms of comments. For example, good comments in general consist of a header (with your name, course section, date, and brief description), comments for each variable, and commented blocks of code.
* Your script should be named “minor3.sh”, without the quotes.
* Your script will be graded based largely on whether it works correctly on the CSE machines (e.g., cse01, cse02, …, cse06), so you should make sure that your script successfully runs on a CSE machine.

**SAMPLE OUTPUT** (user input shown in **bold green**)**:**

$ **./minor3.sh**

usage: ./minor3.sh directory size

$ **./minor3.sh some\_directory**

usage: ./minor3.sh directory size

$ **./minor3.sh bad\_directory 100**

error: unable to access bad\_directory directory

$ **cd testdir**

./testdir$ **ls -l**

total 36

-rw------- 1 mat0299 mat0299 1743 Feb 11 17:02 elems.txt

-rw------- 1 mat0299 mat0299 50 Feb 11 17:02 numbers.txt

-rw------- 1 mat0299 mat0299 52 Feb 11 17:02 phone.txt

-rw------- 1 mat0299 mat0299 40 Feb 11 17:02 rec02.txt

-rw------- 1 mat0299 mat0299 2497 Feb 11 17:02 test1.c

-rw------- 1 mat0299 mat0299 2361 Feb 11 17:02 test2a.c

-rw------- 1 mat0299 mat0299 2336 Feb 11 17:02 test2.c

-rw------- 1 mat0299 mat0299 2598 Feb 11 17:02 test3.c

-rw------- 1 mat0299 mat0299 253 Feb 11 17:02 test.c

./testdir$ **cd ..**

$ **./minor3.sh testdir 100**

Files with size greater than 100:

1 elems.txt 1743

2 test1.c 2497

3 test2a.c 2361

4 test2.c 2336

5 test3.c 2598

6 test.c 253

$ **./minor3.sh testdir 10**

Files with size greater than 10:

1 elems.txt 1743

2 numbers.txt 50

3 phone.txt 52

4 rec02.txt 40

5 test1.c 2497

6 test2a.c 2361

7 test2.c 2336

8 test3.c 2598

9 test.c 253

$ **./minor3.sh testdir 2400**

Files with size greater than 2400:

1 test1.c 2497

2 test3.c 2598

$ **./minor3.sh $HOME/csce3600/sp18 2400**

Files with size greater than 2400:

1 a.out 7477

2 minor2.c 4074

3 test1.c 2497

4 test3.c 2598

5 testdir 4096

$ **./minor3.sh ../sp18 4000**

Files with size greater than 4000:

1 a.out 7477

2 minor2.c 4074

3 testdir 4096

$ **./minor3.sh ../sp18 10000**

Files with size greater than 10000:

-- NONE --

**REQUIREMENTS:**

* For the gawk and sed commands, test out your results on real files on our CSE machines (e.g., cse01, cse02, …, cse06), to make sure that they indeed work. Then, type (or copy and paste) your answers to this document that will be submitted to Canvas.
* This is an individual programming assignment that must be the sole work of the individual student. Any instance of academic dishonesty will result in a grade of “F” for the course, along with a report filed into the Academic Integrity Database.

**SUBMISSION:**

* You will electronically submit this file with your typed solutions for gawk and sed along with your bash script file minor3.sh to the **Minor 3** dropbox in Canvas by the due date and time.