

CSCE 3600: Systems Programming

Minor Assignment 1 – sed and gawk

Due: 11:59 PM on Friday, September 24, 2021

PROGRAM DESCRIPTION:

In this assignment, you will write `sed` and `gawk` commands to accomplish certain requested functionality. Given the many powerful features of `sed` and `gawk`, you are provided with links to manuals and tutorials for `sed` and `gawk` to assist you in completing this assignment.

Using sed

- a) Consider the following file called `palindrome` containing some 6-letter words, some of them palindromes (i.e., words that read the same backwards and forwards) and some not:

```
toyota
pullup
abccba
nissan
redder
```

Write a one-line `sed` command that prints out only the lines containing 6-letter palindromes so that after running the appropriate `sed` command, the following would be output to the terminal:

```
pullup
abccba
redder
```

You may write the `sed` command-line in the space provided below and ensure that this Word document (with your solutions) is submitted to Canvas.

Answer: `sed -E -n '/^(.)(.)(.)\3\2\1$/p' palindrome`

- b) Consider the following file called `phone.txt` containing some arbitrary airline phone numbers as follows:

```
(866) 879-7647
(888) 474-7424
(371) 670-6006
(866) 266-5588
(844) 415-3955
(800) 237-2747
(800) 667-2747
```

Write a complete `sed` script called `minor1.sed` that will encrypt the phone data according to the following:

1. *Preprocessing*:
 - a. Remove the parentheses from each phone number.
 - b. Remove any whitespace from each phone number.
 - c. Remove any dashes (i.e., '-') from each phone number.
2. *Transposition* (i.e., rearrange the order of individual characters):
 - a. Swap the third and eighth number in each phone number.
3. *Substitution* (i.e., replace characters by other characters):
 - a. Substitute each number in the phone number with random letters, being sure to use an even mix of upper- and lower-case letters.
4. *Append extra data*:
 - a. Append the first 10 letters of your first and last name AFTER the third phone number in the file. If your first and last names combined are shorter than 10 letters, use letters from your middle name as well.

In this file, for example, my `sed` script should print the following:

```
$ sed -r -f minor1.sed phone.txt
FSSFDGDStD
FFttDtDFet
rDqSDqSwqS
MarkThomps
FSAeSSASFF
FtGtwArtAA
FqDerDeqtD
FqDSSDeqtD
```

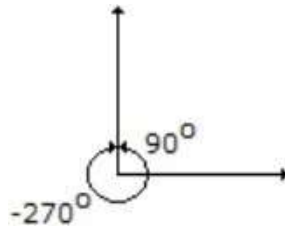
This `sed` script file will be submitted to Canvas.

Using gawk

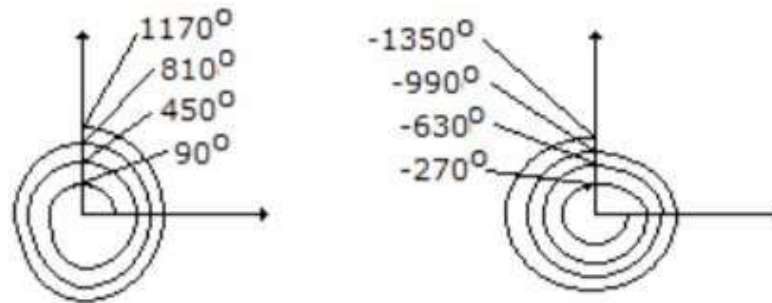
- a) Consider a list of angles in degrees, such as the following `angles.txt` file:

```
Degrees
135
90
1860
-45
0
-270
1170
-180
-795
-630
```

When two angles have the same initial and terminal sides, they are called coterminal. This means that angles of 90° and -270° are coterminal as the following figure shows:



In fact, all of the following angles shown in the figure below are coterminal:



We want to make sure that all angle values in degrees are between $0^\circ \leq x < 360^\circ$. For any file containing a list of angles in degrees with the `Degrees` header line at the top (as in `angles.txt`), write a complete `gawk` program that prints out the original value of the angle in the file as well as its coterminal value between $0^\circ \leq x < 360^\circ$.

In this file, for example, the `gawk` program should print the following:

```
$ gawk -f minor1.gawk angles.txt
135      135
90       90
1860     60
-45      315
0        0
-270     90
1170     90
-180     180
-795     285
-630     90
```

Formatting properly in columns as shown is required. This `gawk` program file will be submitted to Canvas.

b) Consider the following file called `salaries.csv`:

```
uid,loc,div,year,sal
dgarcia,WDC,fin,2017,102900
ejones,SEA,sales,2016,56450
dlarkin,RCH,rsrch,2018,78435
```

```
ccartwright,RCH,acct,2018,48775
mlesko,OTT,rsrch,2017,93870
criggs,WDC,sales,2018,62334
jpetrovski,RCH,acct,2016,68348
tchepregi,SEA,rsrch,2017,84290
amurchin,OTT,sales,2018,49038
gcarlson,RCH,fin,2018,70000
swelsh,WDC,sales,2016,39876
```

Each field in this file is separated by a comma and each record is separated by a newline character. For this file, you will write a one-line `gawk` command-line to filter records for employees who earn \$50,000 or more in 2018. Specifically, you will print out only the `userid` (i.e., `uid`), `location` (i.e., `loc`), and `salary` (i.e., `sal`) for those employees working in 2018 who earned a salary of \$50,000 or more. You may write the `gawk` command-line in the space provided below and ensure that this Word document (with your solutions) is submitted to Canvas.

Answer: `gawk 'BEGIN {FS=","} {if($4==2018 && $5>50000) printf("%-10s %-5s %d\n", $1, $2, $5)}' ./salaries.csv`

REQUIREMENTS:

- Your `sed` script and `gawk` program files should include your name and EUID at the top of the file. No other comments are needed in these files.
- 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. Your solution to the one-line `sed` script and `gawk` program can be typed (or copied and pasted) to this document and will be submitted to Canvas.
- Your program 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 program runs on a CSE machine. Please include any special instructions required to run your `sed` script and `gawk` program.
- 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 one-line solutions for `sed` and `gawk` along with your `sed` script `minor1.sed` and `gawk` program `minor1.gawk` to the **Minor 1** dropbox in Canvas by the due date and time.