Answers for Lab 5

This class and all CS courses do contain a lab component that will require the submission of an email to the instructor, or a posting in the discussion board or use the dropbox folder. Do not worry if you see a series of 10 labs in the dropbox, as explained, some labs will be submitted in a different manner and the grades will be kept in my spreadsheet. Some labs may require group participation. For the campus students, the course is labeled in the schedule as CLWEB (classroom meetings with lab component online)

Your labs will count towards 10% of your final grade. The labs are scheduled to be 3 hours long but in many cases you will probably finish the activity earlier. The labs will have due dates and once the submission period ends, you will not be able to submit it.

It is understood that a few students may not be able to be present during a lab session due to several reasons. If this is your case, you will need to submit your lab activity prior to the lab meeting. These lab meetings will be announced via email and in the "New material available Announcement" Topic section in the discussion board. Please make sure to ask questions so you do not miss out on the lab activities.

Answers for Lab 5

On the top of all your documents, always include these items:

your name:
date:
Lab number

Names:_____

You can work with ONE partner for this activity but it is no required. Upload the text file with the answer to the Dropbox under “Lab 5

Submission”

Find the answers, using awk, for the following statements; you will get extra points for each correct answer. Points will be given only if 100% correct. These awk commands are just one line long each.

DO NOT SHARE YOUR ANSWERS, THIS IS EXTRA CREDIT, THAT MEANS INDIVIDUAL EFFORT. You have 3 hours to complete this task or complete it prior to the official lab meeting and email your answers to wmorales@pcc.edu with the subject CS140U - your names - AWK activity

Hint: you may consult your notes and the Web if necessary.

You will be doing your awk searches on a file called longfile and winelist. To obtain these files use the command:

yourid@syccuxfs01:~> cp ~wmorales/awkexercise/longfile ~

and

yourid@syccuxfs01:~> cp ~wmorales/awkexercise/winelist ~

This will copy the two files that you will need for this exercise to your home directory.

(3 points - easy)

1)Using awk, write the command that would print the line that contains the word "selected". This file has a lot of junk in it, so you are just searching for the lines that contain the word selected (there will be only one). Your awk script will look into the file called longfile and once it finds all the lines that contain the word "selected" it will display the line output that should be:

output: (There should be only one line)

server, or at least to a "randomly" selected one of the servers covering

Some of the answers are close to perfect, but the idea behind this exercise it was for you to explore the power of AWK and I hope that in the process you did learn something new. Also these are some samples from past students. Some very creative answers and none of them have ever worked with AWK in the past. So, to come up with some of these answers it does take time.

YOUR AWK COMMAND:

```
awk '/selected/ { print $0 }' longfile
```

(10 points hard one)

2) Write the command that would find all lines that have an email address and place a label email = before the line in the file longfile

output will multiple lines similar to this one :

email= Message-Id: 199801232015.MAA21444@mail-gw2.pacbell.net

NOTE: this will be a long listing. The line above would not be a good example since 199801232015.MAA21444@mail-gw2.pacbell.netwould not be a valid email address. You would be searching for lines with the words from: to: Cc:, usually after these characters, there could be a good, valid e-mail address.

YOUR AWK COMMAND:

awk '(/@/ && (/[Tt]o/||/[Ff]rom/||/[Ee]-mail/)) print' longfile

awk '/^[Tt][Oo]:/ || /^[Ff][Rr][Oo][Mm]:/ || /^[Ee]-[Mm][Aa][Ii][Ll]:/ || /^[cC][cC]:/ {print "email=" $0}' longfile | more

awk '/To:/ || /From:/ || /CC:/ || /Sender:/ {print "email=" $0}' longfile

awk '(/@/ && (/[Tt]o/||/[Ff]rom/||/[Ee]-mail/)) {print "email= " $0}' longfile

grep -E -o "\b[a-zA-Z0-9.-]+@[a-zA-Z0-9.-]+\.[a-zA-Z0-9.-]+\b" longfile | awk '{ print "email = " $0 }'

awk '/cc:|CC:|bcc:|BCC:|to:|To:|TO:|from:|From:|FROM:/' longfile | awk '{ print "email = " $0 }'

gawk '/@/' longfile | gawk '/From:/ || /Cc:/ || /To:/ {print "email=" $0}'

awk '{for(i=1;i<=NF;i++) { if ($(i) ~ "@") print ("email=") $i}}' longfile

awk '/To:/ || /From:/ || /CC:/ {printf "email = " $0 "\n"}' longfile

awk '/(^To: .*\@)|(^From: .*\@)|(^Sender: .*\@)|(^CC: .*\@)/ {print "email=",$0}' longfile

awk '$0 ~ "@" {gsub ("[,<>:\[()'\"\"]" , " "); gsub ("]" , " "); print $0}' longfile | awk -v RS=" " '$0 ~ "@" {$1=$1; print "email= " $1}' | uniq

awk '/To:/ || /to:/ || /From:/ || /from:/ || /CC:/ || /cc:/ || /Cc:/' longfile | grep -i -o '[A-Z0-9._%+-]\+@[A-Z0-9.-]\+\.[A-Z]\{2,4\}'

awk '$1 !~ /Message/' longfile | awk '/To:|From:|CC:/ {print "email = ", $0}' longfile

awk -v RS='[[:alnum:]_.]+@[[:alnum:]_]+[.][[:alnum:]]+' 'RT{print RT}' longfile

awk '/[Tt]o: / || /[Fr]rom: / || /[Cc]c: /' longfile | grep -E -o "\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,6}\b" | sort | uniq | awk '{print "email = "$1}'

awk '/From:|from:|Cc:|To:|to:/ && /@/ { print "email = " $0}' longfile

awk '(/to:/ || /from:/ || /To:/ || /From:/ || /cc:/ || /CC:/ || /bcc:/ || /BCC:/) && /@/ {print "email= ", $0}' longfile

awk '/^To:|^to:|^From:|^from:|^Cc:|^cc:/ {print "email = ", $0}' longfile | sort | uniq

awk '{for (i=1;i<=NF;i++) {if ($i ~ /[[:alpha:]]@[[:alpha:]]/) {print "email= "$i }}}' longfile | sort | uniq

awk '/[Ff]rom:/&&/@/ || /[Tt]o:/&&/@/ || /[Cc]c:/&&/@/ {print "email=" $0}' longfile

awk -F '<' '/From:/ || /from:/ || /To:/ || /to:/ || /Cc:/ || /cc:/ || /CC:/ || /BCC:/ || /BCc:/ || /Bcc:/ || /bcc:/ {print $2}' longfile | awk -F '>' '/@/ {print "email=", $1}' | awk '!x[$0]++'

awk -v RS='[[:alnum:]_.]+@[[:alnum:]_]+[.][[:alnum:]]+' 'RT{print "email = " RT}' longfile

awk 'match($0, /[a-zA-Z0-9_.-]+@[a-zA-Z0-9_.]+.com/) {print "email = ",substr($0,RSTART,RLENGTH)}' longfile

awk '{for (i=1;i<=NF;i++) if ($i ~/[*@]/) if($i ~/*/) {break} else print "email = " $i}' longfile | uniq

awk 'BEGIN { ORS="\nemail= "} /[Ff]rom/ && /@/ ; /[Tt]o:/ && /@/ ; /[Cc]c:/ && /@/' ~/longfile

(3 points - easy)

3)Write the awk command that would show the user wmorales (or yourself) that is online (you have to use a pipe)

Output, depending on the system, would be similar to:

$ wmorales pts/0 192.220.32.72 4:18pm 0.00s 0.07s 0.02s w

YOUR AWK COMMAND:

$ w | awk '/wmorales/ {print $0}'

(10 points hard one) -

4) finger the users on our system that have the letter r in their first name, or last name, but nowhere else. Note also that the r can be r or R. The output of the finger command usually varies depending on the system that you are using.

Output would be similar to:

tring Tong Ring 4 church (helix.cse.ogi.edu)

tzhou Roger Ring 0:15 p2 helix (zephyr.cse.ogi.e)

zab Rachu Brown 1:40 p7 bogart (sampo.grumblesmu)

note: the letter r or R has to appear either in their first name or last name only, not on the name of the machine that they are using if it happen to have r in it.

YOUR AWK COMMAND:

```
finger | awk '$2 ~/[rR]/ print $3 ~/[rR]/ print'

finger | awk '$2 ~ /[rR]/ || $3 ~ /[rR]/ {print}'

finger | awk '$1 ~ /[rR]/ { print }'

finger | awk '/[Rr]/ && /[^Rr]\>/ && $0 !~ /(instr)/ && $0 !~ /(Fri)/'

finger | awk '$1 ~ /.*[rR]+.*\./ || $1 ~ /.*\..*[rR]+/ {print $0}'

who | awk 'BEGIN {FS = "." } ; {print $1, $2}' | awk -F '' '{OFS=""; $1=toupper($1)}1' | awk -F ' ' '{print $1" " toupper(substr($2,0,1))substr($2,2)}' | awk '$1 && $2 ~ /[rR]/ {print $1, $2}'

finger -l | gawk '/Name:.*[Rr]/'

finger -lp | awk '$6~/[Rr]/||$7~/[Rr]/ {print $6,$7}'

who | awk '($1 ~ /r/ || $1 ~ /R/) && ($5 !~ /r/ && $5 !~ /R/)'
```

5) (3 points - easy)

from the file called winelist extract a list of wines that cost less then $7.00

output (it may be different depending the time you do it):

1977 6.99 Franciscan

1977 6.75 Alexander Valley

1977 5.99 Charles Krug

YOUR AWK COMMAND:

$ awk '$1 ~ /197[56787]/ && $2 < 7.00' winelist