Ibrahim Gunsever – Sed Lab – Linux Administration – 10/11/20

FYI:

**After some experimenting I figured out if you use the -i after sed it will write the changes to the original file. I was able to have Linux make changes to the requested lines via the -i command. -i is the in-place option. Also if your afraid to make changes to the original file you can put a file extension after the -i so sed knows to save the results as a different extension and not overwrite the original file. Example sed -I .bak**

I plan on posting my labs to my portfolio, so I wanted to be able to explain what I did.

1. Change the name Jon to Jonathan .

To change the name from Jon to Jonathan in the text file:

sed -n ‘s/Jon/Jonathan/gp’ datebook

The s tells sed I want to substitute any matches to Jon with Jonathan. P tells sed to print matched lines instead of all lines in the file. The g tells sed to search the file globally. Oh and -n tells sed to suppress unmatched text. I believe its considered the quiet option.

**2.**Delete the first four lines.

sed ‘1,4d’ datebook.

The numbers inside of the parentheses tells sed I want this range of lines. For example 1,4 means lines 1 to 4. The d tells sed I want those lines deleted. And of course datebook is where I want this to occur.  
  
**3.**Print lines 7 through 11

sed -n ‘7,11p’ datebook

Just like deleting lines I told sed that I want lines 7,11. And instead of using d for delete I used p for print.

**4.**Delete lines containing Lane .

sed -n ‘/Lane/d’ datebook

This tells sed to look for all lines that contain ‘Lane’ and delete it.

**5.**Print all lines where the birthdays are in October or December .

sed -rn '/^[^:]\*:[^:]\*:[^:]\*:(10|12)\//p'

Okay this took me more time then it should but it outputted the lines of people that have bdays in October or December. Was able to figure it out using this site <https://www.geeksforgeeks.org/sed-command-in-linux-unix-with-examples/> . So ^ means the beginning of the line. And [^:] the way I placed it three times means any number of characters, in the first second and third position, oh and it ignores the :. Then the 10 | 12 means I want either 10 for October or 12 for December followed by a /. And of course p means to print. I guess instead of [^:] you could do [0-9] but for some reason it wasn’t working for me.

**6.**Append three asterisks to the end of lines starting with Fred .

sed ‘/Fred/ s/$/ \*\*\*/’ datebook

This tells sed for every line that starts with Fred put 3 asterisks at the end. Now if I wanted it to append the asterisk to lines that only have Fred in the beginning I would use a ^. So it would look like sed ‘s/^Fred/s/$/\*\*\*/’ Datebook  
  
**7.**Replace the line containing Jose with JOSE HAS RETIRED .

sed -e 's/.\*Jose.\*/JOSE HAS RETIRED\./g'

The \* act as wild cards and they allow me to replace the word ‘Jose’ with ‘Jose has retired’.

**8.**Change Popeye 's birthday to 11/14/99 . Assume you don't know Popeye's original birthday. Use a regular expression to search for it.

 sed -e '/Popeye/s#[0-9][0-9]\*:[^:]\*/[0-9][0-9]#11/14/46#g' datebook

So I tried using [^:] for the whole reg ex expression instead of [0-9] and it would not work. For some reason it only worked for the position of the date. Im not sure why It did not work. Maybe my formatting was wrong?

**9.**Delete all blank lines.

sed '/^$/d' datebook

So ^ means we are starting with and $ means the end of the pattern and d means delete. Since I put the ^$ together without giving sed any further information it will delete the spaces in the file. Now if you wanted a copy of the file without the spaces you could use the in place option and specify a different extension. For example in this case it would be sed -i .bak ‘/^$/d’ datebook. This will create a file called datebook.bak that has all the blank lines removed!

**10.**Write a sed script that will

**a.**Insert above the first line the title  - PERSONNEL FILE -.

**b.**Remove the salaries ending in  6 00 .

**c.**Print the contents of the file with the last names and first names reversed .

**d.**Append at the end of the file  THE END .

#I would create this script as 10lab.sed. Since its in a .sed format linux knows to apply sed to all #commands in the file

# in order to run this file all I would need to do is type this command

# sed -i -f 10lab.sed datebook

#this allows sed to insert the title PERSONNEL FILE above the first line in datebook

1i\PERSONNEL FILE’

#this says for every salary that ends in 600 delete it

/600$/d

#this tells sed to find the last names and first names and print them as reversed.

s/\(^[A-Za-z]\*\)\([ \t]\*\)\([A-Za-z]\*\):\(.\*\)$/\3\2\1:\4/g

#this tells sed at the end of the file add the two words THE END

$a\THE END

This is not part of the script. Struggled to figure out c. regexr.com helped a lot along with some YouTube videos I researched.