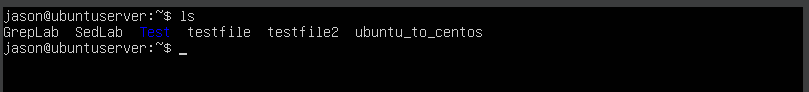
Graphical user interface

Description automatically generated



Inside of Sed File:

Text

Description automatically generated

Part 1: Change name

Command: sed s/Jo/Josephine/ SedLab (s means to substitute)

Text

Description automatically generated

Step 2: Deleting last 5 lines.

I made it a lot easier for me by deleting the blank lines prematurely as the command believes that the blank lines are lines themselves and called is ***Nospace.***

Text

Description automatically generated

Command: sed ’23,$d’ Nospace

This command essentially deletes every line after line 23. $d just means Delete and since we gave it the parameter to start at 23 it will delete everything after it.

Text

Description automatically generated

Part 3: Print lines 3-15:

Sed -n ‘3-15p’ SedLab

-n means to suppress and the P is essentially you telling the command to print these Lines.

Text

Description automatically generated

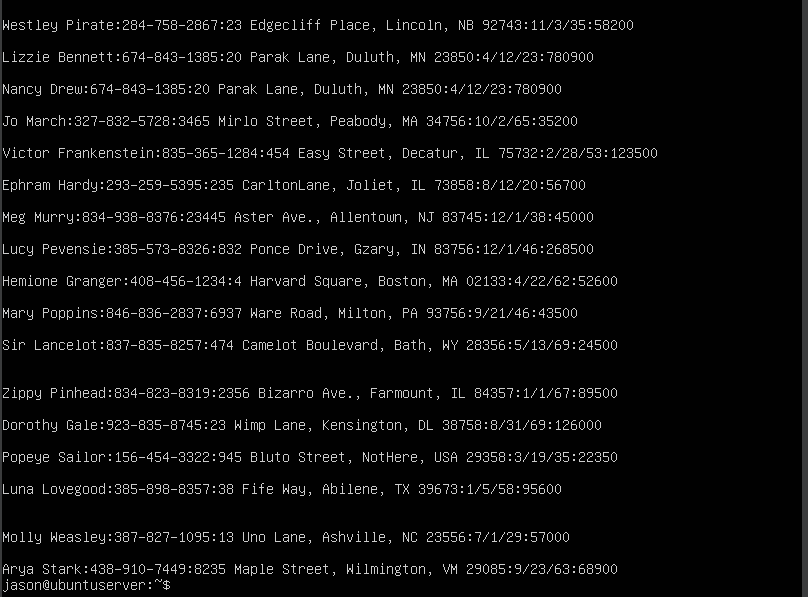
Part 4:

This part is for the deletion of lines for people who live in California:

You will have to type the following in order to remove them.

***sed -i ‘/CA/d’ SedLab***

This command is saying please edit the file “i” Look for CA (‘/CA) and delete them inside the SedLab file (/d’ SedLab)



Part 5:

This part we are given the task of printing people who have their birthday on the first week of the month.

The command will be the following:

***sed -rn ‘/:[1-12]\/[0-7]\/[0-9]{2}:/p’ SedLab***

Text

Description automatically generated

The part that’s underline is the month and since months are 1-12 that’s what I wrote. Then if you take a look at the boxed in [0-7], the reason why we did 0-7 once was because we wanted the first week of the month. Finally at the end [0-9]{2} this means it can be two numbers in the range within 0-9

Part 6:

We will have to append three asterisks to the end of every line starting with Sir.

In order to do that we would have to type the following command:

***sed ‘/^Sir/ s/$/\*\*\*/’ SedLab***

The command is telling the system to do the following:

Look for whatever starts with Sir (‘/^Sir) and append \*\*\* at the end. “S” is just specifying that you want the following to do and finally ($/\*\*\*/’) is just you append \*\*\* at the end and the $ is because of it.

Once you run that command you will get the following:

Text

Description automatically generated

The underline is where the append happened.

Part 7: We will be replacing the entire lines that start with “Westley Pirate”

The command that we will be using for this is:

***sed -I ‘s/^Westley Pirate.\*$/As you wish./’ SedLab***

This will delete the entire line that contains Westley Pirate and instead place “As you wish.”

Don’t be afraid if nothing happens you will have the type: ***cat SedLab*** to see the results. It should look something like this:

“-I” means to Edit in Place.

Just like step 6 we will be searching for anything that starts with Westley Pirate (^Westley Pirate) and wipe the rest of that line (\*$).

And to finish it off we will be replacing it with “As you wish.” Which -I does for use since it’ll edit it in place for us.

\*\*\*\* My Vm won’t allow me to page up \*\*\*\*

Text

Description automatically generated

Part 8: We will be changing Minerva McGonagall’s birthday to 10/04/1935 without any knowledge as to what their current birthday is using RegEx.

***sed -n ‘/^Minerva McGonagall.\*$/p’ SedLab | sed ‘s/:[0-9]{2}\/[0-9]{2}\/[0-9]{2}/ 10\/04\/1935/’***

The first half of the command where we see the person’s name is the part of the program where it searches for Minerva at the beginning of the line. This is done by the “^” character and then ends it at the their last name. This is done by the “\*$”.

Then we pipe the command to do another thing which so done with the “|” character.

Here everything that says [0-9] means the numbers are between 0-9 and the anything with the {} tells the command just how many times a number should appear.

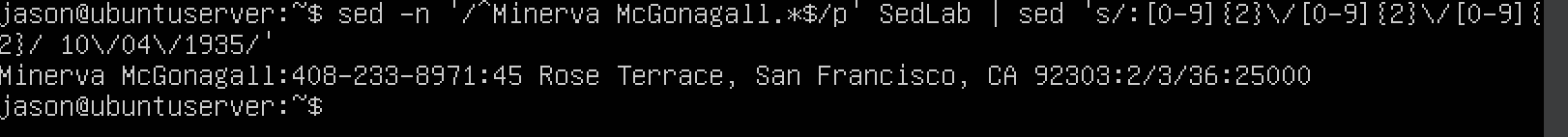
For an example /:[0-9]{2}\ means “/” begin here with that specified character “/” then I want {2} two character that fall within the range of [0-9] and it start when it sees “\”. That begins a new set of characters.

**\*\*\***

**I had to SCP the file over again since it seems like I’ve deleted her entrée by accident.**

**\*\*\***

Once you’ve hit enter you should get something like the picture provided below:



Step 9: Delete all Blank Lines

I’ve moved the file that has the lines without space on to it’s own file to make step 2 a lot easier.

This command will delete the lines that have nothing in them. The “/” indicates that the command starts here and the following [[:space:]] is the parameters for the command. Finally, the \*$/d is telling the command to delete everything till the end of the space. Please understand since we started with “^” the program will only delete the lines that start with space aka nothing.



(Source for removing space <https://www.linuxshelltips.com/remove-lines-from-file-using-sed-command/>)

What’s inside Nospace:

Text

Description automatically generated

Step 10: On this step we will be creating a script:

Few things we will need to know when creating a script.

First.) In order to create a script we will need to the do the following:

***(vi ‘scriptname’)***



When you do that it will take you to this page:

Text

Description automatically generated with medium confidence

Please be aware that you won’t be able to make any changes until you hit “I” on your keyboard. This will tell the server that you would like to “Insert” text.

You know you’ve done it correctly when you get the “—INSERT—” words on the bottom left.



In order to save you will have to hit “CTRL + C”.

This will prevent you from modifying the file. You’ve done it correctly when “—INSERT—” is no longer there. Instead, it will be blank or contain the file name.



To save the file after you’ve done the previous step just type the following:

“:WQ” This will save and quite the screen

To simply quit without saving just remove the “W”

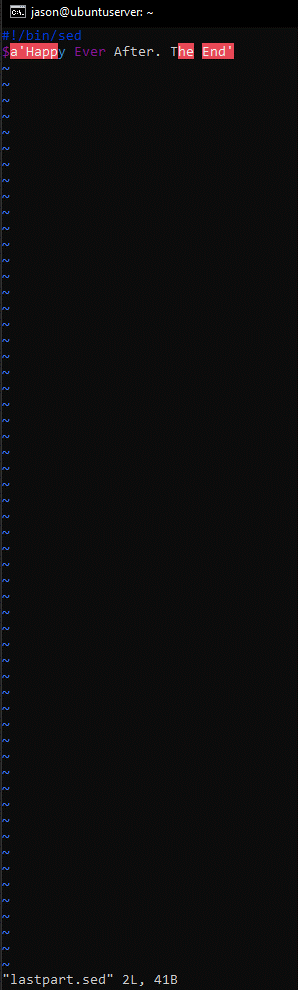
Now I will create the script that will append the following to the end of the entire list.

I wrote this command:

***{ #/bin/sed***

***$a’Happy Ever After. The End’***

***}***

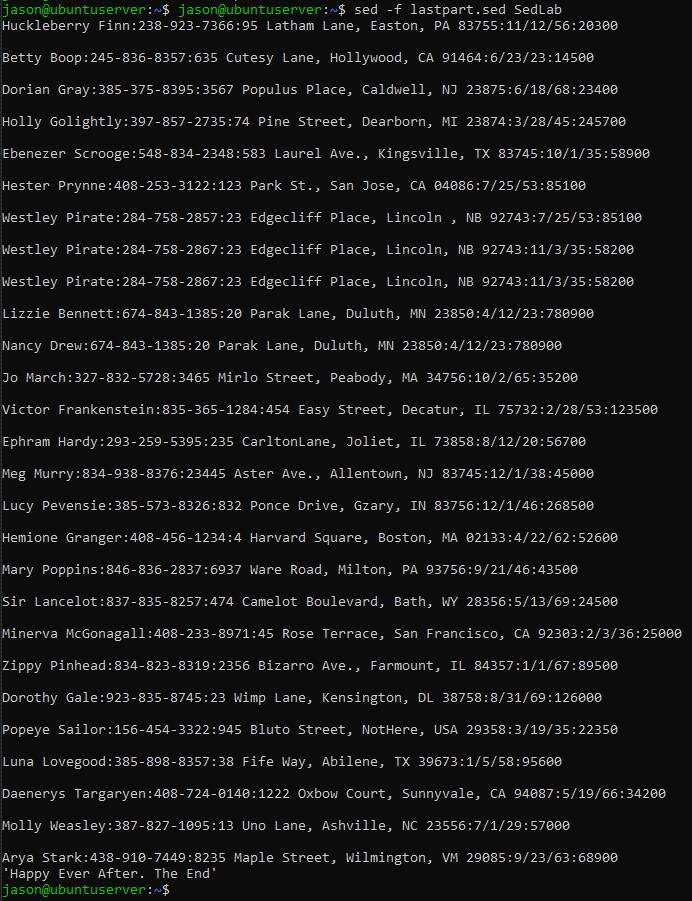


At the very end $ means “at the very end”

So at the very end this will attach the new line and to run the script please type the following:

***(sed -f lastpart.sed SedLab)***

You can make the script be one instead of what I did which was separate them.



I created another one through the “vi” command and filled it in with the following information.

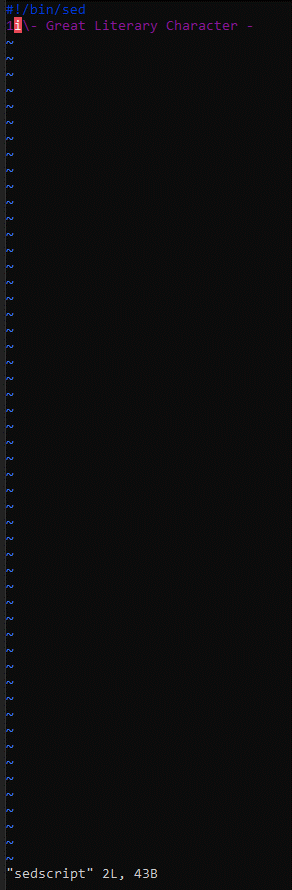
***{***

***#!/bin/sed***

***1i\- Great Literary Character –***

***}***

“1I” means the edit as edit line 1



Finally ran it with the same command as the pervious step, only difference being the name of the file.

