Lindsay Graham

Linux Administration

Week 3 Lab – Sed

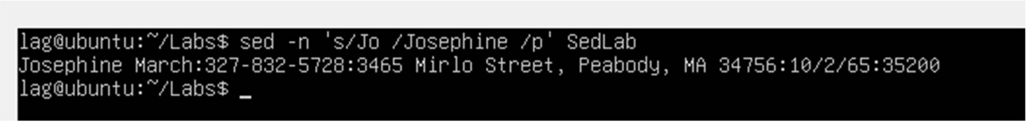
Sed Tutorial

Sed is a stream editor primarily used for substitutions. It can work with regular expressions (regex), making it a very powerful tool. By default, sed doesn't modify the original file, a safety feature that helps preserve the original. For these examples, the file used was named SedLab. For clarity, the file will be referred to as 'filename' throughout this tutorial.

Example 1. Change the name Jo to Josephine

- **sed -n ‘s/Jo /Josephine /p’ filename**

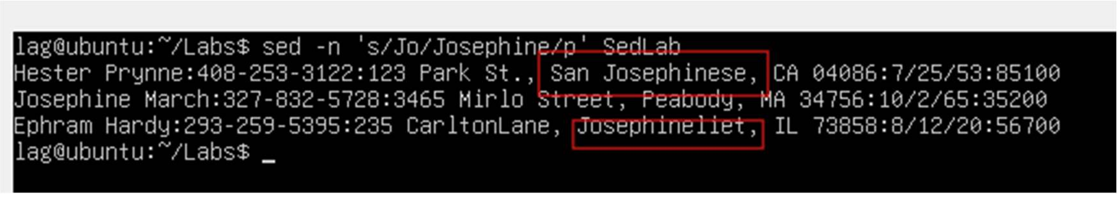
Figure 1.1 Output showing line where Jo was changed to Josephine



```
lag@ubuntu:~/Labs$ sed -n 's/Jo /Josephine /p' SedLab
Josephine March:327-832-5728:3465 Mirlo Street, Peabody, MA 34756:10/2/65:35200
lag@ubuntu:~/Labs$ _
```

The syntax for this command is as follows: sed -n 's/text to match/text to substitute/p' filename. The -n option with sed prevents it from printing anything unless explicitly requested, allowing you to focus on the desired output. The s option in sed is used for substitutions. In this example, sed was used to find "Jo" and replace it with "Josephine." The p at the end specifies to print the result to the screen. To ensure that only exact matches of "Jo" were replaced, a space was included after "Jo" and "Josephine" in the pattern.

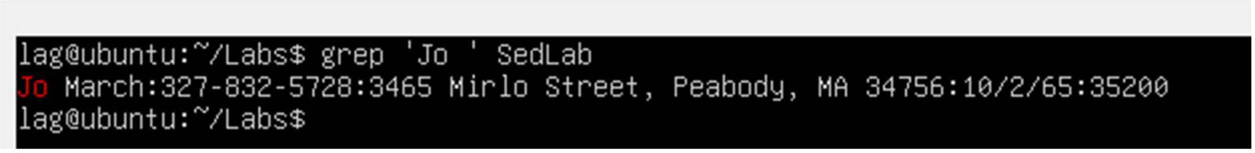
Figure 1.2 Example of results if a space had not been included in the pattern



```
lag@ubuntu:~/Labs$ sed -n 's/Jo/Josephine/p' SedLab
Hester Prynne:408-253-3122:123 Park St., San Josephinese, CA 04086:7/25/53:85100
Josephine March:327-832-5728:3465 Mirlo Street, Peabody, MA 34756:10/2/65:35200
Ephram Hardy:293-259-5395:235 CarltonLane, Josephineliet, IL 73858:8/12/20:56700
lag@ubuntu:~/Labs$ _
```

Hint: It can be helpful to use grep first as a way to check that the sed command did the correct number of substitutions.

Figure 1.3 Using grep to identify all instances of “Jo”

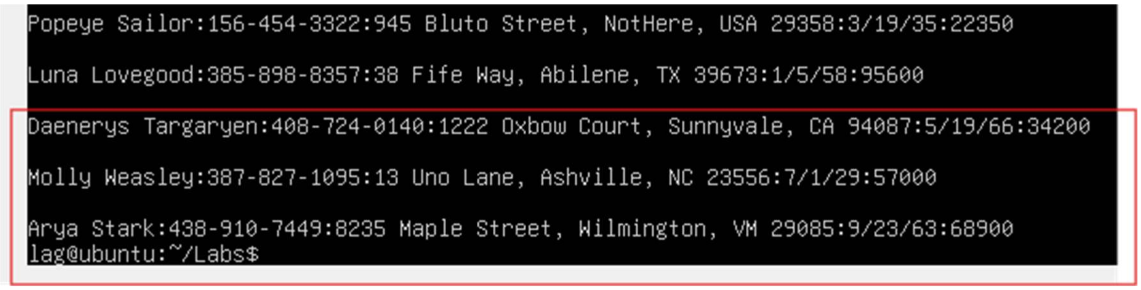


```
lag@ubuntu:~/Labs$ grep 'Jo ' SedLab
Jo March:327-832-5728:3465 Mirlo Street, Peabody, MA 34756:10/2/65:35200
lag@ubuntu:~/Labs$
```

Example 2. Delete the last 5 lines

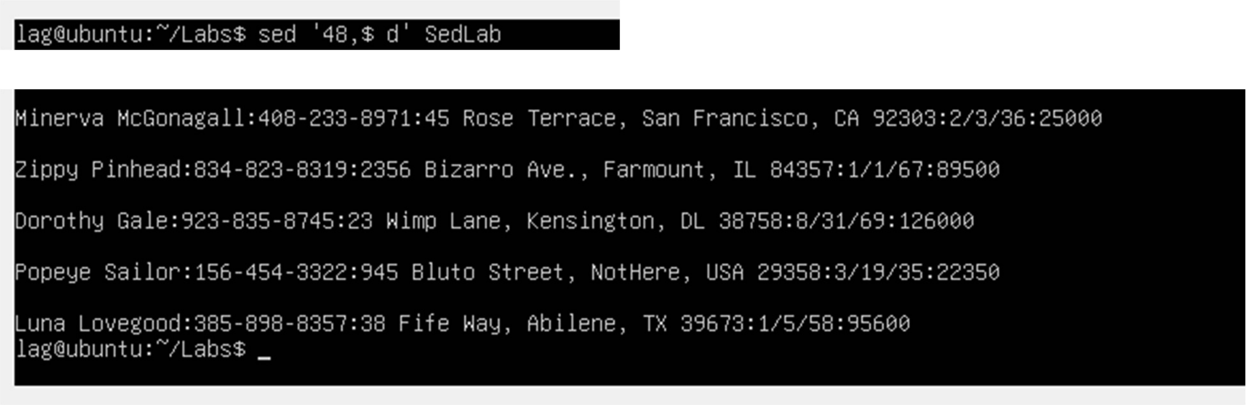
- **sed ‘48,$ d’ filename**

Figure 2.1 Output showing last fine 5 lines of file before sed



```
Popeye Sailor:156-454-3322:945 Bluto Street, NotHere, USA 29358:3/19/35:22350

Luna Lovegood:385-898-8357:38 Fife Way, Abilene, TX 39673:1/5/58:95600

Daenerys Targaryen:408-724-0140:1222 Oxbow Court, Sunnyvale, CA 94087:5/19/66:34200

Molly Weasley:387-827-1095:13 Uno Lane, Ashville, NC 23556:7/1/29:57000

Arya Stark:438-910-7449:8235 Maple Street, Wilmington, VM 29085:9/23/63:68900
lag@ubuntu:~/Labs$
```

Figure 2.2 Output showing the last lines of the file after sed

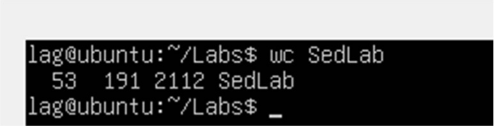


```
lag@ubuntu:~/Labs$ sed '48,$ d' SedLab
```

```
Minerva McGonagall:408-233-8971:45 Rose Terrace, San Francisco, CA 92303:2/3/36:25000

Zippy Pinhead:834-823-8319:2356 Bizarro Ave., Farmount, IL 84357:1/1/67:89500

Dorothy Gale:923-835-8745:23 Wimp Lane, Kensington, DL 38758:8/31/69:126000

Popeye Sailor:156-454-3322:945 Bluto Street, NotHere, USA 29358:3/19/35:22350

Luna Lovegood:385-898-8357:38 Fife Way, Abilene, TX 39673:1/5/58:95600
lag@ubuntu:~/Labs$ _
```

The syntax for this command is as follows: sed ‘starting line, to the end delete’ filename. In this example, the starting line was 48, the $ represented the end of the file, and the d specified the deletion. To find the starting line number, use the wc (word count) command. In this example, there were 53 lines. To print the last 5, simply subtract 5 from 53.

Figure 2.3 Output of wc (word count) command to get line count



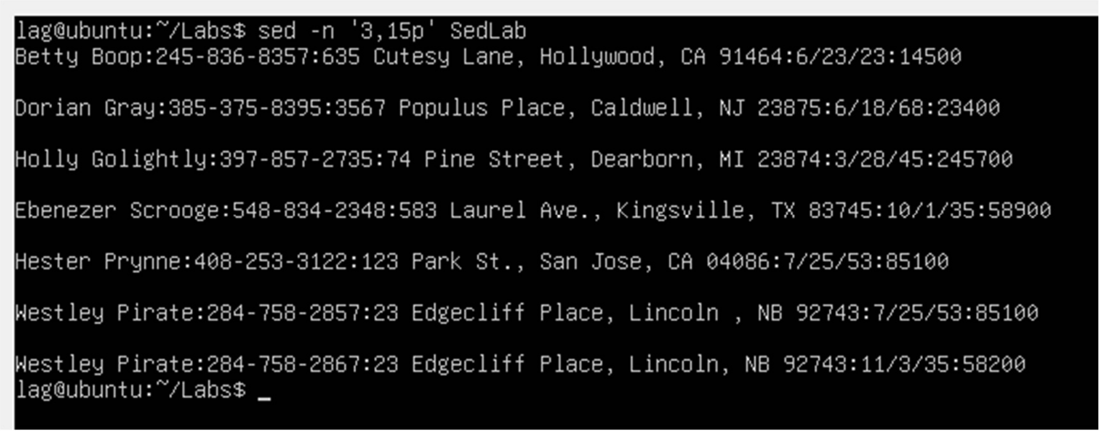
```
lag@ubuntu:~/Labs$ wc SedLab
  53  191 2112 SedLab
lag@ubuntu:~/Labs$ _
```

The results of wc show the count of lines, words, and characters in the file. Blank lines are included as lines in the count.

Example 3. Print lines 3-15

- **sed -n '3,15p' filename**

Figure 3.1 Output of lines 3-15



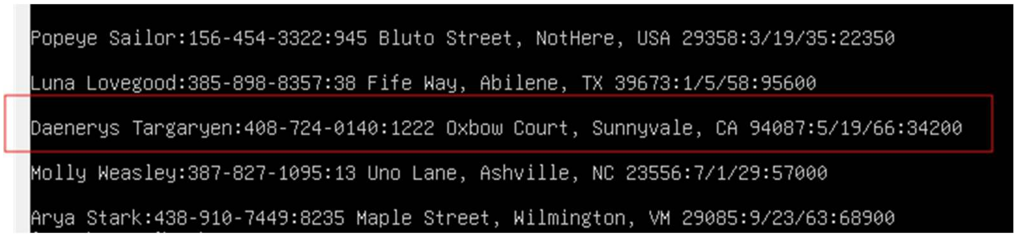
```
lag@ubuntu:~/Labs$ sed -n '3,15p' SedLab
Betty Boop:245-836-8357:635 Cutesy Lane, Hollywood, CA 91464:6/23/23:14500

Dorian Gray:385-375-8395:3567 Populus Place, Caldwell, NJ 23875:6/18/68:23400

Holly Golightly:397-857-2735:74 Pine Street, Dearborn, MI 23874:3/28/45:245700

Ebenezer Scrooge:548-834-2348:583 Laurel Ave., Kingsville, TX 83745:10/1/35:58900

Hester Prynne:408-253-3122:123 Park St., San Jose, CA 04086:7/25/53:85100

Westley Pirate:284-758-2857:23 Edgecliff Place, Lincoln , NB 92743:7/25/53:85100

Westley Pirate:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
lag@ubuntu:~/Labs$ _
```

The syntax for this command is as follows: sed -n 'line to start at, line to end print' filename. In this example, line 3 was the starting line and 15 was the ending line. The p option, used in conjunction with -n, allowed only lines 3-15 to be printed to the terminal. Blank lines are still lines and are included in the line count.

Example 4. Delete lines for people who live in California

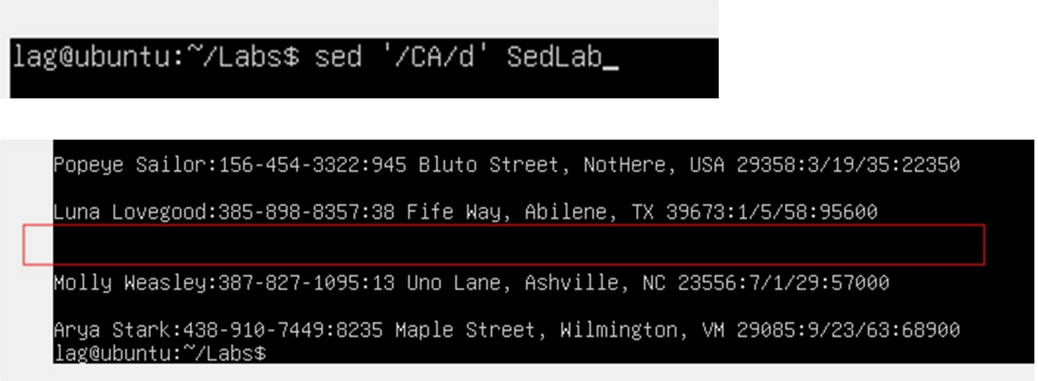
- **sed '/CA/d' filename**

Figure 4.1 Sample of results before deletion



```
Popeye Sailor:156-454-3322:945 Bluto Street, NotHere, USA 29358:3/19/35:22350

Luna Lovegood:385-898-8357:38 Fife Way, Abilene, TX 39673:1/5/58:95600

Daenerys Targaryen:408-724-0140:1222 Oxbow Court, Sunnyvale, CA 94087:5/19/66:34200

Molly Weasley:387-827-1095:13 Uno Lane, Ashville, NC 23556:7/1/29:57000

Arya Stark:438-910-7449:8235 Maple Street, Wilmington, VM 29085:9/23/63:68900
```

Figure 4.2 Sample of results after deletion



```
lag@ubuntu:~/Labs$ sed '/CA/d' SedLab_
```

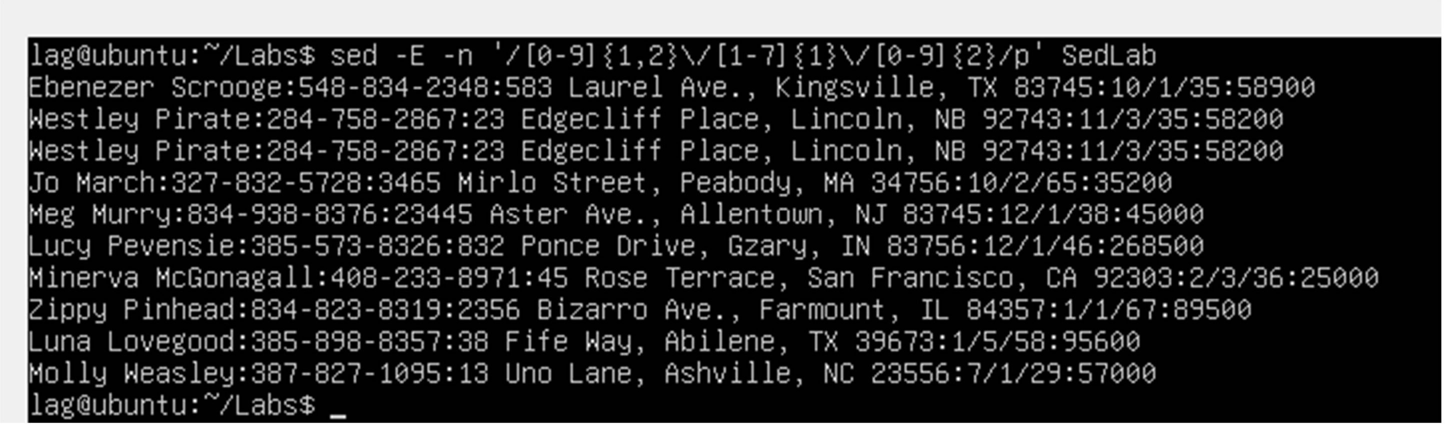
```
Popeye Sailor:156-454-3322:945 Bluto Street, NotHere, USA 29358:3/19/35:22350

Luna Lovegood:385-898-8357:38 Fife Way, Abilene, TX 39673:1/5/58:95600


Molly Weasley:387-827-1095:13 Uno Lane, Ashville, NC 23556:7/1/29:57000

Arya Stark:438-910-7449:8235 Maple Street, Wilmington, VM 29085:9/23/63:68900
lag@ubuntu:~/Labs$
```

The syntax for this command is as follows: sed '/text to find/d' filename. In this example, the command instructs sed to find all lines containing 'CA' and delete them. The first figure shows a portion of the file using the cat command. The next figure shows a portion of the file after the deletion.

Example 5.Print all lines where the birthdays are in the first week of the month

- **sed -E -n ‘/[0-9]{1,2}\/[1-7]{1}\/[0-9]{2}/p’ filename**

Figure 5.1 Output of all lines where the birthdays are in the first week of the month



```
lag@ubuntu:~/Labs$ sed -E -n '/[0-9]{1,2}\/[1-7]{1}\/[0-9]{2}/p' SedLab
Ebenezer Scrooge:548-834-2348:583 Laurel Ave., Kingsville, TX 83745:10/1/35:58900
Westley Pirate:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
Westley Pirate:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
Jo March:327-832-5728:3465 Mirlo Street, Peabody, MA 34756:10/2/65:35200
Meg Murry:834-938-8376:23445 Aster Ave., Allentown, NJ 83745:12/1/38:45000
Lucy Pevensie:385-573-8326:832 Ponce Drive, Gzary, IN 83756:12/1/46:268500
Minerva McGonagall:408-233-8971:45 Rose Terrace, San Francisco, CA 92303:2/3/36:25000
Zippy Pinhead:834-823-8319:2356 Bizarro Ave., Farmount, IL 84357:1/1/67:89500
Luna Lovegood:385-898-8357:38 Fife Way, Abilene, TX 39673:1/5/58:95600
Molly Weasley:387-827-1095:13 Uno Lane, Ashville, NC 23556:7/1/29:57000
lag@ubuntu:~/Labs$ _
```

The syntax for this command is as follows:

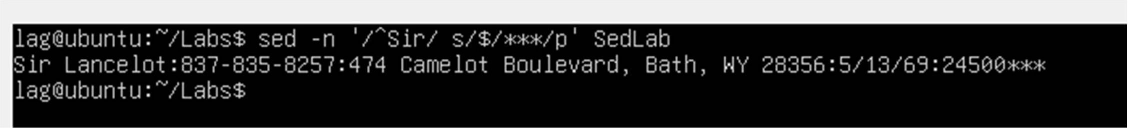
- sed
- -E: Extended regex option
- -n: suppress printing unless instructed
- [0-9]{1,2}\/[1-7]{1}\/[0-9]{2}: Regex for birthdate in format mm/dd/yy
- /p: print results
- filename

The -E option represents extended regular expressions, which are generally considered more powerful and flexible than basic regular expressions. The regular expression searches for birthdates formatted as mm/dd/yy. The [1-7]{1} part of the expression identifies birthdates in the first week of the month. Since there are 7 days in a week, and no leading zeros in the address book, [1-7]{1} ensures that only one digit between 1 and 7 is matched. Similarly, the [0-9]{1,2} part of the expression specifies that the pattern for birthdates may have one or two digits.

Example 6. Append three asterisks (*) to the end of lines starting with Sir

- **sed -n ‘/^Sir/ s/$/***/p’ filename**

File 6.1 Output showing three asterisks appended to the end of lines starting with Sir



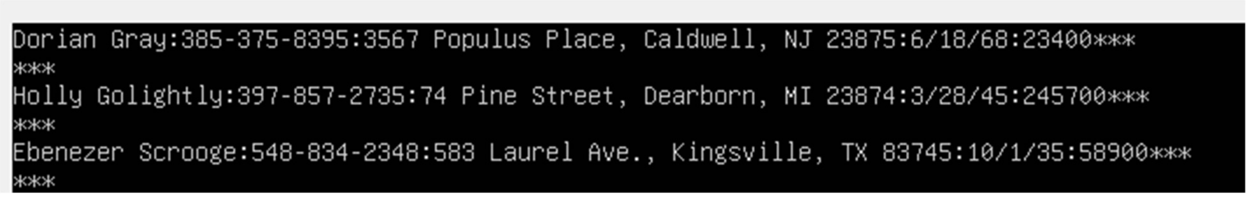
```
lag@ubuntu:~/Labs$ sed -n '/^Sir/ s/$/***/p' SedLab
Sir Lancelot:837-835-8257:474 Camelot Boulevard, Bath, WY 28356:5/13/69:24500***
lag@ubuntu:~/Labs$
```

The command syntax is as follows:

- sed
- -n: suppress printing unless instructed
- /^Sir/ : Find lines that start with Sir
- s/$/: Substitute at the end of the line
- *** : text to substitute/append
- /p : Print results
- filename

Sed searched the file for lines that start with 'Sir' and then appended the three asterisks to the end of the line. If '^Sir' wasn't specified, asterisks would have been appended to the end of all lines.

Figure 6.2 Results if '^Sir' had not been specified

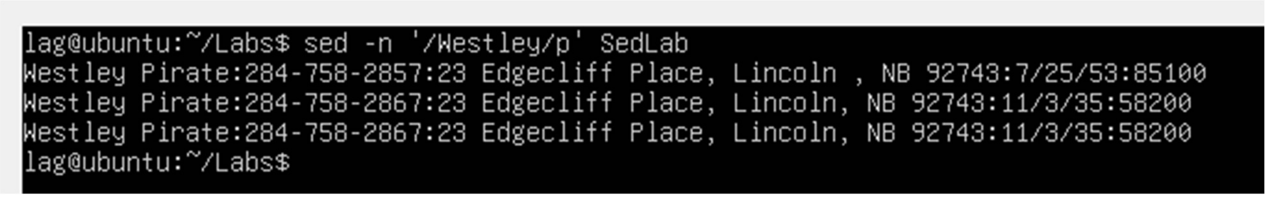


```
Dorian Gray:385-375-8395:3567 Populus Place, Caldwell, NJ 23875:6/18/68:23400***
***
Holly Golightly:397-857-2735:74 Pine Street, Dearborn, MI 23874:3/28/45:245700***
***
Ebenezer Scrooge:548-834-2348:583 Laurel Ave., Kingsville, TX 83745:10/1/35:58900***
***
```

Example 7. Replace the entire line containing "Westley Pirate" with the phrase "As you wish"

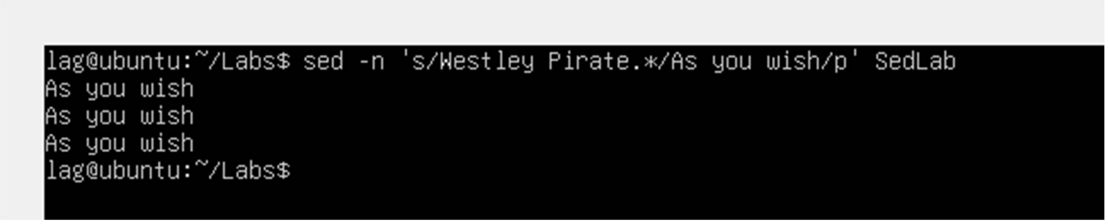
- **sed -n 's/Westley Pirate.*/As you wish/p' filename**

Figure 7.1 Result before the replacement



```
lag@ubuntu:~/Labs$ sed -n '/Westley/p' SedLab
Westley Pirate:284-758-2857:23 Edgecliff Place, Lincoln , NB 92743:7/25/53:85100
Westley Pirate:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
Westley Pirate:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
lag@ubuntu:~/Labs$
```

Figure 7.2 Result after the replacement



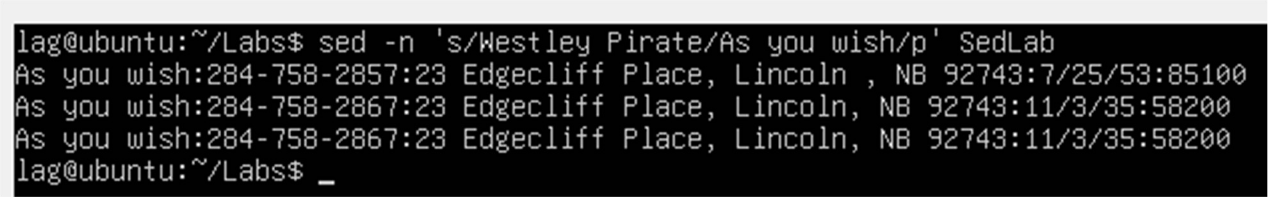
```
lag@ubuntu:~/Labs$ sed -n 's/Westley Pirate.*/As you wish/p' SedLab
As you wish
As you wish
As you wish
lag@ubuntu:~/Labs$
```

The command syntax is as follows:

- sed
- -n: suppress printing unless instructed
- s/: Substitute
- /Westley Pirate.*/: Text to match with the '.*' signifying the whole line to be replaced
- /As you wish/ : The text to substitute the line containing Westley Pirate
- /p: Print the results
- filename

In this example, sed identified all lines that contained the text 'Westley Pirate' and then substituted the whole line with the new text 'As you wish.' If the '.*' hadn't been included to signify that the whole line should be replaced, then only the text itself would have been replaced.

Figure 7.3 Results if '.*' were not included

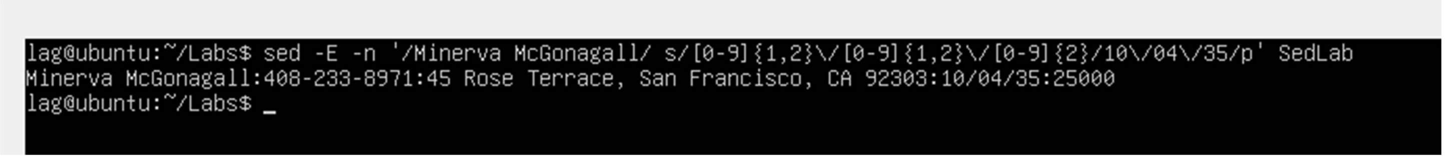


```
lag@ubuntu:~/Labs$ sed -n 's/Westley Pirate/As you wish/p' SedLab
As you wish:284-758-2857:23 Edgecliff Place, Lincoln , NB 92743:7/25/53:85100
As you wish:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
As you wish:284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
lag@ubuntu:~/Labs$ _
```

Example 8. Change Minerva McGonagall's birthday to 10/04/1935. Assume that the birthday is unknown and use regex to search for it.

- **sed -E -n '/Minerva McGonagall/ s/[0-9]{1,2}\/[0-9]{1,2}\/[0-9]{2}/10\/04\/35/p' filename**

Figure 8.1 Results showing Minerva McGonagall's birthday after sed substitution



```
lag@ubuntu:~/Labs$ sed -E -n '/Minerva McGonagall/ s/[0-9]{1,2}\/[0-9]{1,2}\/[0-9]{2}/10\/04\/35/p' SedLab
Minerva McGonagall:408-233-8971:45 Rose Terrace, San Francisco, CA 92303:10/04/35:25000
lag@ubuntu:~/Labs$ _
```

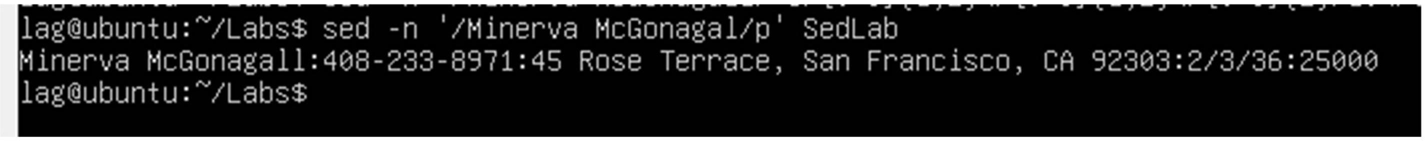
The command syntax is as follows:

- sed
- -E: Extended regex option
- -n: suppress printing unless instructed
- /Minerva McGonagall/: find lines that contain this name
- s/: Substitute
- [0-9]{1,2}\/[0-9]{1,2}\/[0-9]{2}: Regex for birthdate in format mm/dd/yy

- /10\/04\/35/p: Substitution text
- /p: print results
- filename

In this example, the regular expression searches for birthdates formatted as mm/dd/yy. The {1,2} in the brackets used in the month and day portion of the expression signifies that the month and day can be either one or two digits. Minerva's original birthdate was 2/3/36. If the regular expression had only allowed for two digits and used a pattern like [0-9]{2}\/[0-9]{2}\/[0-9]{2}, it would not have identified Minerva's birthdate as a match.

Figure 8.2: Results of Minerva’s original birthdate

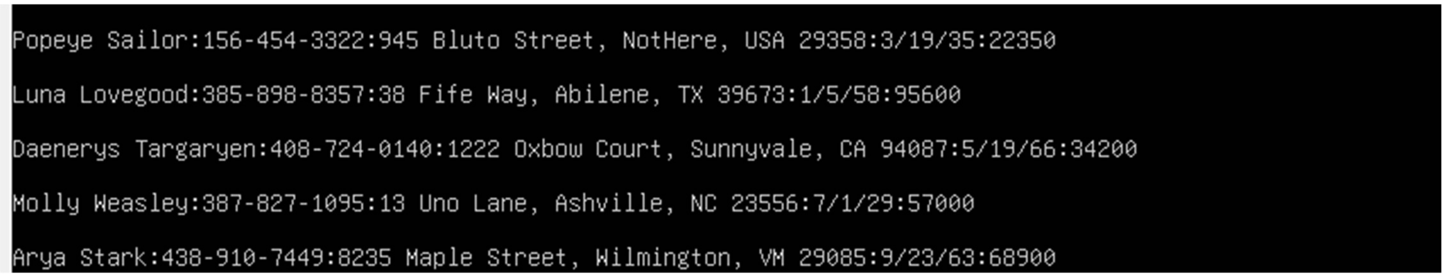


```
lag@ubuntu:~/Labs$ sed -n '/Minerva McGonagal/p' SedLab
Minerva McGonagall:408-233-8971:45 Rose Terrace, San Francisco, CA 92303:2/3/36:25000
lag@ubuntu:~/Labs$
```

Example 9. Delete all blank lines

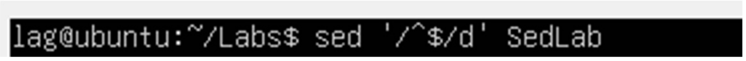
- **sed ‘/^$/d’ filename**

Figure 9.1 Sample of results before deletion

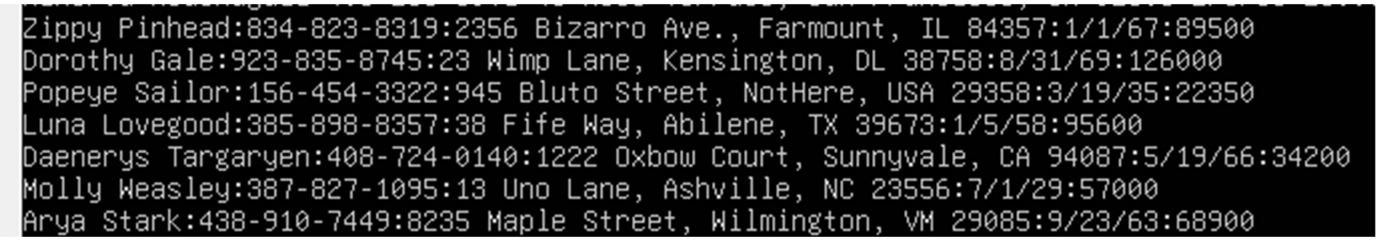


```
Popeye Sailor:156-454-3322:945 Bluto Street, NotHere, USA 29358:3/19/35:22350

Luna Lovegood:385-898-8357:38 Fife Way, Abilene, TX 39673:1/5/58:95600

Daenerys Targaryen:408-724-0140:1222 Oxbow Court, Sunnyvale, CA 94087:5/19/66:34200

Molly Weasley:387-827-1095:13 Uno Lane, Ashville, NC 23556:7/1/29:57000

Arya Stark:438-910-7449:8235 Maple Street, Wilmington, VM 29085:9/23/63:68900
```

Figure 9.2 Sample of results showing all blank/empty lines deleted



```
lag@ubuntu:~/Labs$ sed '/^$/d' SedLab
```



```
Zippy Pinhead:834-823-8319:2356 Bizarro Ave., Farmount, IL 84357:1/1/67:89500
Dorothy Gale:923-835-8745:23 Wimp Lane, Kensington, DL 38758:8/31/69:126000
Popeye Sailor:156-454-3322:945 Bluto Street, NotHere, USA 29358:3/19/35:22350
Luna Lovegood:385-898-8357:38 Fife Way, Abilene, TX 39673:1/5/58:95600
Daenerys Targaryen:408-724-0140:1222 Oxbow Court, Sunnyvale, CA 94087:5/19/66:34200
Molly Weasley:387-827-1095:13 Uno Lane, Ashville, NC 23556:7/1/29:57000
Arya Stark:438-910-7449:8235 Maple Street, Wilmington, VM 29085:9/23/63:68900
```

The command syntax is as follows: sed '/^$/d' filename. The ^$ pattern, without any text specified, matches the beginning and end of a line. Since no pattern was found, sed finds and deletes blank lines only.

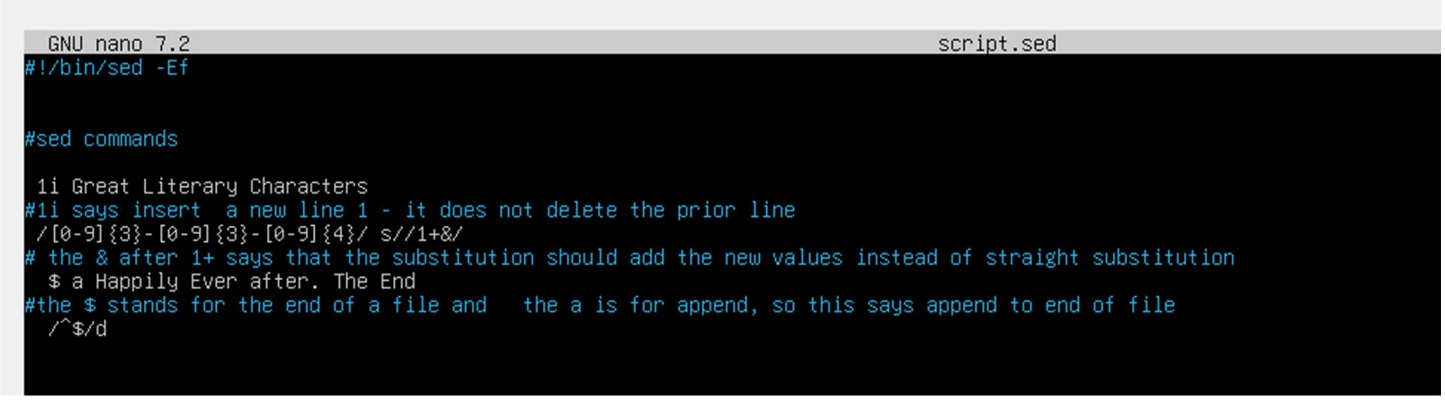
Example 10. Write a sed script that will:

a) Insert above the first line the title 'Great Literary Characters'

b) Print the contents of the file, but instead of the phone number starting with an area code, have it start with a 1+, then include the area code and number

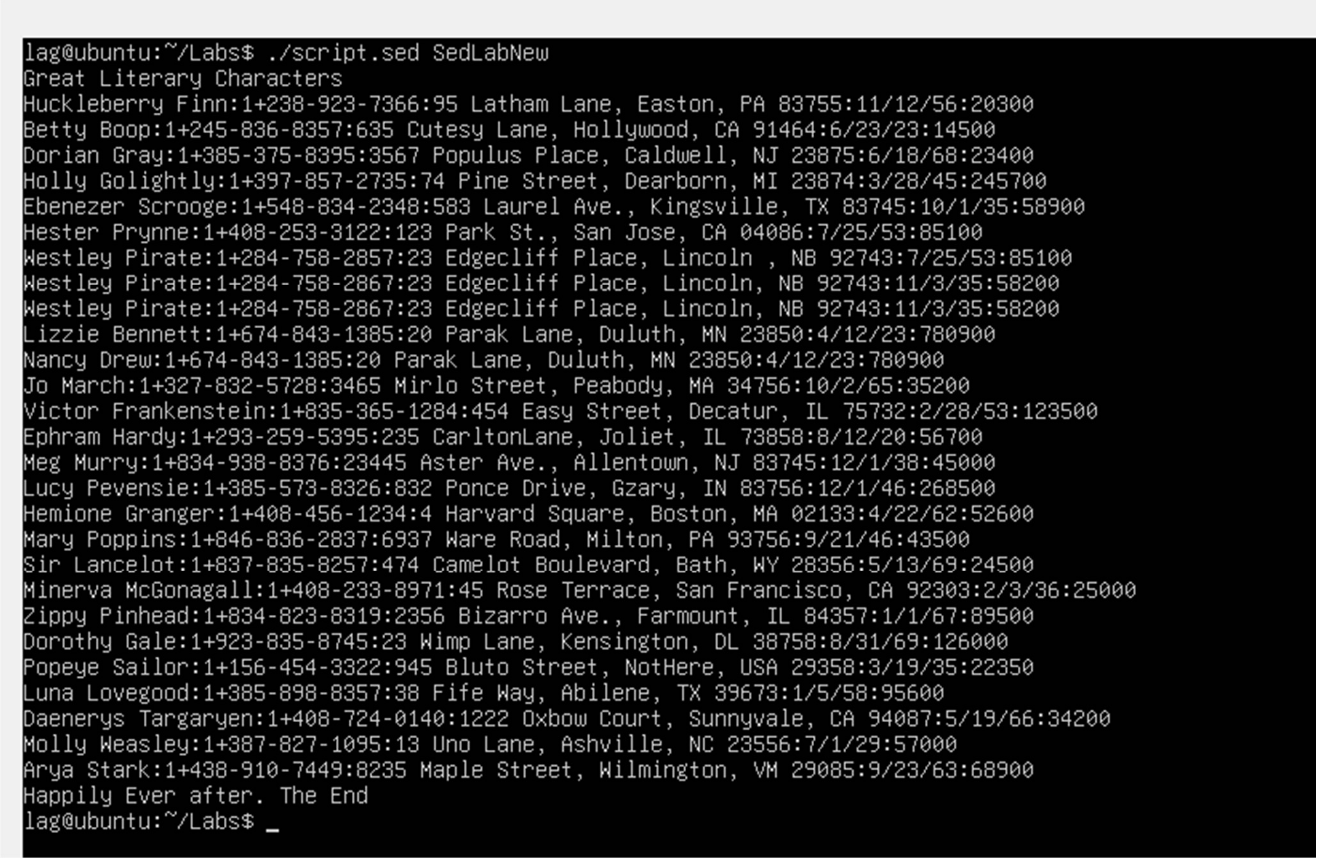
c) Append at the end of the file 'Happily Ever after. The End'

Figure 10.1 Screenshot of script.sed



```
  GNU nano 7.2                                                    script.sed
#!/bin/sed -Ef


#sed commands

 1i Great Literary Characters
#1i says insert  a new line 1 - it does not delete the prior line
 /[0-9]{3}-[0-9]{3}-[0-9]{4}/ s//1+&/
# the & after 1+ says that the substitution should add the new values instead of straight substitution
  $ a Happily Ever after. The End
#the $ stands for the end of a file and   the a is for append, so this says append to end of file
  /^$/d
```

Figure 10.2 Output of ./script.sed SedLabNew



```
lag@ubuntu:~/Labs$ ./script.sed SedLabNew
Great Literary Characters
Huckleberry Finn:1+238-923-7366:95 Latham Lane, Easton, PA 83755:11/12/56:20300
Betty Boop:1+245-836-8357:635 Cutesy Lane, Hollywood, CA 91464:6/23/23:14500
Dorian Gray:1+385-375-8395:3567 Populus Place, Caldwell, NJ 23875:6/18/68:23400
Holly Golightly:1+397-857-2735:74 Pine Street, Dearborn, MI 23874:3/28/45:245700
Ebenezer Scrooge:1+548-834-2348:583 Laurel Ave., Kingsville, TX 83745:10/1/35:58900
Hester Prynne:1+408-253-3122:123 Park St., San Jose, CA 04086:7/25/53:85100
Westley Pirate:1+284-758-2857:23 Edgecliff Place, Lincoln , NB 92743:7/25/53:85100
Westley Pirate:1+284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
Westley Pirate:1+284-758-2867:23 Edgecliff Place, Lincoln, NB 92743:11/3/35:58200
Lizzie Bennett:1+674-843-1385:20 Parak Lane, Duluth, MN 23850:4/12/23:780900
Nancy Drew:1+674-843-1385:20 Parak Lane, Duluth, MN 23850:4/12/23:780900
Jo March:1+327-832-5728:3465 Mirlo Street, Peabody, MA 34756:10/2/65:35200
Victor Frankenstein:1+835-365-1284:454 Easy Street, Decatur, IL 75732:2/28/53:123500
Ephram Hardy:1+293-259-5395:235 CarltonLane, Joliet, IL 73858:8/12/20:56700
Meg Murry:1+834-938-8376:23445 Aster Ave., Allentown, NJ 83745:12/1/38:45000
Lucy Pevensie:1+385-573-8326:832 Ponce Drive, Gzary, IN 83756:12/1/46:268500
Hemione Granger:1+408-456-1234:4 Harvard Square, Boston, MA 02133:4/22/62:52600
Mary Poppins:1+846-836-2837:6937 Ware Road, Milton, PA 93756:9/21/46:43500
Sir Lancelot:1+837-835-8257:474 Camelot Boulevard, Bath, WY 28356:5/13/69:24500
Minerva McGonagall:1+408-233-8971:45 Rose Terrace, San Francisco, CA 92303:2/3/36:25000
Zippy Pinhead:1+834-823-8319:2356 Bizarro Ave., Farmount, IL 84357:1/1/67:89500
Dorothy Gale:1+923-835-8745:23 Wimp Lane, Kensington, DL 38758:8/31/69:126000
Popeye Sailor:1+156-454-3322:945 Bluto Street, NotHere, USA 29358:3/19/35:22350
Luna Lovegood:1+385-898-8357:38 Fife Way, Abilene, TX 39673:1/5/58:95600
Daenerys Targaryen:1+408-724-0140:1222 Oxbow Court, Sunnyvale, CA 94087:5/19/66:34200
Molly Weasley:1+387-827-1095:13 Uno Lane, Ashville, NC 23556:7/1/29:57000
Arya Stark:1+438-910-7449:8235 Maple Street, Wilmington, VM 29085:9/23/63:68900
Happily Ever after. The End
lag@ubuntu:~/Labs$ _
```

This is a simple script that lists each sed command and then prints the file to the terminal without blank lines for easy viewing.

The #!/bin/sed shebang at the beginning of the script specifies that it should be executed using the sed interpreter, making it a sed script. The -E and f options instruct sed to use extended regular expressions and read the script from a file, respectively. It can be very helpful to add comments to all of your scripts. Comments can provide valuable documentation, which can be helpful for troubleshooting and understanding the intended purpose.

The first sed command inserted a new line 1 with the 1i option, essentially saying 'insert at line 1.' The next sed command used extended regular expressions to match the phone number pattern and then substitute/append a '1+' to it. The '1+&' used with the s for substitution specified that the '1+' should be added to the pattern, not replace it. The '$ a text' in the third sed command signified that the text to print should be appended (a) at the end of the file ($). Lastly, a sed command was used to delete all blank lines, so that the file would be printed on the screen with no blank lines. I found it helpful to create different versions of the scripts as I troubleshooted why things weren't working as expected. I used nano and would just save the changes under a new name, so that if the changes I made didn't work, I could go back to the original script.

References

Sed - An Introduction and Tutorial (grymoire.com)

Remove the Last N Lines of a File in Linux | Baeldung on Linux

Print Lines Using Sed p Command: Comprehensive Guide (likegeeks.com)

Appending to end of a line using 'sed' - Ask Ubuntu

https://www.reddit.com/r/linux4noobs/comments/yxoedz/sed_whole_lines_replacement/

How to Delete Specific Lines from File in Linux Command Line (linuxhandbook.com)

How to Insert a Line at the Beginning of a File on Linux - LinuxConfig

Sed Command in Linux - Append and Insert Lines to a File - #!/DevDude/

https://www.gnu.org/software/sed/manual/sed.html#Command_002dLine-Options

https://www.gnu.org/software/sed/manual/sed.html#sed-script-overview

https://www.grymoire.com/Unix/Sed.html#uh-20

https://www.diskinternals.com/linux-reader/linux-bin-bash-shell/

https://unix.stackexchange.com/questions/95939/how-exactly-do-i-create-a-sed-script-and-use-it-to-edit-a-file