Exercises-Sed

Description

sed is very powerful. We only cover the most useful of **sed's** commands. Even limiting yourself to these you can use **sed** to do some things that are otherwise impossible. The only commands we cover are substitute, print, delete, and placing pattern and line-number restrictions on each of these. We also cover capturing and substituting patterns.

Remember, with **sed**, the input is normally copied to the output. If you only want to output only the lines that you match, use the **p** flag and specify the **-n** option to **sed**.

In each of the examples below, we use **sed** to output the changed text to standard output. Of course if you wanted to actually change the input file, you would have to redirect standard output *to a different filename* and then rename it to the original using **mv**.

For the duration of these exercises, you will be using test files from the directory **samples/Data** beneath the public data area on hills.

Part One

Description

In this part we will practice simple operations with **sed**. We will use other Unix commands to get raw data for our filters. This exercise set uses options to sed but the only regular expressions that are necessary are the anchors **^** and **$**

Exercises

Using the file **u3**, do the following using **sed**, displaying the result on the screen

1. output only the lines that contain **cow**
2. output the lines that don't contain **cow** (read this: delete any line that contains **cow**)
3. change the first instance of ***** on each line to **!** (make sure you put the **sed** expression in single-quotes)
4. change all occurrences of ***** on each line to **!**
5. output only the lines that contain either **cow** or **calf**
6. change all occurrences of **cow** to **cows and cows** using the parenthesis operators and **\1** substitution
7. output the file after changing **cow** to **COW** on lines **10-20**
8. output the entire file except lines **1-20**
9. append three exclamation points to the end of each line in **u3** that contains **student**
10. repeat the previous command, but only output the lines that you change.
11. If you wanted to actually change the original file for questions #3,4,6,7, and 9, how would you do it?

Part Two

Description

In this part we will add regular expressions to our use of **sed**, using the files in the **samples/Data** directory.

Exercises

1. output the file **Depts** after changing the first character of the department number from **D** to **E**.
2. output only the lines in 1. that changed.
3. output the lines in **Depts** whose id (the third field) consists of either two or three digits.
4. output the lines in **sorttest** whose last field is all upper-case

5. output **sorttest** after adding the domain name **@ccsf.edu** to each email address
6. output the lines in **Emp_Manager** whose two department fields are the same. (Note: you can use the capture operation (the parenthesis operator) in the pattern used for an address as well as in the substitution pattern)

Answers - Part One

1. `sed -n '/cow/p' u3`
2. `sed '/cow/d' u3`
3. `sed 's/*/!/' u3`
4. `sed 's/*/!/g' u3`
5. `sed -n -e '/cow/p' -e '/calf/p' u3`
6. `sed 's/\(cow\)/\1s and \1s/' u3`
7. `sed '10,20s/cow/COW/g' u3`
8. `sed '1,20d' u3`
9. `sed '/student/s/$/!!!/' u3`
10. `sed -n '/student/s/$/!!!/p' u3`
11. Save the output of the **sed** command in a temporary file and then use the **mv** command to rename it to the original. *Never redirect output to the same file you are using for input within the same command or pipeline!* Example (#9):
    `sed '/student/s/$/!!!/' u3 > xxx` # <-- the shell overwrites **xxx** BEFORE it starts **sed**
    `mv xxx u3`

Answers - Part Two

1. `sed 's/^D/E/' < Depts`
2. `sed -n 's/^D/E/p' < Depts`
3. `sed -n -e '/:[0-9][0-9]:/p' -e '/:[0-9][0-9][0-9]:/p' < Depts`
4. `sed -n '/#[A-Z]*$/p' < sorttest`
5. this is difficult, but you can use some liberties from the file format. Since the email address is the next-to-last field, this will work:
   `sed 's/\(#[^#]*\)$/@ccsf.edu\1/' < sorttest`
6. `sed -n '/.*:\([DE][0-9]*\):\1:/p' < Emp_Manager`