Jackson Mediavilla

Recitation Section 103

Step 1

unzip Lab2_RequiredFiles.zip -d lab2

cd lab2

Step 2

- Cranberry, Nectarine, and Prickly Pear have been removed from fruitsOld.txt to get to fruitsNew.txt. Jackfruit has been added to fruitsOld.txt to get to fruitsNew.txt.
- '>' means that the text has been added to the old file to get the new file. '<' means that the text has been removed from the old file to get to the new file.
- The -c option outputs NUM lines of copied context. If NUM is not specified, the default, 3, is used.

Step 3

- wc -l testPasswd.txt
  - There are 15 lines in the testPasswd.txt
- wc -m testPasswd.txt
  - There are 692 characters in the testPasswd.txt

Step 4

- awk -F ':' '{print $1}' testPasswd.txt
- awk '{print $2 " " $4}' grades.txt

Step 5

- cut -d: -f4 testPasswd.txt | sort -g | uniq
- cut -d: -f4 testPasswd.txt | sort -g | uniq > /home/file.txt
- cut -d: -f1,6 testPasswd.txt | grep '^[mws]'

Step 6

- Remove all letters
  - sed 's/[a-zA-Z]//g' leetSpeak.txt
- Remove all numbers
  - sed 's/[0-9]//g' leetSpeak.txt
- Replace all numbers with an '_'
  - sed 's/[0-9]/_/g' leetSpeak.txt

- Replace each number with its matching character
    - sed 's/[0]/o/g' leetSpeak.txt | sed 's/[4]/a/g' | sed 's/[3]/e/g' | sed 's/[5]/s/g' | sed 's/[7]/t/g' | sed 's/[1]/i/g'
    - Because you can separate commands within sed with a semicolon, this same operation can be done without piping.
- I can make it so that the script does not have to change each time I want to run it on a different file by saving it in a bash file.

Step 7

- awk '{if (NR==1) {print "FN LN Grade"} else {for (i = 3; i <= NF; i++) j+=$i; print $1 " " $2 " " j/5*100; j=0 }}' grades.txt
  FN LN Grade
  Ryan Slaven 80
  Jephthah Eustathios 40
  Andreas Saša 60
  Godofredo Gerard 100
  Edwin Babur 80
  Ahmad Marin 0
- awk '{sum+=$3} END {print sum/(NR-1)*100}' grades.txt
  71.4286

Step 8

1. grep -E '[0-9]{3}[-][0-9]{3}[-][0-9]{4}' regex_practice_data.txt | wc -l
   770
2. grep "303-441-" regex_practice_data.txt | wc -l
   51
3. grep -E -o "\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,6}\b" regex_practice_data.txt | wc -l
   17705
4. grep -E -o "\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.gov\b" regex_practice_data.txt | wc -l
   52
5. grep -E -o "\b[A-Za-z]{2,}+\.[A-Za-z]{2,}+@[A-Za-z0-9.-]+\.[A-Za-z]{2,6}\b" regex_practice_data.txt | grep "^[a-mA-M]" | wc -l
   335

   This expression checks the following conditions:
    - the line is an email address in the format ‘first.last’ name
    - each name is at least 2 characters (to filter out initials)
    - the first name starts with a letter in the first half of the alphabet