sLab 4

# Using Sed

1. ***We have a customer who would like to buy a car, and we want to produce a list of cars that meet her requirements. She wants to see a list of all the cars on the lot that cost less than $10,000, except she doesn't want a chevy. The list should be sorted from lowest cost to highest, and we want to produce the list so that the alphabetic characters are in uppercase.***

***Let's start by displaying all the records in 'cars' that are not chevy's. Use a grep command to do this, and don't forget to ignore case:***

Answer: grep –iv “chevy” cars

1. ***Now let's delete the cars that are $10,000 or more. Pipe the output of the grep into a sed to do this, by deleting records that match a regular expression representing 5 (or more) digits at the end of a record (DO NOT use a repetition factor, such as {5}, for this):***

Answer: grep -iv chevy cars |

sed '/[0-9][0-9][0-9][0-9][0-9]$/ d'

1. ***Now let's sort the cars by price. Pipe the output of the sed into a sort to do this, and don't forget to sort numerically on the 5th field:***

Answer: grep -iv chevy cars | sed '/[0-9][0-9][0-9][0-9][0-9]$/ d' | sort –nk 5

1. ***Finally, let's display the output in uppercase. Pipe the output of the sort into a tr command:***

Answer: grep -iv chevy cars | sed '/[0-9][0-9][0-9][0-9][0-9]$/ d' | sort –nk 5 | tr “[a-z]” “[A-Z]”

# Using awk

1. ***We have another customer who would like to buy a car, and we need a list of cars that meet his requirements. He wants to rebuild a classic car built between 1975 and 1983. He's willing to pay up to $9,000, but he cares less about price than he does about low mileage, so the list should be sorted from lowest mileage to highest.***

***Let's start by displaying all the records in 'cars' that are newer than 1974. Use an awk command to do this, selecting those records that have a third field greater than 74:***

Answer: awk ‘$3 > 74’ cars

1. ***Now let's pare the list down to the cars that are older than 1984. Pipe the output of the awk into a second awk, selecting those records that have a third field less than 84:***

Answer: awk ‘$3 > 74’ cars | awk ‘$3 < 84’

1. ***Now let's delete the cars that are more than $9,000. Pipe the output of the second awk into a third awk to do this, selecting those records that have a fifth field less than or equal to 9000:***

Answer: awk '$3 > 74' cars | awk '$3 < 84' | awk '$5 <= 9000'

1. ***Finally, let's sort the cars by mileage. Pipe the output of the third awk into a sort to do this, and don't forget to sort numerically on the 4th field:***

Answer: awk '$3 > 74' cars | awk '$3 < 84' | awk '$5 <= 9000' | sort –nk 4

# Review Exercise

1. ***Display only the fifth line in the file, using sed.***

Answer: sed –n ‘5 p’ inventory

1. ***Display all of the lines in the file, changing the characters "Jam" to "Marmalade", using sed.***

Answer: sed ‘s/Jam/Marmalade/’ inventory

1. ***Display all of the lines in the file showing only the quantity and product name, in that order and separated by a space, using awk. Product name is the first field, and quantity is the second field.***

Answer: awk -F, '{print $2, $1}' inventory

1. ***Display all of the lines in the file with less than 100 items in inventory, using awk. Quantity is the second field.***

Answer: awk -F, ‘$2 < 100’ inventory

Use these notes:

<https://scs.senecac.on.ca/~les.czegel/ULI101/lectures/Lecture9.html>

<https://scs.senecac.on.ca/~les.czegel/ULI101/lectures/Lecture11.html>