WEEK 11 - SED AND AWK UTILITIES

sed utility

-> sed stands for stream editor

syntax: sed [-n (optional)] 'address instruction' filename

-> reads all lines in the input file and exposes to the expression line by line

-> you can use either single or double quotes

-> if the line matches the address, then it will perform the instruction on that line

-> sed '' cars = cat cars

-> default action of sed utility is print

-> -n to suppress the default print action (silence)

ADDRESS/MATCH CONDITION

-> line number

-> range of line numbers

-> regular expressions inside / /

INSTRUCTION

p -> print lines that match

q -> quit after first line that matches

d -> delete lines that match

s -> substitute

-> g flag for global change

-> you can also specify the occurrence

a -> append (sed '3a line added' cars)

i -> insert

c -> change

awk utility

syntax: awk [-F(optional)] 'selection-criteria {action}' filename

-> action, enclosed within curly braces, is executed when the criteria is matched

-> -F option to specify the delimiter (field separator)

-> default is whitespace, spaces or tabs

SELECTION CRITERIA

-> regular expressions inside / /

-> ~ test whether a specific field matches the regex

-> !~ test no match

-> numeric/string comparisons: > >= < <= == !=

-> || (OR) && (AND) operators

-> NR -> Number of record, line number

-> NF -> Number of field

-> $0 -> all fields

-> $1 -> first field

-> $n -> nth field

-> $NF -> last field (column)

A picture containing text, scoreboard

Description automatically generated

Append

Graphical user interface, text

Description automatically generated

Append “WEEK11” after 5th and 6th line

Graphical user interface, text

Description automatically generated

insert “line added” before the line 3

Text

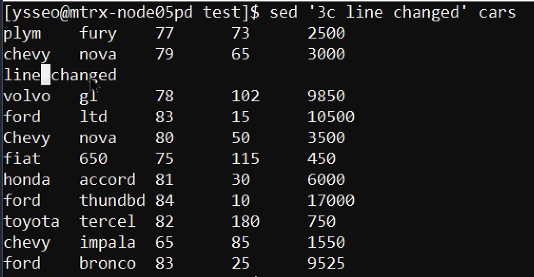
Description automatically generated

insert “line added” before the match line

A picture containing graphical user interface

Description automatically generated

Change line 3 to “line changed”



print all lines with line 5 twice

Text

Description automatically generated with medium confidence

print the last line of the file cars

Text

Description automatically generated with medium confidence

print from line 6 to the end of the file cars

A picture containing text

Description automatically generated

print from line 1 to the line 6 of the file cars

Text

Description automatically generated with medium confidence

print from line 1 to line 8 of the file cars

A picture containing text

Description automatically generated

print from line 1 to second last line of the file cars

Table, calendar

Description automatically generated with medium confidence

print from line 1 to the line that has “ford” keyword

A picture containing text

Description automatically generated

substitute ford with FORD

A picture containing text, scoreboard

Description automatically generated

substitute chevy with CHEVY (i : ignore case) / sed ‘s / [cC]hevy / CHEVY/ ’ cars

A picture containing text

Description automatically generated

substitute 0 with x

A picture containing text

Description automatically generated

substitute all 0s with x

Text

Description automatically generated with low confidence

substitute 每一行既第二個 0 with x

A picture containing table

Description automatically generated

get all cars that are from the 80’s

\s : whitespace character

(-E: allows you to use extended regular expressions : +)

\s+ : one or more whitespace characters

(without -E: we use regular expression : \*)

\s\* : zero or more whitespace characters

Graphical user interface, text

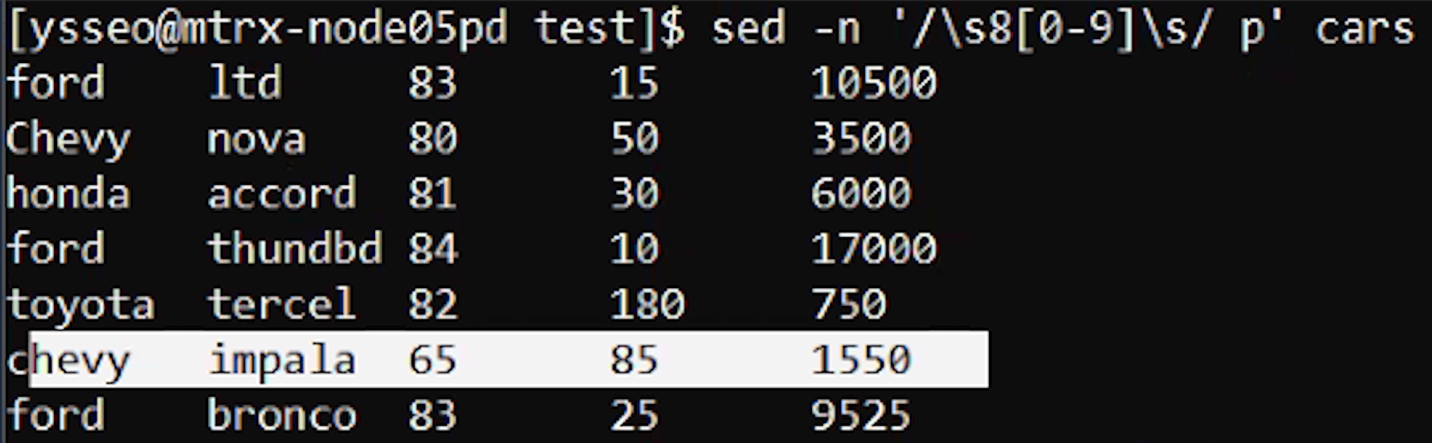
Description automatically generated

get all cars that are not from the 80s

A screenshot of a computer

Description automatically generated with medium confidence

if no [a-z], chevy impala 65 will also be included



sed -En '/[a-z]\s+8[0-9]\s/ d' cars.txt will print all lines in the file cars.txt that do not match a specific pattern.

sed -E '/[a-z]\s+8[0-9]\s/ d' cars.txt will delete (i.e., remove) all lines in the file cars.txt that match a specific pattern.

sed -n '/[a-z]\s\*8[0-9]\s/ !p' cars.txt will print all lines in the file cars.txt that do not match a specific pattern.

Text

Description automatically generated

print lines from 1-4, and 7-10

Table

Description automatically generated with medium confidence

OR

Table

Description automatically generated

print 第4欄 < 100 既所有lines

Table

Description automatically generated with low confidence

print 第5欄 > 10000 既所有lines既第1同第5欄 （中間既， “ “ \t \n 好影響格式）

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

-F: set delimiter

print 第3欄 > 1000 所有lines



print all lines from the file cars where the first field (i.e., the first column) does not contain the string "chevy" or "Chevy".

!~: test not match

$1 : first field

Text

Description automatically generated with medium confidence

OR

Text

Description automatically generated

print all lines from the file cars where the entire line does not contain the string "chevy" or "Chevy".

$0: Refers to the entire line of each record in the input file.

Graphical user interface, text

Description automatically generated

OR

Graphical user interface, text

Description automatically generated

print the second field (i.e., column) of each line in the file cars.txt if 第2欄長度is less than or equal to 4 characters.

Text

Description automatically generated

OR

Text

Description automatically generated

print all the lines if its second field contains a word that consists of 1 to 4 alphanumeric characters

$2 : second field

\<: Matches the beginning of a word.

\>: Matches the end of a word.

Text

Description automatically generated

calculate the sum of the fifth field (i.e., column) of each line in the file cars and print the total at the end.

n: variable

Text

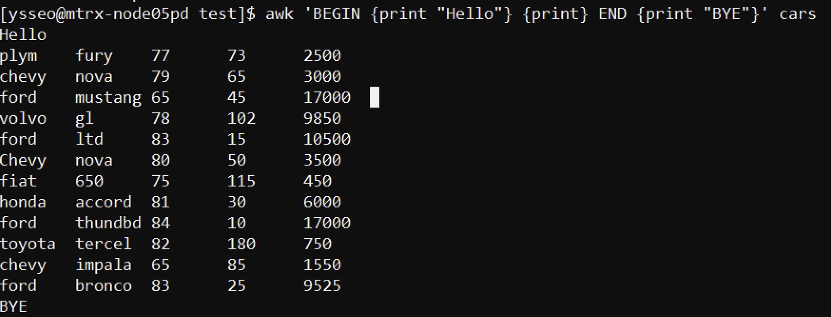
Description automatically generated

Only print the total

Text

Description automatically generated

BEGIN & END



print the last field (i.e., the field at the end) of each line in the file cars.

NF : Number of field

Text

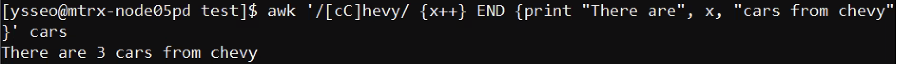
Description automatically generated

print the value of the last field (i.e., the field at the end) of the last line in the file cars.

Text

Description automatically generated

count the number of cars that are made by chevy, and print the following: “There are n cars from chevy” where n is the number of chevy cars



print the second field of the third and fifth lines of cars

NR : Number of record, line number

Text

Description automatically generated

print the line that has the largest value in the fifth column of the file cars

a. awk ‘max<$5 {max=$5; line=$0} END {print line}’ cars

b. awk ‘{if (m < $5) {m=$5; line=$0}} END {print line}’ cars

