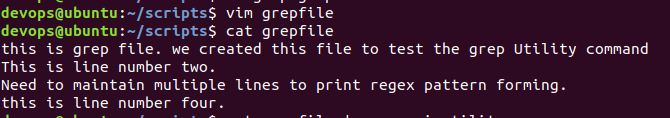
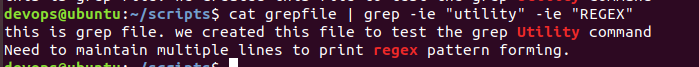
GREP:

‘Grep’ is kind of a utility operator which searches line for a matching regex pattern and print the line on terminal

Syntax: grep -e “Pattern”

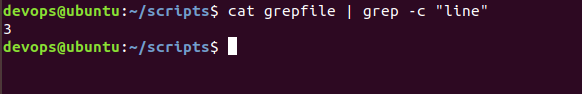
Question 1: Create a use case to search 2 regex pattern from same grep operation in a file.



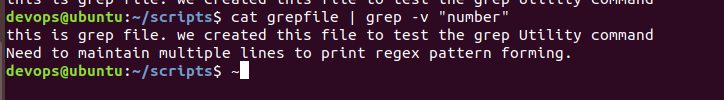


Question 2: Create use cases of 5 X flags for grep operation

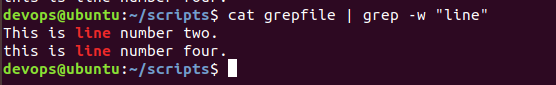
1. Grep -c (to print count of pattern matches)



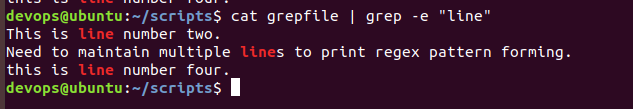
1. Grep -v (to print non-matching pattern lines)



1. Grep -w (to print line with exact matches of the word)



1. Grep -e (to print lines with matching expressions of the word)



1. Grep -o (to print only the matching word from the lines)



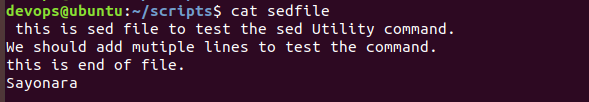
SED:

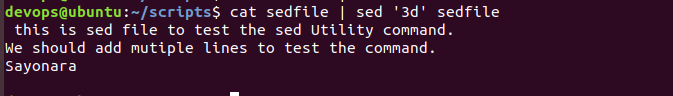
SED command in UNIX stands for stream editor and it can perform lots of functions on file like searching, find and replace, insertion or deletion. Though most common use of SED command in LINUX is for substitution or for find and replace. By using SED you can edit files even without opening them, which is much quicker way to find and replace something in file, than first opening that file in VI Editor and then changing it.

* SED is a powerful text stream editor. Can do insertion, deletion, search and replace(substitution).
* SED command in LINUX supports regular expression which allows it perform complex pattern matching.

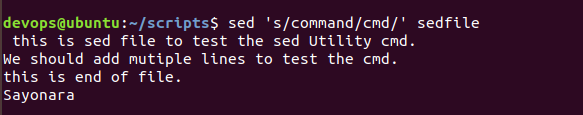
Syntax: Sed 's/ <pattern to replace> / <pattern to be replaced> /' <filename>

1. Sed ‘nd’ (Delete particular lines from file )

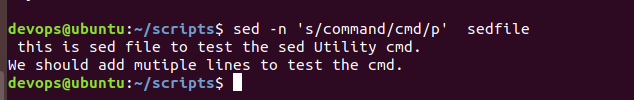




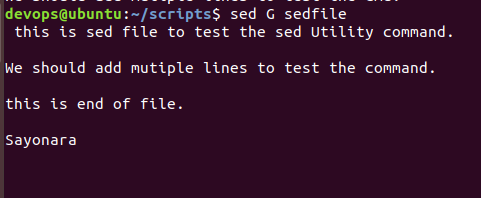
1. Replace lines in file



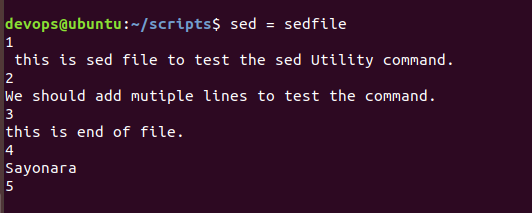
1. Sed -n 's/ <pattern to replace> / <pattern to be replaced> /p’ (Print only replaced lines)



1. Sed G <filename> (To insert one blank line after each line)



1. Sed = <filename> (To number each line)



\t is used for tab between number and sentence –

\n reads next line into pattern space

N adds new line into pattern space

