FILTERS

=> Filters are programs that take plain text (either stored in a file or produced by another program) as standard input, transforms it into a meaningful format, and then returns it as standard output. Linux has a number of filters. Some of them are :-

1. cat :- displays the text of the file line by line.

Syntax :-

```
cat [path]
```

2. head :- Displays the first n lines of the specified text files. If the number of lines is not specified then by default prints first 10 lines.

Syntax :-

```
head [- no.of lines to print] [path]
```

3. tail :- It works the same way as head, just in reverse order. The only difference in tail, is, it returns the lines from bottom to up.

Syntax :-

```
tail [- no.of lines to print] [path]
```

4. sort :- Sorts the lines alphabetically by default but there are many options available to modify the sorting mechanism.

Syntax

```
sort [-options] [path]
```

5. uniq :- Removes duplicate lines. uniq has a limitation that it can only remove continuous duplicate lines

Syntax:- uniq [options] [path]

6. wc :- It gives the number of lines, words and characters in the data.

Syntax:-

wc [-options] [path]

7. grep :- grep is used to search a particular information from a text file.

Syntax

grep [options] pattern [path]

8. tac :- tac is just the reverse of cat and it works the same way i.e. instead of printing from lines 1 through n, it prints lines n through 1.

Syntax

tac [path]

9. sed :- sed stands for stream editor. It allows us to apply search and replace operation on our data effectively. sed is quite an advanced filter and all its options can be seen on its man page.

Syntax

Sed [path]

Eg

$ cat ~~see~~ scoobydoo.txt

↳ Scoopy Dooby Doo

$ sed 's/Scooby/Scrapy/g' scoobydoo.txt

↳ Scrapy Dooby Doo

10. nl :- nl is used to number the lines of our text data

Syntax

nl [-Options] [path]

Regular Expressions

Linux regular expressions are special characters which help search data and matching complex patterns. Regular expressions are shortened as 'regexp' or 'regex'. Regexps are most commonly used with the Linux commands :- grep, sed, tr, vi.

Types of Regular Expressions :-

1. Basic Regular Expressions

.	replaces any character
^	matches start of string
$	matches end of string
*	matches up zero or more times the preceding character.
\	Represent special characters.
()	Groups regular expressions.
?	Matches up exactly one character.

These are used with commands like tr, sed, vi & grep commonly.

2. Interval Regular Expressions

These expressions tell us about the number of occurences of a character or string.

Expression

{n}	matches the preceding character appearing 'n' times exactly
{n,m}	Matches the preceding character appearing 'n' times but not more than 'm'.
{n,}	Matches the preceding character only when it appears 'n' times or more.

Example

iii. Extended Regular Expressions :- These regular expressions contain combinations of more than one expression.

Expression

\+	Matches one or more occurence of the previous character.
\?	Matches zero or more occurence of the previous character.

Q Given a file 'sample' containing texts

apple
bat
ball
ant
eat
pant
people
taste

1. search for content containing letter 'a'.

ii. search for content starts with 'a'.

iii. search for content ~~do not~~ that end with 't' using $

iv. search for content that contain 'p' appearing 2 times in a string one after the other.

v. search for content where character 'a' precedes 't'.

Soln (i) $ cat sample | grep a

↳ apple
bat
ball
ant
eat
pant
taste

ii. $ cat sample | grep ^a

↳ apple
ant

iii. $ cat sample | grep t$

↳ bat
ant
eat
pant

iv. cat sample | grep -E p\{2}

↳ apple.

v. ~~Search for co.~~

cat sample | grep "a\+t"

↳ bat
eat