Introduction 3

Note: Here I'm using egrep insted of grep command. Which is a newer version of grep command.

egrep "[A-Za-z]" test.txt : searches for a capital letter or a small letter.

egrep "[AB]" test.txt : searches for A or B.

egrep "[^AB]" test.txt : search for any thing expect A and B.

egrep "[.]" test.txt : searches for a .

egrep "." test.txt : searches for anything

egrep "hameed$" test.txt : searches for any sentence that hameed is the last word in it.

egrep "^hameed" test.txt : searches for any sentence that hameed if the first word in it.

egrep "Mrs?" test.txt : searches for Mr or Mrs

egrep "Mr(Abdal)?hameed" test.txt : searches for Mr abdalhameed or Mr hameed.

egrep "(A|B|D)a" test.txt : searches for Aa or Ba or Ca.

egrep “\.” test.txt : searches for .

egrep “\-\-” test.txt : searches for --

egrep “[0-9]{5}” test.txt : searches for any 5 numbers.

egrep “[0-9]{3,5}” test.txt : searches for at least 3 numbers and at most 5 numbers.

egrep “[0-9]{1, }” test.txt : searches for at least one number.

egrep “a.*b” test.txt : searches for a then zero or infinite number of characters then b.

egrep “of*.*ice” test.txt : searches for o then f is optional then zero or any number of characters then ice.

egrep “[0-9]+” test.txt : searches for at least one number.

+ sign: Allows multiple.

egrep “^(.)\1” test.txt : searches for any word that the first two numbers in it are the same.

egrep “^(.).*\1$” test.txt : searches for any word that first two letters and last letter are the same.

Notation	Meaning	Example	Matches
.	*any* character	`a..`	`a` followed by any two characters
^	beginning of line	`^wood`	`wood` only if it appears at the beginning of the line
$	end of line	`x$`	`x` only if it is the last character on the line
		`^INSERT$`	a line containing just the characters `INSERT`
		`^$`	a line that contains *no* characters
*	zero or more occurrences of a character	`W.*S`	`W` followed by zero or more characters followed by an `S`
		`.*`	zero or more characters
[*chars*]	any character in *chars*	`[tT]`	lower- or uppercase `t`
[^*chars*]	any character not in *chars*	`[^0 -9]`	any nonnumeric character
		`[^a-zA-Z]`	any nonalphabetic character
\{min,max\}	at least *min* and at most *max* occurrences of previous regular expressions	`x\{1,5\}`	at least 1 and at and at most 5 x's
		`[0-9]\{3,9\}`	from 3 to 9 successive digits
		`[0-9]\{3\}`	exactly 3 digits
		`[0-9]\{3,\}`	at least 3 digits
\(...\)	store characters matched between parentheses in next register (1-9)	`^\(.\)`	first character on line and stores it in register 1
		`^\(.\)\1`	first and second characters on the line if they're the same

Sed Command:

sed ‘s/old/new/’ test.txt : changes the word “old” to the word new.

sed -i ‘s/old/new/’ test.txt : this is the same of the last command, but in this case the file will be changed , unlike first case.

sed ‘s/old/new/g’ test.txt : changes all of the occurrence’s of the old word to new word.

sed -n ‘1,2p’ test.txt : prints first two lines.

sed -n ‘/abd/p’ test.txt : prints the lines that contains abd only.

sed ‘1,2d’ test.txt : deletes first two lines.

sed ‘1,10 s/old/new/g’ test.txt : changes the old word to the new word only in the first 10 lines.

sed ‘s/…//’ test.txt : removes first 3 characters form each line.

sed command	Description
`sed '5d'`	Delete line 5
`sed '/[Tt]est/d'`	Delete all lines containing Test or test
`sed -n '20,25p' text`	Print only lines 20 through 25 from text
`sed '1,10s/unix/UNIX/g' exp1.txt`	Change unix to UNIX wherever it appears in the first 10 lines of `exp1.txt`
`sed '/jan/s/-1/-5/'`	Change the first -1 to -5 on all lines containing jan
`sed 's/...//' exp1.txt`	Delete the first three characters from each line of `exp1.txt`
`sed 's/...$//' exp1.txt`	Delete the last 3 characters from each line of `exp1.txt`
`sed -n 'l' text`	Print all lines from text, showing nonprinting characters as \nn (where nn is the octal value of the character), and tab characters as \t

سبحانك اللهم وبحمدك أشهد أن لا إله إلا أنت أستغفرك وأتوب إليك