Week-6:

a) Write a sed command that deletes the first character in each line in a file.

b) Write a sed command that deletes the character before the last character in each line in a file.

c) Write a sed command that swaps the first and second words in each line in a file.

a. Write a sed command that deletes the first character in each line in a file.

```
$ cat mytable
1425 Ravi 15.65
4320 Ramu 26.27
6830 Sita 36.15
1450 Raju 21.86
7820 Anil 23.34
9000 Allam 35.56

$ sed 's/^./ / ' mytable

425 Ravi 15.65
320 Ramu 26.27
830 Sita 36.15
450 Raju 21.86
820 Anil 23.34
000 Allam 35.56
```

b. Write a sed command that deletes the character before the last character in each line in a file.

```
# Delete the last character in each line in a file.
$ sed 's/.$/ /' mytable
1425 Ravi 15.6
4320 Ramu 26.2
6830 Sita 36.1
1450 Raju 21.8
7820 Anil 23.3
9000 Allam 35.5
```

c. Write a sed command that swaps the first and second words in a file.

```
# Swaps the first and second words in each line in a file.
$ sed 's/^\(.*\),\(.*\)$/\2\\\1/g' mytable
Ravi 1425 15.65
Ramu 4320 26.27
Sita 6830 36.15
Raju 1450 21.86
Anil 7820 23.34
Allam 9000 35.56
```

Sed command

sed command in Linux stands for stream editor and it can perform lots of functions on file like searching, find and replace, insertion or deletion. Its most common use is for substitution or for find and replace. Files can be editing by using sed command 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 command in Linux supports regular expression which allows it perform complex pattern matching.

Syntax:

sed [OPTIONS] [SCRIPT] [file]

Sr. No.	Option	Description	Command
1	s/…/.. /	used to replace the text in a file.	$sed 's/name/surname/' text.txt
2	s/…/2	Replacing the nth occurrence of a pattern in a line	$sed 's/name/surname/2' text.txt
3	s/…/g	Replacing all the occurrence of the pattern in a line	$sed 's/name/surname/g' text.txt
4	s/…/3g	Replacing from nth occurrence to all occurrences in a line	$sed 's/name/surname/3g' text.txt
5	n s/…/…/	Replacing string on a specific line number	$sed '3 s/name/surname/' text.txt
6	s/…/p	Duplicating the replaced line	$sed 's/name/surname/p' text.txt
7	-n s/…/p	Printing only the replaced lines	$sed -n 's/name/surname/p' texte.txt
8	n,m s/…/p	Replacing string on a range of lines	$sed '1,3 s/name/surname/p' texte.txt
9	nd	Deleting lines from a particular file	$ sed '5d' text.txt
10	$d	To Delete a last line	$ sed '$d' text.txt
11	x,yd	To Delete line from range x to y	$ sed '3,6d' text.txt

Regular Expression Metacharacters

Metacharacter	Function Example	Matches
^	Beginning of line anchor	'^word' Matches all lines beginning with word.
$	End of line anchor	'word$' Matches all lines ending with word.
.	Matches one character	'l..e' Matches lines containing an l, followed by two characters, followed by an e.
*	Matches zero or more characters	'*love' Matches lines with zero or more spaces, of the preceding characters followed by the pattern love.
[ ]	Matches one character in the set	'[Ll]ove' Matches lines containing love or Love.
[^]	Matches one character not in the set	'[^A–KM– Z]ove' Matches lines not containing A through K or M through Z, followed by ove.
+	Matches one or more of the preceding characters	'[a–z]+ove' Matches one or more lowercase letters, followed by ove. Would find move, approve, love, behoove, etc.
?	Matches zero or one of the preceding characters	'lo?ve' Matches for an l followed by either one or not any o's at all. Would find love or lve.
a|b	Matches either a or b	'love|hate' Matches for either expression, love or hate.
()	Groups characters 'love(able|ly) (ov)+' Matches for loveable or lovely.	Matches for one or more occurrences of ov.