**Call this file as lines**

Line one.

The second line.

The third.

This is line four.

Five.

This is the sixth sentence.

This is line seven.

Eighth and last.

**Exercise1 : 5 mins**

1. Print lines 1,4.
2. Print lines 5,7
3. Print lines 1,3 and 6,8 on all files in a directory
4. *In the lines file print all  lines that begin with capital T*

sed -n '1,4 p ' lines

sed -n '5,7 p' lines

sed -n -e '1,3 p' -e '6,9 p' lines

sed -n '/^T/ p' lines

**Exercises 2**

1. After fifth line insert Text called “After fifth line”
2. Before Eights line insert text called “before 8th line”
3. Replace lines 3 to 6 with text saying lines 3 to lines 6 replaced.

sed '5 a \after fifth line\ ' lines

sed '8 i \before 8th line\ ' lines

sed ' 3,6 c\3 to 6 lines replaced\ ' lines

**Exercises 3**

1. Delete lines 3, to 5.
2. Delete lines in range from is to the

sed '3,5 d' lines

sed ' /is/,/the/ d' lines

Exercises

Use this file. (compound.in)

1. The words on this page…

      2. The words on this page…

      3. The words on this page…

      4. The words on this page…

Exercise 1 .

sed -f compound2 compound.in

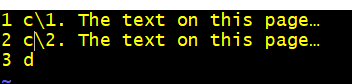
my output should be

1. The text on this page…

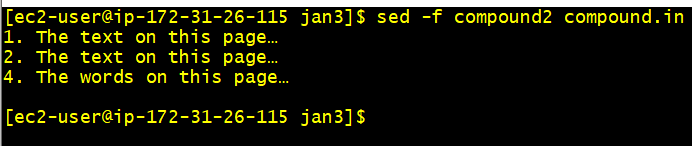
2. The text on this page…

4. The words on this page…

vi compound2



Output;



**sed -e '1 c\1. The text on this page…' -e '2 c\2. The test on this page...' -e '3 d' compound.in**

Hint : it requires more than one sed instruction and they should be applied in order.

**Exercise 2**

sed -f compound4 compound.in. But there is a constraint to achieve the solution: use line **number 2 and not line number  1 for all modifications. Idea  is to demonstrate order of instructions matter. Have a delete instruction at least once.**

sed -f compound4 compound.in

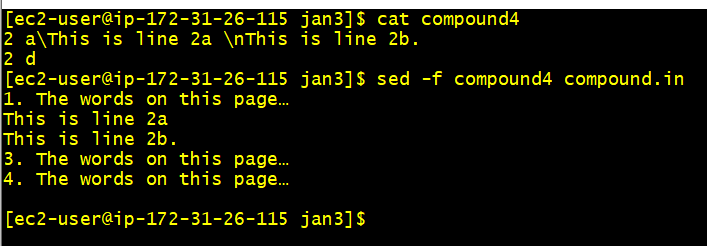
1. The words on this page…

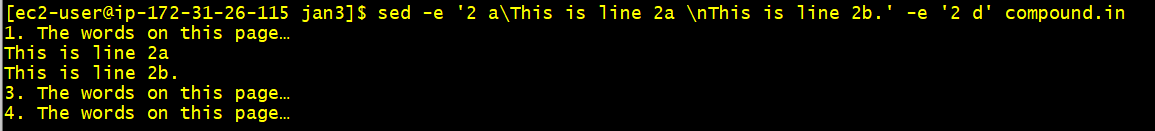
This is line 2a.

This is line 2b.

3. The words on this page…

4. The words on this page…



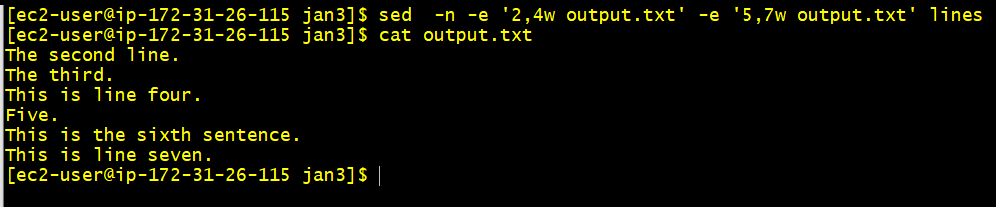


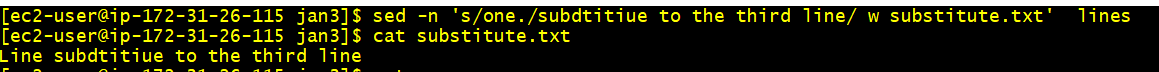
**Exercises 3 .**

1. **Save lines 2 to 4 and 5 to 7 in a single file in a single command.**
2. **Substitute sentence to a line and save to sentence file.**

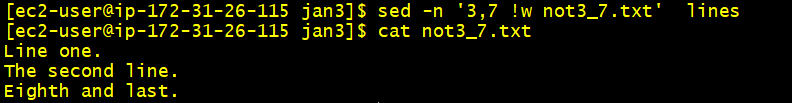
**2a) all lines other than those modified should also be saved. File is sentence\_all\_modified**

1. **Save files other than between 3 to 7 to not\_3\_to\_7.**





sed -e 's/the/sentence modified/ ' lines > sen\_all\_modified



**Exercises 4**

1. **Print all lines except those containing the. Call the sed file as lines\_without\_the**
2. **Produce a sed file which when instructed will produce output below.  (very difficult)**

Line one.

The second line.

The third.

This is line four.

Five.

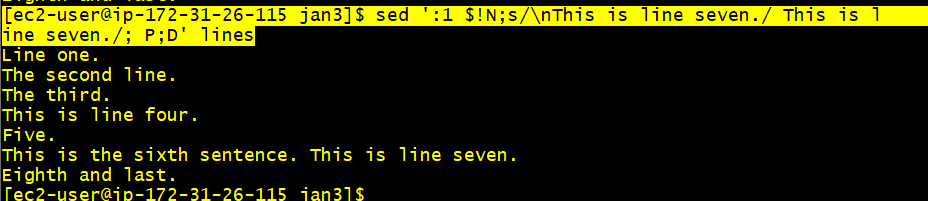
This is the sixth sentence. This is line seven.

Eighth and last.

ANS:

sed -n '/the/d; w lines\_wo\_the' lines

grep -v 'the' lines > usinggrep\_wo\_the



**sed ':1 $!N;s/\nThis is line seven./ This is line seven./; P;D' lines**

sed ':a $!N;s/\nThis is line seven./ This is line seven./;ta P;D' lines

1. **Write a shell script or a sed instrcution that adds tabs to all the lines on all files supplied on the command line. Use sed as a shebang line.**

sed 's/^/\t/' lines

1. **Write a shell script that trims all lines on a file.**

todayfile

**This is Today and this line should show up.**

**Today is Friday.**

**This should not be removed .**

**today in lower case**

**today is Saturday**

**Today it is slightly sunny**

1. Write a sed command that copies a file to standard output, removing all lines that begin with the word Today. Make it case insensitive.

2. Write a sed command that copies only those lines of a file that begin with the word Today to standard output.

3. Write a sed command that copies a file to standard output, removing all blank lines (i.e., lines with no characters on them).

ANS:

sed '/today/I d' todayafile 1>stdoutput

sed -n '/^Today/ p' todayafile 1>stdouttoday

sed -n '/^$/ p' todayafile 1>stdoublank

**For 4 and 5 use the file below call it catsfile**

4. Write a sed program named ins that copies a file to standard output, changing all occurrences of cat to dog and preceding each modified line with a line that says the following line is modified:. cat to dog.

 case insensitive and only whole words should be replaced. Example: concatenate should not become condogenate and cat(s) should be dog(s) . Case insensitive.

5. Write a sed program named div that copies a file to standard output, copies the first five lines to a file named first, and copies the rest of the file to a file named last.

4.ANS

sed 's/cat/dog/I' catsfile 1>catsmodified

5. ANS:

sed -n '1,5p' catsfile 1>firstfive

sed -n '6,$p' catsfile 1>last

**catsfile**

**The Naming of Cats is a difficult matter,**

**It isn’t just one of your holiday games.**

**You may think at first I’m as mad as a hatter**

**When I tell you, a cat must have THREE DIFFERENT NAMES.**

**All of them have sensible everyday names.**

**Names that never belong to more than one cat.**

**Pussy cat, pussy cat, where have you been?**

**I’ve been down to London to visit the Queen.**

**Pussy cat, pussy cat, what did you there?**

**cats should be dogs.**

**But above and beyond there’s still one name left over,**

**The name that no human research can discover—**

**But THE CAT HIMSELF KNOWS, and will never confess.**

**When you notice a cat in profound meditation,**

**Of the thought, of the thought, of the thought of his name:**

**His ineffable effable**

**Deep and inscrutable singular name.**

**catch should not be be dotch**

**CATS should be DOGS**

**Do the exercises below More sed exercises. (do it folder called alice)**

<https://www.upriss.org.uk/perlcgi/alice.txt>

1) Retrieve all lines from alice.txt that do not contain /the /. Retrieve all lines that contain "the" with lower or upper case letters. Save it to file called only\_the

sed -n '/the/I W onlythe' alice.txt

sed -n '/the/ !w notthe' alixe.txt

2) a) Retrieve lines that contain a word of any length that starts with t and ends with e. Modify this so that the word has at least three characters.

Sed -n /^t.\*e/p’ alice.txt

b) Retrieve lines that start with a. Retrieve lines that start with a and end with n. Hint: You need to specify the beginning of the line, "a", any number of any characters in the middle, "n", end of line.

sed -n '/^a.\*n$/ p' alice.txt

c) Retrieve blank lines. Think of at least two ways of doing this.

Sed -n ‘/^$/p’

d) Retrieve lines that contain a word that starts with an uppercase letter.

Sed -n ‘/^[[:upper:]]/p’

3) What is the difference between the following expressions?

a) abc\* and (abc)\*

b) ! preg\_match("/yes/"...) and /[^y][^e][^s]/

c) [A-Z][a-z]\* and [A-Z][a-z]+