Sed Practice Problems

Instructions for Mac users

If you are on mac and wish to run sed from zsh terminal (your own terminal), then follow the instructions to download gnu-sed. (preinstalled sed on Mac has some limitations!). (If you are unable to click on the link use this: https://medium.com/@bramblexu/install-gnu-sed-on-mac-os-and-set-it-as-default-7c17ef1b8f64)

1 Let's start with Fruits

Write a sed expression that takes a list of fruit names as input(each on a new line), and appends "fruit_" at the start of every fruit name.
Usage: sed -f fruits.sed input.txt > output.txt
{-f: sed expression is contained in a file; fruits.sed will contain your code}
input.txt:
apple
banana
mango
dragon_fruit
apple
output.txt:
fruit_apple
fruit_banana
fruit_mango
fruit_dragon_fruit
fruit_apple

2 Indian Economy

Imagine you have a file which has words separated by underscores instead of spaces. In this file replace every occurrence of 3trillion_{word} with {word}_5trillion. The following sample outputs will make it clear:
Usage: sed -E -f growth.sed input.txt > output.txt
input.txt:
3trillion_India
US_3trillion_India_3trillion_India
output.txt:

```
sed -E 's/3trillion ([^ ]+)/\1 5trillion/g' input.txt > output.txt
```

India_5trillion
US_India_5trillion_India_5trillion

input.txt:
3trillion_India
US_3trillion_
output.txt:
India_5trillion
US_3trillion_
Explanation: The second line has 3trillion but no word after that, so we don't replace it.

3 Let's Trade

You are given demat IDs of some traders (each on a new line), you need to extract their date and month of joining. The ID codes are stored as: N...CCCCMMMDD where N... is a sequence of numbers (arbitrary length > 0). CCCC are the last 4 characters of there PAN card(not of much interest to us). MMM represents month and DD represents date. You need to replace these IDs with "DD MMM". Write a sed script. You can skip invalid lines. (If you write a correct code, sed will automatically do this!)
Usage: sed -E -f trade.sed input.txt > output.txt
input.txt:
122CDEFAPR22
CCDFMAY06
11CDEJUN05
134_6_DJAN14
output.txt:
22 APR
CCDFMAY06
11CDEJUN05
14 JAN

```
sed -E 's/([0-9]+)([^0-9])(.{3})([A-Z]{3})([0-9]{2})/\5 \4/' input.txt
```

Explanation: The first line has 3 digits then 4 uppercase characters then month in 3 uppercase characters and two digits at the end, so it is a simple case. The second line is invalid because it has no digits at the start. The third line is also invalid because it does not have four characters CCCC. The fourth line is valid because 'CCCC' can be any set of 4 characters. 'MMM' will always be uppercase letters.

4 Try Something Difficult

Write a sed script that makes every first letter of the word capitalized and other letters small. (Words can be separated by spaces or full stops.) Also, for every letter with a space after a period/dot(.), it should not only be capitalized, but also enclosed in brackets.
Usage: sed -E -f tough_replace.sed input.txt > output.txt
input.txt:
hEllo
hello.champ

LaTeX. sed. bash. awk, anything else?
sAkshaM. rAthi
output.txt:
Hello
Hello.Champ
Latex. (S)ed. (B)ash. (A)wk, Anything Else?
Saksham. (R)athi

Explanation: In the first line, we make the first character 'h' capital and rest small. In the second line both hello and champ will have their first letter capitalized, but since there is no space between period and c, we do not enclose it in brackets. In the third line we, will make all the first letters capitalized, then we will make s, b and a enclosed in brackets, because they are after a period. Similarly for the fourth line after changing case, r is enclosed in brackets.

```
sed -e 's/\(.*\)/\L\1/' input.txt|sed -e "s/\b\(.\)/\u\1/g"|sed -e "s/(\.\.\)/\u\1/g"|sed 's/\. \(.\)/\. (\1)/g'
```

lowercase everything
uppercase first letter
uppercase first letter after .
put the letter after . inside ()