SED and AWK Problem Statement
**********

---
Solving Instructions for outlab assignment
---

Strictly follow the instructions given below:

**1. Solution for each assignment is to be typed into the .sh or .sed or .awk file already kept inside the corresponding folder.**

- **Use InLab folder structure and replace files / make folders accordingly. We will provide only input files for problem 4 and problem 6 for outlab.**

2. Remember, you need to change the permission of the .sh file before executing it:
chmod u+x <scriptfile> or chmod 777 <scriptfile>

3. Do not change the structure and the names.. The automatic checker will give you a "notsubmitted" grade otherwise, if it does not find the names.

4. Remember to follow the exact output formats specified. Extra characters will lead to incorrect evaluation by automatic checker.

5. Inside the script if you use any temporary file, write command to remove it after the operation within that script only.

---
Submission Guidelines for assignment
---
1. Create a file readme.txt in team-name directory, which contains contribution of each team member and references (cite where you get code/code snippets from).

2. Compress the directory to <team_name>.tar.gz

3. Submit one assignment per team, preferably the lowest roll number.

**********

Out Lab Assignment

---
Problem 1:
---

You are leading the organization of IPL this year and have been provided with the Played-Win-Loss tally of all the teams.

Given input file has 4 fields. There are field titles and then data in each column.

In our case, the input file has fields to represent team names, no of matches played, no of wins, tied matches. You have to calculate points scored by every team and place it in the fourth column titled "Points".

4 points for win, 0 for loss and 2 for tie.

Write a script to generate and print the output on screen.

for example, if the input is

| Team | Played | Wins | Tied |
|---|---|---|---|
| A | 2 | 1 | 1 |
| B | 2 | 0 | 1 |

then the output is

| Team | Played | Wins | Tied | Points |
|---|---|---|---|---|
| A | 2 | 1 | 1 | 6 |
| B | 2 | 0 | 1 | 2 |

The table is tab separated for both input and output.

How to execute:
./script1.sh <inputFileName>
(here filename is passed as an ARGUMENT and NOT AS AN INPUT STREAM)

---
Problem 2:
---

Your input file contains text along with new line characters and white spaces. You should write a script that will collapse every sequence of multiple white-space characters between two words in a single line into a single space.

Your script should also collapse every sequence of blank lines (multiple white spaces between two lines) into a single new-line character.
Prefix and suffix whitespace characters must be removed.
You should print your output on the terminal

Example: If the input is as follows:

"  how  are  you

  fine.

"

The output should look as follows:

"how are you
fine."

How to execute:
./script2.sh <inputFileName>
(here filename is passed as an ARGUMENT and NOT AS AN INPUT STREAM)

---
Problem 3:
---

How about a simple challenge! Do you know about a tool called ‘wc’? It helps you count the no. of lines, no. of words, characters and even bytes of a text file.

Can you design your own version of ‘wc’ without using ‘wc’ itself? Let us trivialize the problem a little. Write a bash script to take a filename as an input argument and calculate the no. of line, no. of words, and no. of characters. Your script should also be able to take flags which are optional and should be able to print simply the no. of line if the line flag is given as an argument. Note that only one flag will be provided at a time At minimum, the flags should be:

-lines = when this is provided to the script along with the input file, the script should ONLY print the no. of lines in the file.

-words = when this is provided to the script along with the input file, the script should ONLY print the no. of words in the file.

-chars = when this is provided to the script along with the input file, the script should ONLY print the no. of characters in the file (you must ignore the whitespaces) and count other characters.

-paras = when this is provided to the script along with the input file, the script should ONLY print the no. of paragraphs in the file (you can count a blank line as a separation between two paragraphs - simple!) and count other characters.

For e.g.:

```
   How are      you
 I am fine
 Thank you!

 How   are you?
```

output:
35 characters, 11 words, 12 lines, 2 paragraphs

The above should be the output when nothing is provided as a flags to the script, but in case a flag is provided you script must only print the value for the flag provided.

For e.g.:  >>./script3.sh <file> -para

>>2 paragraphs

---
Problem 4:
---

Let’s deal with a real life problem now. You have dealt with enough imaginary scenarios. We have a list of Name, Addresses, and E-mail ID’s in a PDF file. We use Optical character recognition to bring it to DOC/DOCX and then TXT. You will find the TXT file in the folder for “script5”. If you look at the file carefully, you will notice that there is a pattern to the records maintained in it. Line 1 is usually a name, Line 2 onwards you have address, unless you encounter E-mail ID and then the last line has e-mail ID. There are some blank lines here and there, of course, you would be able to identify and clean them up. The challenge here is to be able to create a CSV file out of this. A file which has column headings:
Name||Address||EmailID

and later every row entailing the heading is a record and the data is stored in the respective columns separated by ||.

The task at hand:

Take the input text file as an argument to your script. Make sure you take into account whitespaces, newlines, blank lines, here and there and eventually produce the output as described above.

Note: The order in the output should be the same as in input file

---
Problem 5:
---

National Survey Association has hired you for the post of Associate Data Analyzer. As part of training programme, they have given you the following assignment. The assignment is pretty simple and is as follows:

You will be given a .csv file as an argument. It contains ‘Name’, ‘City’ and ‘Salary’ fields. You have to write an AWK script that will sum the salary of all the people working in a city and output the total salary grouped and sorted by city. By sorted we mean an alphabetical sort on the city names.

Note: The input file will contain records separated by ‘,’ (comma)
(You must print the output on the terminal)

For example:
Ram,Mumbai,200
Shyam,Mumbai,500
David,Delhi,300

Output should be:
Mumbai,700
Delhi,300

How to execute:
awk -f <fileName> <inputFileName>

---
Problem 6:
---

Have you heard of Natural Language Processing (NLP) / Artificial Intelligence? NLP is quite heavily done at Google, FB, MS and many other organizations and startups. What we would like you to do now is to calculate TF i.e. Term Frequency for all the unique words in the given set of documents.

Go here, if you need to know more:
https://en.wikipedia.org/wiki/Tf-idf
(just look at the TF part, IDF requires one to calculate the log of values which is not easily available in bash and we are not expecting you to code it up, so just TF would be good).

Term frequency is calculated using the count of words . Only consider unique set of words.
E.g.
**Input file contains:**
*We are the students of IIT Bombay, we are here to learn Computer Science and Engineering. We were surprised to know that our TA's are themselves students from the same institute. Ta is a ta and among the ta's "students" "student"* student

**Unique words are given below:**
a
among
and
are
bombay
computer
…
...
iit
…
...
...
surprised
students
student
ta's
ta
that
the
themselves
to
we
Were

**NOTE:** ***So, remove all the punctuation marks at the end of the words, or at the beginning. The ones within the words should stay where they are (e.g. TA's -> ta's) . So frequency of*** **students** ***in above file is 2(not 1).***

Now count the frequency for each word in these set of words.

The word dictionary you created now gets extended to a file where you also get the count of each word which occurs in the document. This count can be used to calculate the TF. Please do check out other references from the web as well, which may help you with this.

We provide you with three documents:
<input1> <input2> <input3>

Your script must be able to take as input 3 arguments of these documents and should be able to calculate TF for all the unique set of words in the documents/input files.

Output:
The output should printed on the terminal and not stored in a file. Please be wary of this. I really do not want to repeat… idiosyncracies..deduction of marks..

Thanks! May the Bash be with you!