CS251 Inlab 2 : RegEx (Sed + Awk)

Please refer to the general instructions and submission guidelines at the end of this document before submitting.

(PFA the testcases for the Inlab 2 questions. Simply copy-paste your scripts to appropriate folders in inlab2-testcases/ and run the test.sh for every sub-folder within it)

Task 1 - (10 Marks)

Create a **sed** script to parenthesize the first character of each word in a text file only in the case that the word starts with a digit or an alphabet.

Create a file q1.sed which does so.

Sample Input : cs251 is one of the MOST interesting lab in 2nd Year .
Sample Output : (c)s251 (i)s (o)ne (o)f (t)he (M)OST (i)nteresting (l)ab (i)n (2)nd (Y)ear.

How to execute:
sed -f q1.sed <filename>

Tip:- Be careful what is a word boundary and look at the **test cases apriori**.
(Do not worry about the word boundaries and trust what \b demarcates as a word boundary)

Run **./test.sh** to test for the visible testcases. Your code will be tested on additional hidden testcases as well.

Hints:- Use \b , \1 (confused what is \1 , search the net)

Task 2 - (10 marks)

A program written by you gets the input from an embedded systems device (Joystick on a gamepad) through a serial port on your machine. The program generates output in the following format:

```
1,X!2,Y!3,Z!2.5,O!
```

where the **, (comma)** denotes a field separator and **! (exclamation mark)** denotes a record separator.

Write an awk program **q2.awk** which takes as an input argument a file of the format specified above and generates the output on the command line (prints the output; not in a file) of this format:

How to Execute:

```
awk -f q2.awk <inputFile>
```

Output:

```
Value  SensorNumber
1      X
2      Y
3      Z
2.5    O
```

The above should be the output on the command line and nothing else should be generated, not a file, not anything extra written. The script output will be graded programmatically, and any idiosyncrasies shall lead to deduction of marks.

Run **./test.sh** to test for the visible testcases. Your code will be tested on additional hidden testcases as well.

Task 3 - (15 Marks)

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

Given the 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**, **number 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 a win, 0 for loss and 2 for tie

Write a bash script named **q3.sh** to generate and print the output on the command line.

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:
./q3.sh <inputFileName>

Run **./test.sh** to test for the visible testcases. Your code will be tested on additional hidden testcases as well.

Task 4 - (65 Marks)

Write ~~one sed script and one awk script~~ **one bash script q4.sh** to create a dictionary. You can have other supporting awk and sed file . Please make sure to submit them. Dictionary has the following properties

1. All words are **lowercase**. Convert Uppercase to lowercase. (tYPo->typo)
2. All words are **unique**. No word should appear twice in the dictionary.
3. Each word appears on a **separate line**.
4. No punctuations are considered as a word and should not appear in the output unless it is in between a word like “don’t”.
5. Two words are separated by whitespace (space,tab,newline etc). Whitespaces are not considered as words i.e. there should not be any blank line in the output.
6. Punctuations at starting and ending of words are to be removed. i.e. “its” and “it’s” are different words but “here” and “here,” are the same. Also “ users ” and “ users’ ” are the same in out problem though they have separate meanings. Also note once removing a punctuation exposes another punctuation then that also has to be removed i.e. “say…”->”say”.
7. Safely assume there are no numbers in the file provided.

Dictionary should be created as an output the following command
~~sed -f q4.sed <input> | awk -f q4.awk~~
./q4.sh <input>

Marks distribution /Subtasks:-

A. Convert all words to lower_case.(5 marks)
B. Move all words to a new line. Remove Empty lines too. (15 marks)
C. Removing the punctuations before and after words. Remove lines containing only punctuations too. (20 marks)
D. Remove duplicate words and sort. (25 marks)

Note:- These are to be treated as checkpoints. In case you pass part D it means you have passed all the above tasks too and are awarded 65(25+20+15+5) marks. In case you fail D you will checked for part C and will be awarded 40(20+15+5) marks in case you pass it. If C is failed you will be checked for B. and so on. In case you fail A you will get 0 marks out of 65.
~~Note B will be graded if task A is complete and C will be graded only if B is complete and so on.~~ **~~Also, you are restricted to exactly one sed script and exactly one awk script~~**. ~~Any other format will attract 0 marks. In case you feel you can do with only sed then also you need to submit both scripts where the awk can be left untouched(not blank as the command will fail otherwise). Similarly in case you think you can do only with awk then leave the .sed script untouched.~~

(You are not allowed to use any other regex related command other than sed and awk eg. grep/sort/cut. You can create new files but make sure at the end you have deleted all the extra files created)

Also make sure you update the test.sh by redownloading the link after 7:20 pm

Run **./test.sh** to test your implementation. It checks your output in order D,C,B,A and shows whether you have passed it or not. In case you pass D it will not check for C . In case you ,B,A .

Sample Input and Output is in the folder structure.

General Instructions

- Make sure you know what you write, you might be asked to explain your code at a later point in time.
- The submission will be graded automatically, so stick to the naming conventions strictly.
- The deadline for this lab is **Thursday, 8th August, 11:55 PM**.

Submission Instructions

After creating your directory, package it into a tarball

`inlab2-<roll_number>.tar.gz`

The directory structure should be as follows (nothing more nothing less). Also, even if you don't have any references then please just add an empty text file with the same name.

Make sure all your scripts are executable . To do so you can run **chmod 755 <scriptname>.**

The problem statement is frozen and you can write assumptions if any in assumptions.txt to submit it.

```
inlab2-<roll_number>/
├── Task1
│   ├── q1.sed
├── Task2
│   └── q2.awk
├── Task3
│   └── q3.sh
├── Task4
│   ├── q4.sh
│   └── Any other files that is required by your script
├── references.txt
└── assumptions.txt(optional)
```