CS251 OUTLAB - 5
Advanced Bash (sed and awk)
(Autumn-2021)

General Instructions:

- The submission will be graded automatically, so stick to the naming conventions and follow directory structure strictly.
- In cases where output is required to be submitted in the form of .csv/.txt/.png make sure that your **code** should **generate** the file at desired location at runtime.
- The deadline for this lab is **Sunday, 26th september, 11:59 PM.**
- After creating your directory, package it into a tarball: **rollno1-rollno2.tar.gz**
- **Command to generate tarball:**

  tar -czvf rollno1-rollno2.tar.gz rollno1-rollno2/
- **Incorrect directory structure will lead to marks penalty**. **Please make sure we see the exact directory structure as shown below when we decompress the archive.**
- Don't put any additional files in your submission.
- Submit once only per team from the moodle account of the smallest roll number.
- **The directory structure should be as expected.** Quote major references in `references.txt`.
- You get 20% of the total marks of every question if the solution passes the given sample input output. The rest of the test cases are hidden. Hard coding solutions will lead to penalties.
- **You have 5 questions in total , with a bonus question at the end.**
- **Evaluation:** min(100, marks_for_submitted_questions)

Here are some resources to start with .

1. awk & sed
2. awk

Q1.Text Preprocessing (use sed) **(23 marks)**

This is a common data cleanup stage that will be widely used in your further NLP projects This makes the text less noisy and ready to be fed to the model.

Let's replicate this task of text preprocessing , precisely the steps given below.

- Remove empty samples i.e., word length =0
- Make sure text has only letters . Remove any punctuations, numbers, symbols.
- Lower case all the alphabets
- Removing URLs (starting with https: or http: or www.)
- Remove Stop words (read list of stop words provided in stopwords.txt)
- Make sure spaces are used appropriately
- Remove words that have word length <=2
- Stemming - The task of removing certain prefixes and suffixes and retaining the word. The valid suffixes to be removed are in the files suffixes.txt . Remove only the suffix, not the entire word.

Given in sample.txt are noisy sentences which are to be processed.

Write the code in Q1.sh to take input from file sample.txt and generate solution to output.txt.

Provided File Structure:	Submission File Structure:
Q1/ ├── sample.txt ├── sample_output.txt ├── stopwords.txt └── suffixes.txt	Q1/ ├──Q1.sh ├── sample.txt ├── sample_output.txt ├── stopwords.txt └── suffixes.txt

Further Details:

1. Remove stopwords first and then go for stemming.
2. Figure out the correct order of executing the rest of the instructions.
3. Remove the word if it is a suffix by itself.
4. Stemming:
    a. Every word will have only one suffix. No recursive removal of suffix is expected
    b. Suffixes are to be removed in the order as in suffixes.txt . Example: “goes” can have two suffixes “es” and “s”
        i. goes -> goe if we see “s” first
        ii. goes -> go if we see “es” first

Q2. Text Tokenization (use awk) **(13 marks)**

The stage after preprocessing is tokenization, separating the text word wise and giving them a unique number and feeding for further processing.

Example Input sentence : they never said they were never going they were going

Example Output sentence:0 1 2 0 3 1 4 0 3 4

Let the unique number assigned to a word be according to the order of their occurrence starting from 0.

Given sample.txt are processed text which are to be tokenized to numbers.

Write the code Q2.sh to take file from sample.txt and generate solution to output.txt

File Structure:

Provided File Structure:	Submission File Structure:
Q2/ ├── sample.txt └── sample_output.txt	Q2/ ├── Q2.sh ├── sample.txt └── sample_output.txt

[PTO]

Q3. Data Analysis (use awk) **(18 marks)**

Even before jumping into the math , analysing the thus cleaned data is an important stage as it helps in finding trends and drawing conclusions.

Here we are considering the problem of spam classification (if the message is a spam it classified as class-1 else class-0).In the file sample.txt you are given tokenized sentences , each starting with their class label, followed by : and then each token separated by | (Tokens here are numbers).You are also provided with word_token_mapping.txt that has the token word matching in token-word pattern one in each line.

You are expected to return two files ham.txt and spam.txt with each having N most repeated words (not tokens) of that class (in decreasing order of their frequency)(break the ties lexicographically if necessary) ,one in each line. N here is the command line argument. Provided are the sample_ham.txt and sample_spam.txt for sample outputs of sample.txt with N=3.

Write the code Q3.sh to take file from sample.txt and N from command line and generate solutions to ham.txt and spam.txt

```
$ ./Q3.sh 3
```

File Structure:

Provided File Structure:	Submission File Structure:
Q3/ ├── sample.txt ├── sample_spam.txt ├── sample_ham.txt └── word_token_mapping.txt	Q3/ ├── Q3.sh ├── sample.txt ├── sample_spam.txt ├── sample_ham.txt └── word_token_mapping.txt

[PTO]

Q4. Attendance (use awk) **(28 marks)**

You've been assigned as a TA for the CS251 course. You're given a csv file in which each row consists of the name of the student, "Joined" or "Left" and the time of joining or leaving respectively (a student can join and leave the meeting multiple times, although some diligent ones might go as far as to never leaving the meeting :P).

You want to find out the total amount of time for which a student stayed online during the meet.

Your task is to create a bash file which will take in 3 arguments: 1. the input csv file 2. the starting time HH:MM:SS for example: 14:00:00 (3. the ending time HH:MM:SS for example: 15:30:00

```
$ ./Q4.sh sample.csv 14:00:00 15:30:00
```

Running the above command should generate an output file with the name output.txt in which there will be a sorted(on names) list of the names and opposite to each name the time for which the student was online (HH:MM:SS).Please refer to the given sample input and output files for further clarification. Clarifications regarding this question:

1. Student joining before starting time is to be considered joining at start time itself, same with the leaving case, Student leaving after ending time is to be considered leaving at end time.

2. There will not be any case of type Join-Join-Leave or Join-Leave-Leave chronologically for any student A (aap chronology samajhiye…).

3. No two students will have the same name.

4. Output should be sorted based on names of the students.~~In case of students having same names ascending order of the time duration should be taken.~~

File Structure:

Provided File Structure:	Submission File Structure:
Q4/ ├── sample.csv └── sample_output.txt	Q4/ ├── Q4.sh ├── sample.csv └── sample_output.txt

[PTO]

Q5. Plan Vacay **(18 marks)**

In the file sample.txt you are given one location in each line. The file covid_status.html is the saved html of a copy of this site which shows the covid cases in each state. Code to use covid_status.html file to extract the number of cases of each state and to return the thus asked state names and total confirmed cases ,one in each line in output.txt in increasing order of the number of cases which will help me plan my vacation.

You can create any temporary files .Make sure your final submitting structure is as asked.

Write the code Q5.sh to take file from sample.txt and generate output to output.txt

File Structure:

Provided File Structure:	Submission File Structure:
Q5/ ├── covid_status.html ├── sample.txt └── sample_output.txt	Q5/ ├──Q5.sh ├── covid_status.html ├── sample.txt └── sample_output.txt

[PTO]

Q.Bonus (use awk): **(18 marks)**

1)Your task for this question is to perform addition of two numbers and write the result to file for each test case.

Sample Input:

```
1

10 2

9 9 9

1 0 0
```

Input Description:

Sample Input is given in a bonus_sample_input.txt file.First line of the file indicates number of test cases.Second Line gives the base of the input and output separated by spaces. The next two lines of each test case have two numbers. The digits of the numbers are space separated. The base is guaranteed to be less than or equal 16.

2<=Base<=16

Expected output for the above input:

```
1 0 0 0 1 0 0 1 0 1 1
```

Output Description:

Output should be written to bonus_output.txt.The digits in each result should be space separated.Each line should contain only one result.Write the code bonus.sh to take file from bonus_sample_input.txt and generate output to bonus_output.txt

File Structure:

Provided File Structure:	Submission File Structure:
Bonus/ ├── bonus_sample_input.txt └── bonus_sample_output.txt	Bonus/ ├── bonus.sh ├──bonus_sample_input.txt └── bonus_sample_output.txt

[PTO]

One last check

Final Submission Directory:

```
rollno1-rollno2/
├── Q1
├── Q2
├── Q3
├── Q4
├── Q5
├── Bonus
└── references.txt
```

- Individual question folders **should** have structure as expected in individual questions.

```
 _______________________
/                       \
|      The End          |
\_______________________/
```