Lab-3 Content + Exercises:

Evaluation: Evaluation will happen during lab hours. As and when you complete exercises on every section, call your TAs to evaluate you on the code/command and its output.

NOTE:

1. Exercises are to be done during lab time. Evaluations will not continue after 5 pm.
2. Some questions are marked as practice questions, for you to try later or during the lab. They won't be counted for evaluation.

Topic 1: Regex

Regex is grep's way of pattern matching. The simplest pattern to search for is a string. For example,
> grep "this" file.txt
searches for all occurances of the word "this" in file.txt.
It doesn't stop at this. You can match classes of characters. Say you want to find a pattern starting with the letter c, followed by a vowel, and ending with the letter t. Regex makes it simple to search for "any one of the given characters". This will match with **cat**, **cot**ton, **cut**e, but not ceat, since it matches with only one vowel.
> grep "c[aeiou]t" file.txt
To complement a class,
> grep "c[^aeiou]t" file.txt

For certain specific cases like capitalisation only at the beginning of a sentence we use ^[A-Z}, or matching a digit at the end of a line is done using [0-9]$. These are known as anchors.

Other useful things to look up: Repetition, Grouping.

Here's a resource that might help. https://qntm.org/files/re/re.html
Feel free to look up other resources as well.

Test yourself: (not to be submitted, discuss this in lab amongst yourselves or with your TA)

Qn: Find which lines do each of the below patterns match with.

Patterns:

\([a-z]\)\1
f[^aei]r
h.l
te*n

ro*m
b[aeiou]g\>
s[aeiou]*[s-z]
[ATGI]\{2,3\}

Lines:

1. I found a bug in the tenth line of my code.
2. A fairly big southern state is abbreviated as TN.
3. The teenager's baggage is in your room.
4. This song's for you.
5. Arms, legs, and hands are parts of the body.

Practice:

1. List all files in Documents directory that were created in Jan and are txt files.
2. In "hamlet.txt" find all those lines where the pattern "th" occurs at least twice.

Exercises:

1. For the file "hamlet.txt", find all patterns that:
    a. Matches "ly", "rea" or "self".
    b. Sentence does not begin with "T".
    c. Contain "s" or any special character (which in this case are ? , -)
    d. Do not contain the pattern "the", but contains the pattern "to"

Topic 2: Sed

Sed is a stream editor that lets you edit a file without opening it in an editor like vim, emacs or sublime. Editing takes place through the command line or from within a script.

How does sed work?

- Read: The file is read from an input stream (stdin, file, or pipe) and is stored in an internal buffer
- Execute: All the sed commands are applied sequentially on this buffer
- Display: The changed content is sent to the output stream, and the buffer is emptied.

This process is repeated till the file ends. This happens on all lines unless otherwise specified.

Do note that all the sed commands are applied on the buffer, therefore the original file remains unchanged.

Useful fact: Multiple sed commands can be written in a text file and can be given as an argument to sed. Google how to do this !!

Practice:

1. In one command (use pipes), replace 2018 with 2019 in the line “Welcome to 2018”.
2. Create a new file named “file1.txt”, with the line “This is fun”. Change it to “This is great fun” and print on the screen. Now print the contents of the file.
3. Parenthesize the first character of each word in the file “hamlet.txt”.
4. Use the file “hamlet.txt”. Replace all occurrences of “be” with “me”, and every 2nd occurrence of “to” with “for” in a line, using a single command. (sed flags)
5. Simulate the cat command using sed.

Exercises:

1. Given a file named card.txt, write the command that will replace the first 10 digits to ‘#’, and save the output in a file called cardOut.txt. (use regex with sed)

   card.txt contains:
   1234 5678 9101 1234
   2999 5178 9101 2234
   9999 5628 9201 1232
   8888 3678 9101 1232

   cardOut.txt contains:
   #### #### ##01 1234
   #### #### ##01 2234
   #### #### ##01 1232
   #### #### ##01 1232

Topic 3: Awk

Awk is a scripting language for text processing, and is much more powerful than sed. It allows us to define variables, use arithmetic and string operators, use control flow, and generate reports.
Here’s a good tutorial to learn its usage: https://likegeeks.com/awk-command/
Try out commands from the tutorial before starting with the exercises. Also discuss the bonus questions of Lab-2 with your TAs.

Practice:

1. Write an awk command that numbers the lines of the file hamlet.txt
2. Write the command that changes all characters in login/user field of /etc/passwd to uppercase letters.

Exercises:

1. Given a file with 3 space separated fields (marks in 3 subjects, out of 100 each), concatenate a ":" and mention the grade as per (A if avg>=80, B if avg>=60, C if avg>=40, FAIL if avg<40) using a single awk command which takes the file marks.txt as input. Write the output to grades.txt.

   marks.txt
   75 78 80
   25 27 50
   35 37 75
   99 88 76

   grades.txt
   75 78 80 : B
   25 27 50 : F
   35 37 75 : C
   99 88 76 : A

Topic 4: Git (version control)

- Here's a quick intro to why version control is necessary: https://www.atlassian.com/git/tutorials/what-is-version-control
- The idea is to use a tool to keep track of versions of your project. It gives you a clear idea of the history of changes, you need not worry about breaking things that are currently working, and merging changes from multiple people is almost effortless.
- Git Guide: https://guides.github.com/introduction/git-handbook/
- **Create your github account, and your first repository.** https://guides.github.com/activities/hello-world/
- Git terminology can be overwhelming, so here's a quick dictionary: https://www.cloudways.com/blog/git-cheat-sheet/

Practice:

1. Create a repository on Github and clone it
2. Create a few commits on the master branch.
3. Then create a branch and create a few more commits
4. Merge master onto feature.
5. Checkout previous commits.
6. Try git reset after the merge (--hard)

Exercises:

1. Create a local repository
2. Create file foo.txt with some text and commit it
3. Create another branch <new>
4. Create file bar.txt in branch <new>

5. Merge master into <new>
6. Do the same things as above with a cloned repository and push the changes

Topic 5: HTML/CSS

Here's a good resource: https://www.w3schools.com/html/
An HTML file will contain the content of your website, and your CSS sheet — which stands for Cascading Style Sheet — will style the content of your HTML file.

Related readings:

About the internet:
https://developer.mozilla.org/en-US/docs/Learn/Common_questions/How_does_the_Internet_work
WWW:
https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/How_the_Web_works

Exercises:

1. Create the main page of the lab exercise as shown in figure 1. Save it as **index.html**

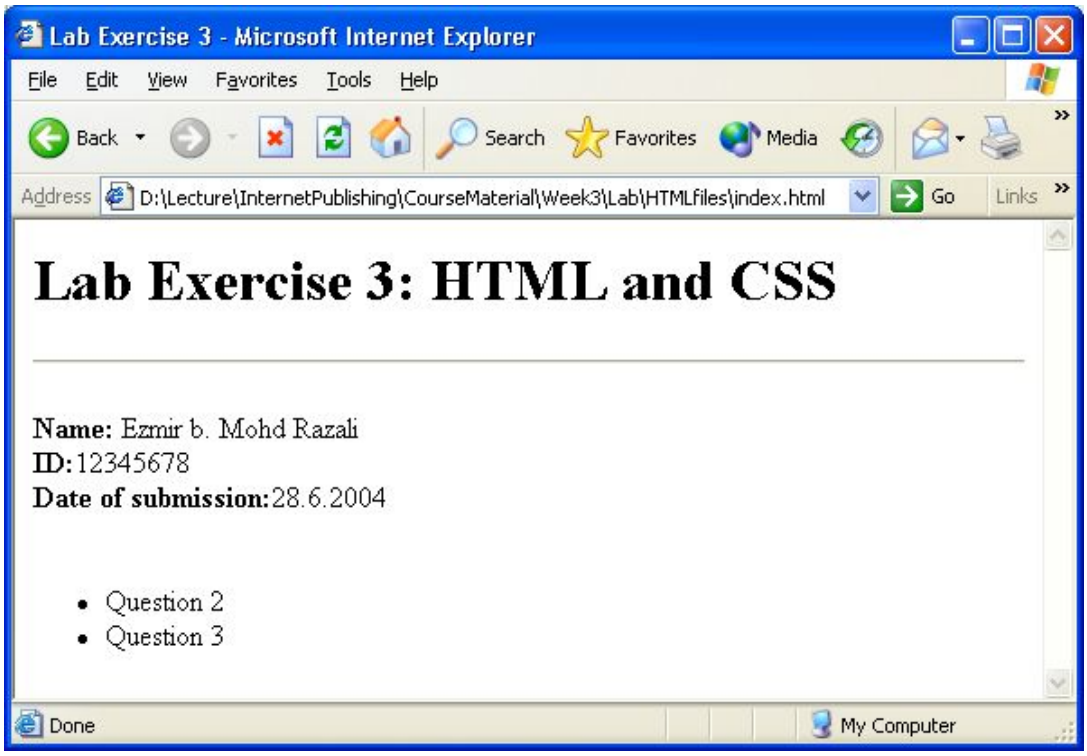

Figure 1: Main page (index.html)

2. Create a HTML document as shown in figure 2 and save it as **question2.html** (same directory with index.html). You can replace the first 3 links with your own favorite links

Figure 2: question.html

Instructions for Question 2:

- The first 3 links will open another HTML document in a **new window**(hint: use attribute target = '_blank' in anchor element)
- The last link (Back to Main Page) will redirect the user to the main page (index.html)