Lab 08: Introduction to Regular Expressions and Grep

What you will practice

- Grep
- Elementary regular expressions

Prerequisite

- Download the following **phonebook.dat** file using the following command:

```
$ wget https://raw.githubusercontent.com/mmehediha42/csc2500/master/phonebook.dat
```

Introduction

You will practice some simple regular expression using grep.

Getting Started

Grep

For this lab, your will be using a command called grep (gnu regular expressions). Grep finds patterns of strings in files. By default, when grep finds a given string pattern in a file, it prints the line in which it found the pattern. Call up a man page on grep. The synopsis of grep is:

```
grep [options] [pattern] [file...]
```

The pattern is a regular expression. For our assignment, the most important option that we will use for grep is -E. This option turns on extended regular expressions.

So, what is a regular expression? It is a pattern that describes a set of possible strings. For example, the regular expression for any string with the word John in it is just 'John'. If you want to print the lines of a file called phonebook.dat that have the word John somewhere in the line, you would give the following command:

```
grep  -E 'John' phonebook.dat
```

This lab will give you more instructions on regular expressions as you do the problems described below.

Lab Practice

Write a single bash shell script that accomplishes each of the following problems. Your shell script should print the problem number and description, and then execute the grep command that gives the answer. For example:

```
$ ./lab8.sh

1. starts or ends with Jose

Josephine

San Jose

San Jose

Jose

San Jose

San Jose

Joseph

2. start with at least twenty-seven upper or lower case characters a-z

http://www.chinesetranslationresources.com

http://www.industrialpapershreddersinc.com

http://www.woodbridgefreepubliclibrary.com

http://www.marriotthotelsresortssuites.com

http://www.italianexpressfranchisecorp.com

http://www.kaminskikatherineandritsaki.com

http://www.baileytransportationprodinc.com

and so on...
```

So how do I get my problem to show up in green in the terminal? I use echo with the -e option and escape codes. For example, my command to print problem 1 looks like the following:

```
HI='\033[0;32m'

NORMAL='\033[0m'

echo -e "${HI}1. starts or ends with Jose$NORMAL"
```

Note that I only need to define the variables HI and NORMAL once at the top of my script file, and then use the variables in my echo commands. I use curly braces for HI inside of the echo string so that the dereference (the dollar sign) knows when the variable name begins and ends (e.g. using `$HI1. starts` instead of `${HI}1. starts` would give an error).

Problems:

Using the file phonebook.dat from the link above, write commands to output only the lines with the following characteristics:

1. starts or ends with Jose
   Earlier in this lab, I showed you how to find the lines with the string 'John' appearing anywhere in the line. However, for this problem, you only want to print lines that have 'Jose' at the very beginning or at the very end. You can specify that a pattern must be at the very beginning of the line using the `^` symbol. For example, the following will find lines with 'John' at the beginning.

   ```
   grep -E '^John' phonebook.dat
   ```

   Similarly, you can specify that a pattern must be at the very end of the line by following the pattern with a `$` symbol, such as in the following example:

   ```
   grep -E 'John$' phonebook.dat
   ```

   To get lines where a patter appears at the front *or* the back of the line, you use the `|` character. In the following example, grep prints the lines with either the string 'Mike' or 'John' in them.

   ```
   grep -E 'Mike|John' phonebook.dat
   ```

   Given the above description, you should now be able to answer problem 1 on your own.

2. contain at least twenty-seven upper or lower case characters a - z.
   You can specify character classes with regular expressions using the `[]` symbols. For example, if you wanted to find all lines with a digit in them, then you can use the following command:

   ```
   grep -E '[0-9]' phonebook.dat
   ```

   You can grep from a specific number of characters using the `{}` symbols. For example, to find lines a sequence of three digits in them, you can use the following command:

   ```
   grep -E '[0-9]{3}' phonebook.dat
   ```

3. consists of more than 18 characters. The character can be anything, including alphabetic and numeric.
   To represent any single character in a regular expression, you use the period.

4. contains exactly 10 characters. The character can be anything, including alphabetic and numeric.
   To answer this problem, you only need to apply what you already know. Finding a line that contains exactly 10 characters is that same as finding a line that begins and ends with exactly 10 characters.

5. contains a sequence of between 6 and 8 alphabetic characters, i.e. upper or lower case a through z. The sequence must be separated from the rest of the line by a space or tab character (on each side)
   To match an expression if it occurs at least *n* times, but no more than *m* times, you can use {*n*,*m*}. For example, to match lines beginning with a sequence of 2 to 4 a, s, h, or i characters in any order, you can do the following:

   ```
   grep -E '^[ashi]{2,3}' phonebook.dat
   ```

   You can represent a space character in a regular expression with, you guessed it, a space. Tabs are represented with \t.

6. contains a local phone number: 3 digits, dash 4 digits
   This problem is a little tricky. To be a local phone number, the number has to be 3 digits followed by a dash and then 4 digits. However, the number cannot be preceded by a dash, because that signifies that the number has an area code in front of it. So, a local phone number is a number that has 3 digits, a dash, and then four digits and that is either at the beginning of a line, *or* somewhere else in the line but does not begin with a dash.

   You can specify that a pattern does not have a certain character by using the `^` symbol *inside* the `[]` symbols for character classes. For example, to find all lines that *do not* begin with a capital or lower case `a` through `w` or a number, you can give the following command:

   ```
   grep -E '^[^a-wA-W0-9]' phonebook.dat
   ```

7. contains a valid URL on a line by itself.
   For the purposes of this lab, a valid URL starts with http:// or HTTP:// and is followed by any sequence of upper and lower case a-z characters, a dot, another sequence of upper and lower case characters, another dot, and then `com` or `edu`.

   I will get you started on this regular expression. To make this regular expression work, your will need to use grouping with the `()` symbols. The URL can start with `http` or `HTTP`, so the beginning of your regular expression should look like so:
   ```
   '^(http|HTTP)://'
   ```

   If you do not use the `()`, then the `^` symbol will be associated with the first `http`, but not the rest of the expression that occurs after the `|`. You will have use grouping at the end of the expression when handling the `com` or `edu` also.

How do you represent a sequence of upper and lower case a-z characters? You can represent one or more of a pattern using the + symbol. So `[a-zA-Z]+` represents the sequence of at least one upper or lower case characters.

You need one more piece of information. You need to represent a dot in your regular expression. However, the '.' is a special symbol that represents any character. If you want the dot to represent a literal '.', then you have to escape it like so: `\.`

Submission

Submit your lab8.sh script on ilearn.