CS 279 - Week 9 Lab Exercise

Deadline

Due by the end of lab on 2022-10-20.

How to submit

Submit the files specified below on https://canvas.humboldt.edu.

Purpose

To practice with more BRE features and with some ERE features, to practice using the =~ operator, to practice reading all of the lines from a file, and to get a little more experience with commands `which` and `wc` (and a little piping!).

Important notes

- This exercise assumes that whoever is serving as navigator has a `bin` directory in their home directory, and that this `bin` directory has been added to their `PATH` environment variable (as was set up during the Week 7 Lab Exercise).
- Work in PAIRS for this lab exercise:
  - two people at one computer,
  - one typing (driver),
  - one saying what to type (navigator),
  - both discussing along the way!

  When done, the driver should e-mail the files to the navigator, so BOTH of you can EACH submit them.
- Assume, for all `bash` scripts in this course, that the following are required:
  - Start each script with the line that is considered good style (and is a CS 279 course requirement), that specifies that this script should be executed using the `bash` shell
  - After a blank line, put in one or more **comments** including at least the name of the shell script, your names, and its last modified date
  - And follow these comments with a blank line.

Lab Exercise setup

- use `ssh` to connect to the one of your accounts on `nrs-projects.humboldt.edu`
- make and protect a directory `279lab9` using the commands:

```
mkdir 279lab9
chmod 700 279lab9
```

- go into that directory using:

```
cd 279lab9
```

Problem 1

In a file named `279lab9-prob1.txt`, put:

- your names
- your answers to the following

1 part a

On nrs-projects, use the `which` command to determine the full pathname of the version of the `nano` program available for your use on nrs-projects. Give this full pathname in answer to this part.

1 part b

On nrs-projects, use the `which` command to determine the full pathname of the version of the `sqlplus` program available for your use on nrs-projects. Give this full pathname in answer to this part.

1 part c

Consider the `bin` directory you created in your home directory in the Week 7 Lab Exercise.

You should have at least one bash shell script in that directory -- if not, put one there. 8 -)

Then, from your `279lab9` directory, use the `which` command with the name of one such bash shell script that is in your `bin` directory, and give the pathname `which` returns in answer to this part.

1 part d

Just how many commands are available in the directory containing the `nano` program on nrs-projects?

Give a command, using a pipe and the `wc` command with an appropriate option, whose result would be JUST the number of commands in that directory.

Then, give the result of running that command.

Problem 2

In a file named `279lab9-prob2.txt`, put:

- your names
- your answers to the following

2 part a

In the Week 8 Lab Exercise Problem 3 part b, you wrote a BRE pattern that will match any line containing `CS` followed by a blank followed by a 3-digit number.

For this problem, write a BRE that includes an appropriate interval expression that will match any line containing JUST `CS` followed by a blank followed by a 3-digit number.

2 part b

Write a BRE that includes an appropriate interval expression that will match any line that includes a sequence of 4 or more uppercase letters within it.

2 part c

Write a BRE that includes an appropriate interval expression that will match any line containing JUST 5 or 6 or 7 or 8 lowercase letters.

2 part d

Write a BRE that will match any line that contains JUST a string of 1 or more `X`s followed by a dash followed by the SAME number of `X`s.

(So, for example, these should match:

```
X-X
XX-XX
XXX-XXX
```

...but these should not:

```
XX-XXX
XXXX-X
```

)

2 part e

Write a BRE that will match any line that contains a repeated string of 1 or more digits.

Problem 3

In a file named `279lab9-prob3.txt`, put:

- your names
- your answers to the following

3 part a

Write an ERE that will match any line containing `/*` or `//` or `#` .

3 part b

Write an ERE that will match any line containing JUST a # followed by one or more digits.

3 part c

Write an ERE that will match any line containing JUST a reasonable integer -- here we'll define a reasonable integer as a sequence of digits, optionally starting with a + or -. (Remember that + is special, and needs to be escaped.)

3 part d

Write an ERE that will match any line containing a sequence of digits, optionally containing a single decimal point at any position (including the beginning of the end). (Remember that . is special, and needs to be escaped.)

3 part e

Write an ERE that will match any line containing JUST a reasonable number -- here, we'll define a reasonable number as a sequence of digits, optionally starting with a + or -, and optionally containing a single decimal point. (In this case, we'll accept a decimal point at the end or beginning as well -- `.3` and `9.` for example.)

3 part f

Fun fact: character class `[[:alpha:]]` matches any alphabetic character.

Write an ERE that will match any line containing JUST an acceptable file name with the suffix `.cpp`, the suffix `.h`, or the suffix `.txt`. (Assume for this problem's purposes that an acceptable file name contains one or more alphabetic characters, digits, underscores, and/or dashes before one of those suffixes.)

Problem 4

A researcher wants to study features of students' bash shell script comments. They would like the following shell script to help them.

Write a script `grab-comments` or `grab-comments.sh` that meets at least the following specifications (although you can add more as well, as long as you include at least the following):

- It should expect one command-line argument, and should complain to the screen and exit with a non-zero exit status if given any other number of command-line arguments.
- It should verify that the command-line argument is indeed a regular file, and complain and exit with a different non-zero exit status if it is not.
- If it gets past the above, it should append the string `file:` followed by the name of the command-line argument file to a file `saved-comments.txt` in the current working directory.
- It should try to read all of the lines in the command-line argument file, and for each that contains a #, it should append that line to that file `saved-comments.txt`.
  - NOTE: I had to precede # with a backslash when I wrote my version of this.

Submit these files to Canvas:

- `279lab9-prob1.txt`
- `279lab9-prob2.txt`
- `279lab9-prob3.txt`
- `grab-comments` or `grab-comments.sh`

BOTH of you should then submit copies of these problems' files to Canvas for this lab exercise.

Once both of you have submitted these lab exercise files, you may leave lab if you wish. Or, you can ask questions, read the course text, etc. But note that questions about today's lab exercise will get first priority.