CSE 391, Winter 2020
Assignment 2: More Unix Shell
Due Tuesday, January 21, 2020, 2:00 PM

This assignment continues to practice using the `bash` shell and basics of combining commands using redirection and pipes. For Task 0 there is nothing to submit. For Tasks 1 and 2 you will submit your responses to Gradescope.

Some parts of this assignment depend on compiling and running Java programs from the command line. Many distributions of Linux do not include Sun's Java Development Kit (JDK). You may need to install JDK or use a different Linux machine that already has JDK installed, such as the CSE Virtual Machine (VM), CSE basement lab machines or the shared `attu` server. See the course web site for directions about how to install JDK on your own Linux machine.

Task 0: Log in and prepare a directory

First, **log in to a machine running Linux** and launch a **Terminal** window as described previously in Homework 1.

We have set up a ZIP archive full of support files that you must download to your Linux machine. Download/unzip it to a directory on your system.

```
wget http://courses.cs.washington.edu/courses/cse391/20wi/homework/2/hw2.zip
unzip hw2.zip
```

Task 1: Basic Navigation in a Text Editor

The following are exercises and questions are meant to help you become more comfortable with a text editor that is built into the command line. You can choose either vim or emacs. While the answers to the questions themselves are relatively easy to find by simply looking them up, **the real learning will come from you actually practicing these commands yourself**. While we won't be able to know whether you've really been practicing, this is not for our benefit, it's for yours. We also recommend getting even more practice by writing the answers to your `task1.txt` and `task2.sh` files using this editor ☺

For all of these questions, you will be using the text editor of your choice to practice navigating around the `intro_survey.csv` file in the `hw2` directory.

1. What is the command to move your cursor to the end of the file?
2. What is the command to move your cursor back up to the beginning of the file?
3. What is the command move to line 50 of the file? Practice moving around to different lines in the file. You may notice that emacs and vim do not show line numbers of a file by default but you can enable line numbers. While it's not required that you look this up, you may find it helpful.
4. What is the command to search for the text "chocolate"?
5. What is the command to go to the next match from your search?

Task 2: Bash shell commands

For this task you will use shell commands to process the class' responses to the introductory survey completed last week. The results are stored in the `hw2` folder in a file called `intro_survey.csv`. As you may notice, only four of the questions from the survey have been included: "What's your favorite candy?", "What's your favorite restaurant on the ave?", "Cats or Dogs?" and "Are you currently enrolled in CSE 351?" One of the goals of this assignment is to show you that a few simple commands can be used to process and analyze data.

For each item below, **determine a single `bash` shell statement that will either perform the operation(s) requested or provide the answer to a question**. Each solution must be a one-line shell statement, but you may use input/output redirection operators such as >, <, and `|`. For all commands, do not create any files except those indicated.

(Continued on next page)

To test your commands, you should have unzipped `hw2.zip` into the current directory. You can assume you are in the `hw2` directory when doing these problems.

Write your answers in on the indicated lines in the `task2.sh` file in the `hw2` folder.

1. Compile the Java program stored in the file `ParseColumn.java`.
2. The `ParseColumn` java program takes a column number (1-indexed) as a command-line argument, reads a csv file from stdin line by line and outputs the data from the specified column only. Run the Java `ParseColumn` program stored in `ParseColumn.class` to select columns from `intro_survey.csv` containing the answers to "What's your favorite candy?" and write the output to a file called `candies.txt`. You may use the resulting `candies.txt` file for later problems.
3. Display each of the responses from `candies.txt` that contain the text "`chocolate`", ignoring case. That is, all of the favorite candies that have chocolate as part of the response.
4. Display each of the responses from `candies.txt` that **DO NOT** contain the text "`chocolate`", ignoring case. Don't worry about removing the header "What's your favorite candy?" from the output.
5. Create a new file called `intro_survey_no_header.csv` containing the entire contents of the original `intro_survey.csv` file, without the header. (The header is the first line in the file containing the survey questions.) Note your command should be flexible, regardless of the specific number of lines in the `intro_survey.csv` file. Read the `man` pages for the `head` and `tail` commands for help on how to do this. You may use the resulting `intro_survey_no_header.csv` file for later problems.
6. How many students completed the survey?
7. Assuming that the intro survey lists responses in chronological order, what was the favorite restaurant of the last person to turn in the survey?
8. How many students in the course are not also taking CSE 351? Note that because the answers to the question "Are you currently enrolled in CSE 351?" are "Yes" and "No", you won't be able to simply grep `intro_survey.csv` for "No", as this word may appear as part of answers to other questions.
9. How many unique answers were there to question "What's your favorite candy?" case-insensitively? Don't worry right now about the difference between answers like "kit kat" and "kitkat", but your solution SHOULD correctly count "kitkat" and "KITKAT" as the same. This time you should be careful that the header "What's your favorite candy?" is not counted.
10. *Challenge question:* How many students prefer cats, dogs or neither? Write the command that outputs the count along with the preference, for each preference, as in

    ```
    XX Cats
    XX Dogs
    XX Neither
    ```

    Note that the leading whitespace is expected.