Assignment 5

Microproject

Write a Python program which takes a single argument — a directory name. Your program will, for every file in the directory, count the number of lines in that file **using the Unix command wc**. Then, write to the console each file name and its number of lines.

Main Project

Write a Python program that collects, summarizes, and e-mails all the programming assignments for this course.

You should begin by organizing your assignments in a folder structure which will make this easier. I recommend something like:

- csc344
  - a1
  - a2
  - a3
  - a4
  - a5

Assuming the programs are in subdirectories a_1, a_2, …, a_5 of directory csc344, your program should:

- create a summary_a_i.html file for each a_i, which contains (in reasonably formatted, valid HTML):
  - the name of each source file (linked to the file itself), along with the number of lines long the file is, and
  - an alphabetized list of all identifiers used in the program (class, function, rule, variable etc. names), omitting duplicates. **You do *NOT* need to specially filter keywords or builtin functions**. *Even though something like "for" in C isn't really an identifier, we'll be OK with including it. We won't be OK with including punctuation which isn't part of names, nor will we be ok with commented text.*
- create a valid HTML web page in the csc344 directory called index.html with links to each of

the summary files;
- create a tar.gz file containing all assignment sources, but excluding non-sources (executables, .class files, etc). Also included should be the html files created above. Be sure the links in the webpages work after extraction;
- prompt the user for an email address and send the tar.gz file.

An Important Note…

You *must* run your project on altair and send mail from your @cs.oswego.edu account. You can send mail from the terminal using the `mutt` command. A little Googling and looking at the man pages should tell you how to use it.

A successful demo will involve sending the email and me receiving it, extracting it, and being able to view your pages/code following the links on your page.

If you Haven't Finished Every Project…

That's OK! You can still do this one. If you've finished projects, you must use your source code as submitted. If you haven't, use the following:

- C: Source code for the wc command (https://www.gnu.org/software/cflow/manual/html_node/Source-of-wc-command.html)
- Clojure: The Monty Hall simulation (https://github.com/digitalneoplasm/MontyHallSimulation) code
- Scala: The class example (https://danielschlegel.org/wp/teaching/recursive-descent-parser-in-scala/) code
- Prolog: The Farmer, Goat, Wolf, Cabbage (https://danielschlegel.org/wp/teaching/prolog-farmer-goat-wolf-cabbage/) code

Extra Credit

There are several opportunities for extra credit on this assignment, such as generating XML or JSON files instead of HTML, and displaying them using XSLT or JavaScript, respectively. If you're interested, let me know and we can discuss further.

Dev Environment / Useful Resources

PyCharm (https://www.jetbrains.com/pycharm/)
Regular Expressions in Python 3.6 (https://docs.python.org/3/howto/regex.html)
Regular Expression Debugger / Tester (https://regex101.com/)