CSCA20 - Lab 6

File Input/Output

SIMPLIFIED VERSION

Learning Objectives

We're taking a break from learning about new features of python to focus on learning new ways of making our code interact with the world. In this lab, we can review a lot of what we've done already, but also put it to the test with much more data.

Prelab

Your prelab for this lab is your completed lab for lab 4. So take some time to make sure you completed lab 4 as thoroughly as possible, and if you had trouble with lab4, this is a good opportunity to ask for help.

Demonstration & Evaluation

This lab demonstrates the skills of User I/O, Selection, Lists & Strings, and File I/O.

The Scenario

The Union for Tracking Seaborn Contaminants (UTSC) was so pleased with the work you did for them in lab 4, they've asked you to improve the system. The program has grown, and now they have many rigs and much more data. So simply having users type everything in isn't going to cut it anymore.

Menu

The same general menu should be presented as before, users should be able to add data, generate a report on totals, or generate a report on totals collected, or overload days

Adding Data

This time, instead of inputting data directly, the program should ask the user for the name of a text file (located in the same directory as the python file) containing the data to work on, and the rig number for which you want to collect data.

Input Files

The first line of the file consists of the number of rigs (ocean cleanup vessels) the project currently has.

The rest of the file consists of a line with the day number, followed by the amount of plastic collected by each rig on that day (one rig per line). The rig number will always be followed by a comma, and then the amount of plastic collected.

On some days, rigs are out of commission for repairs or testing. On those days, the rig's total is replaced by the letters OOC.

A sample input file has been provided for you.

Calculating Totals

This should be the same menu options as in the previous version of the code (All dates or range of dates), but instead of just printing to the screen. The code should prompt the user for the name of a file, and append (not write, if the file already has data, add to it don't overwrite) the output to that file. It should add a line that looks something like:

```
RIG 4,25
```

(if you asked for rig 4, and it had picked up a total of 25 kg of plastic in the relevant days

Calculating Overload Days

This should be the same as totals (including asking for an output file, etc), but instead of the total plastic collected beside the rig in the output file, it should append the number of overload days.

Hints

Here are a few hints that might help you with this assignment:

- Start by creating the code to read the file and get the various parts, just read, split, and then print out the data before you worry about calculating or storing anything
- speaking of split... the split function might be helpful here, check the python docs
- You can assume that the number of rigs provided at the start of the file is correct, that all rigs report every day (even if the report is OOC) and that they are always in exactly that order
- You don't need to read the data file right away. You may find it easier to just remember the file name, and only read it once you have the date range/threshold values

Extra Practice

You can get away with reading the file over and over again for each query the user asks, but try to read the data once, when the user chooses to add the data, and then store it for use in multiple reports. (As a hint, you'll need a list of lists where each inside list represents the data for one rig).