CSCA20 - Lab 7

Sets & Dictionaries

Learning Objectives

This week, we're introducing sets and dictionaries. You COULD do everything in this lab with lists (you wouldn't get marks for it, but it would theoretically be possible), but you would have a lot more work to do, and your code would be difficult to update and maintain. Sets and Dictionaries will make things much easier.

Prelab

In order to make the most of your time in the lab, you should arrive with code that already reads and splits up the data file. A good way of testing that is to build a program that lists all of the parties, districts and votes as they appear in the input file. e.g., `In Fredericton, the LIB party received 16316 votes`

Demonstration & Evaluation

This lab demonstrates the skills of: User I/O, Selection, Loops, Loops + Selection, Functions, Testing, Documentation, Strings/Lists, File I/O, and Dictionaries.
This lab will also demonstrate internal documentation... the code is starting to get more complicated now, and you really can't be just jumping into the code without planning first. Your TAs will be evaluating you on whether you are building code as we teach you in lecture: planning your code first with comments, and then filling in the code afterwards.

The Scenario

The Ultimate Tally of Simultaneous Casting (UTSC) wants your help building a tracker for election voting across Canada. As districts report votes, they need to be able to validate the results and compute totals of votes cast per party.

Data Files

The first line of the input file will consist of a line containing all of the party labels, separated by commas.
The rest of the file will consist of the word REPORTING: followed by the district name on a line, and then 1 or more lines consisting of a party label and the number of votes received by that party.

Validating Files

In the chaos of election night, it's possible that mistakes are made in the input files. Your code should validate the input when it is read. If the file contains reports from a district that has already reported, you should let the user know. And if votes are reported for a party whose label was not listed at the top of the file, you should also report that error to the user.
Your code should maintain a set of reporting districts and party labels, and report any errors (districts reporting twice, or votes for a party label that doesn't exist) to the user.

Menu

You should produce a menu that allows the user to validate files, add data from files (i.e., the user can continue to add new files as more districts report throughout the night). UTSC also wants to know the total votes cast per party[1]. The user should also be able to generate a report of the total votes for each party, or to enter a specific party label and get the total number of votes for that party.

Functions

Our code is getting far to complex to have in a single block, so you'll need to use functions to break up your code. Make sure your functions are properly documented. We won't force specific function names/specifications on you, but you should at least have functions to:

- Validate a file
- Read a file and add its data to a dictionary

[1]We tried to explain to them that in Canada, we have a first-past-the-post system so total votes cast are meaningless, but they just looked at us funny and frankly, we were too embarrassed by this system to push the matter any further

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
- Your first order of business should be to generate a set of party labels, and a dictionary mapping party labels to scores (initially set to zero), you can do this after reading the first line of the file
- Break down the problem. Separate the problem of reading the various parts of the file from the problem of generating the sets/dictionaries, from the problem of checking the districts for duplicates, from the problem of counting the scores. Each of these problems on their own are manageable, trying to tackle all of them at once is very difficult.
- You can't assume that each file will have the same set of party labels, consider the problem of what happens when a new file comes in that has a party label you haven't seen before.