

Problem 13. Census Summary

Program: `census.py`

The file `censuspopdata.csv` contains data from the 2010 US Census, and you have the annoying task of going through its thousands of rows to count both the total population and the number of census tracts for each county. (A census tract is simply a geographic area defined for the census.) Each row represents a single census tract.

Write a program named `census.py` that reads data from the given file and calculates the statistics for each county.

The program should read data from the CSV file, count the number of census tracts in each county, calculate the total population of each county, and allow the user to enter filter expressions to select specific summarized information. The types of filters that should be accepted by the program are of the following format:

- `<state>/<county>/pop`: this will display the population of the given county at the given state.
- `<state>/<county>/tracts`: this will show the number of tracts of the county at the given state.
- `<state>/pop`: this will show the total population of the state.

The output of the program should like the following:

```
% python census.py

Welcome to 2010 US Census Summary
Type a filter or at any time, type
'quit' to quit.

Enter a filter expression: AK/Anchorage/pop
The population of Anchorage(AK) is 291826

Enter a filter expression: AK/Anchorage/tracts
The number of tracts in Anchorage(AK) is 55

Enter a filter expression: AK/pop
The population of AK state is 710231

Enter a filter expression: quit
Thanks for using this app! Goodbye.
```

FIGURE 1: SAMPLE OUTPUT FOR PROBLEM 13

Problem 13. Appraisal Summary

Program: `appraisal.py`

Alan Key is evaluating a number of staff. In total, there are 10 attributes for each staff that he wants to report. He is using Excel to keep the data. For each staff he has created a separate sheet. After completing the evaluation of 11 of his staff, he has realized that generating the summary report will involve a lot of manual copy/past work. Using what you learned about Python and Excel, help Alan create the summary report in few seconds.

Create a program named `appraisal.py` that summarizes the data given in `appraisal.xlsx`. The program should iterate over the staff sheets and copy the required summary data to appropriate place in the Excel file.

Acknowledgements

Preparation of this problem set would not have been possible without adaptation from (Sweigart, 2019). The author gratefully acknowledges the work of the authors cited while assuming complete responsibility for any mistake introduced in the adaptation.

References

Sweigart, A. (2019). *Automate the Boring Stuff with Python* (2nd ed.). No Starch Press.