Project #8 – Realty Sales Processing

1 Objective

Read data from a text file. Write data to a text file. Use loop algorithms like counters, accumulators, plus finding minimums and maximums.

2 Problem Overview

Reanna’s Realty maintains a simple text file that records sales data for the month. They want you to write a Python program that reads and summarizes that data, giving them some basic information like a count of sales, the total sales and commission, and the smallest and largest sales for the month. This information will be written into a summary file.

3 User Interface

3.1 Input

There is no user input. A sample file sales file is provided for you (Proj8-Sales.txt). Please use exactly this file name, or the program will not run correctly for grading.

3.2 Output

The output of the program should be a summary file (Proj8-Summary.txt). Again, please use this exact name. Its contents should look like this (with correct numbers, of course):

```
Number of sales             11

Total sales       $ 1,234,567
Total commission  $   123,456

Smallest sale     $    98,765
Largest sale      $   987,654
```

While this code isn’t “chatty” with the user, you should tell them that the processing is complete. Also, tell them how many records were processed, and to check the summary file for the complete report.

4 Calculations

You’ll need some simple loop algorithms including counters and accumulators. You’ll need to look for the smallest and largest sale that month.

5 Functions

A single function (main) is probably enough here. If you wish to write helper functions, you may do so. You shouldn’t use any global variables, though, so design well.

6 Additional Requirements

- *Make only one pass through the file*, i.e., don't close and re-open it and loop through it again. File I/O is less inefficient than in-memory processing, so you want to do all your work in a single pass whenever possible.
- *Use no lists in this project for storing whole lines of data*; handle data as it comes in. For splitting a line of text ("record") into its component fields, then of course a list okay, as that's what is returned from the split function.
- *Use no global variables* in your code. This is a good rule of thumb anyway.

7 Code Specifications

- At the bottom of the program you should have a call to main().
- Include header comments at the top of each file. Include your name, the date, and a brief description of what the program does.
- Include comments for each section saying what's going on in the lines of code below.
- Use comments elsewhere as you think they help guide the reader. Don't overdo, though! Not every line needs a comment; think about describing a block of related code.
- Use blank lines to separate sections and provide visual "breathing room."
- Use descriptive variable and function names.

8 Hints

- Make a backup copy (or two) of the sales file in case you mess it up somehow.
- The looping algorithms you'll need are all described in the chapter on iteration.
- Get the program working with on-screen display before writing to the file.

9 Testing

- Develop an appropriate number of test cases.
- Document your testing and results in comments at the bottom of the program as shown below.

10 Summary

At the bottom of your program, add comments that answer these questions:

- How did you approach this assignment? Where did you get stuck, and how did you get unstuck?
- How did you test your program? What doesn't work as you'd like, perhaps things that you'd like to fix as you learn more?
- What did you learn from this assignment? What will you do differently on the next project?

11 Extra Credit

For extra credit, at the top of the summary file (do you see why it must go there?), also include commission totals *for each realtor*. Each realtor's data is together in the file. ~~The file is sorted by realtor~~, so this should give you some clues as to the algorithm you should consider. Also show the realtors with the highest and lowest total commission for the month.

Extra credit code must not do additional passes through the file to accomplish these additional tasks.

12 Grading Matrix

Area	**Pct**
Sales data read	20
Totals calculated correctly	20
Min and max found correctly	20
Summary file written	20
Test cases	10
Comments, variable names, white space	5
Summary report	5
Extra Credit	5
Total	**105**