COSC 010 Final Project

I. Project Overview

- For this project, you will investigate working with data. The aim is to create a program that read and write files for the purpose of data analysis.
- This project provides an opportunity to practice with ***file input/output*** in Python.
- You will use a "real world" data set (in csv form). You can choose any data set from data.gov (https://catalog.data.gov/dataset) or you can use your own data. If you choose to use your own data, note that, while you cannot just make up a data set (it must be more or less "real"), you can modify it to make it appropriate for a class project submission. Of particular importance: do not use any data that is sensitive or confidential.
- Your program must allow the user some opportunity to look up certain information in the data.
- For the final part of the project (Part 7), you will see that you have two options. I recommend that you choose Option B if you feel up for a challenge (and like pretty graphs).
- As in previous Python programming assignments, there are always many ways to accomplish tasks. You must use the required elements (detailed in Section III), but beyond that you can be creative.
- Remember, if you want to do it, Python can do it. A key learning objective throughout this course is learning how to be resourceful. If you don't know how Python can help you do something, use the Web to search for guidance. That said, you are not allowed to copy and paste a solution that you find on the Internet or anywhere else.
- ***Note: for this project, you are <u>not allowed</u> to use numpy or pandas or any other third-party Python libraries, except for in Part 7.***

II. What You Will Submit

Item 1: The Python Program

You will submit your program as a .py file, named ***LastNameFirstName.py.*** (Replace "LastNameFirstName" with your actual last and first names). Be sure to include plenty of comments (the lack of good comments will result in a penalty of up to -5 points).

Item 2: The Input File

You will submit the input file as a .csv file, named ***LastNameFirstName.csv*** (Replace "LastNameFirstName" with your actual last and first names)

Item 3: The Project Report

1. The first page of your project report will contain an explanation of what your project does and how it works, including:
   - Overview of your program and the data set that you chose
   - Expected inputs and outputs
   - Detailed description of how you used each of the required computing concepts
   - Limitations (e.g., do certain user inputs "break" your program?)
   - Future expansion options.

   All this will help a person to understand your project. You may use screenshots in the explanations.
2. The second page will contain a brief personal reflection that addresses the following questions:
   - Which of the two options did you choose for Part 7, and why? Please be open and honest about your experience working through this final project at the end of this semester of virtual learning.
   - What topic did you find most challenging this semester? Describe your experience learning that topic, including how confident you feel with that topic now at the end of the semester.
   - What topic did you find most interesting and/or useful this semester? Explain your choice.
   - What project did you find most interesting and/or useful this semester? Explain your choice.
3. The third (and more as needed) page(s) will contain three (3) examples of program use – including input and output. You can screenshot these if you wish.
4. The remaining pages will contain a copy of the Python code (***You will also submit the code as .py***).

III. Project Specifications and Required Elements

It is best to do this project incrementally. Make sure that each part works before adding on the next part. To help with this, I recommend that you follow the following steps:

1. The first step is to create a .csv file in Excel (or a similar application). You should name this file ***LastNameFirstName.csv*** (Replace "LastNameFirstName" with your actual last and first names). It must have **at least** 30 rows and 6 columns. More is better. Each column represents a different variable, and the columns must have the variable names stored in Row 1. At least 3 of your variables must be numeric. When you submit this project, you will also submit this .csv file.

   EXAMPLE. (Note that, unlike the file that you will make, this example does not have 20 rows. This picture is just an example of placing data into Excel and saving as .csv.

	A	B	C	D	E	F
1	StockTicker	Name	YearHigh	YearLow	MarketCapBillions	
2	AAPL	Apple	183.5	138.62	913.2	
3	VZ	Verizon	54.77	42.8	198.5	
4	TWTR	Twitter	35.84	14.12	25.6	
5	T	AT_T	42.7	32.55	229.6	
6	GOOG	Google	1186.89	803.37	791	
7	FB	FaceBook	195.32	137.6	528.4	
8						

   Note: Be sure to save your data set it in the same location (folder) as your project's .py file.

2. Your .py file needs first to read in the data (into what is called a *data frame*) and print out the data frame so that the user can see it.

   You should include the following import statement at the very top of your program:

   import csv

   EXAMPLE OF READING IN DATA AND PRINTING IT (Test out the two ways of printing):

```
DataFileName="StockDataSet.csv"
with open(DataFileName, "r") as csvfile:
        MyStockDF=csv.reader(csvfile)
        list_of_rows = [r for r in MyStockDF]
        for row in list_of_rows:
            # print(row)              # This would result in ugly format
              print(", ".join(row))   #This will result in a nicer format for the output
```

3. Next, as proof-of-understanding, you will print out specific portions of the dataframe.

    a) Print only the third row (row 2 per Python) in your dataset.

    b) Then print only the first column (column 0 per Python) in your dataset.

    c) Then print just the value in row 0, column 4 (per Python, where row 0 is the first row).

4. Calculate and print the mean (average), median, max and min for *three* of your numerical variable data. Round all answers that you print to two decimal places.

5. Create a new file (using your Python program - not by hand) called ***output.txt***. Open this file and write into it everything you printed in **number (4) above.** Your program will use ***"with open…"*** to open/create a new file and will write the mean (average), median, max and min for any *three* of your numerical variable data as in (4) above. Round to 2 decimal places.

6. Next, create a new function named **AppendFunction**. This function will *take two parameters* and *return one value*. Its first parameter is the filename: *output.txt* that you used above. The second parameter is the data frame that you already read in from your .csv data file.

    Example of function signature:

    def AppendFunction(filename, dataframe):

    This function opens the output.txt file and appends your ENTIRE data frame into the file. It will not remove or delete what you have already written to the file.

    The function must then return a list containing the number of characters in each line of the current output.txt file as well as (the last item in the list) the total number of characters in the entire file. (If there are N rows in output.txt, there should be N + 1 elements in this list). Take careful note of the following:

    - The sum of the characters in each line should equal the total number of characters in the entire file.
    - **The "newline", (\n), counts as one character**. You cannot see a newline if you open the file and look at it - but it's there.
    - You will call this function from main() and you will **print out the returned values FROM MAIN** (the number of characters in each line of the output.txt and the total number of characters in the output.txt file).
    - Your program needs to count all the characters in the file **after** it writes the entire data frame into the file.
    - Your print statements should be user-friendly, for example: *The total number of characters in the output file, called output.txt is [whatever it is]*.

7. You have two options for this final part. Option A is relatively straight-forward (you already have all you need to know to complete it), but Option B will require you to do a bit of reading on how to create graphs (http://drgates.georgetown.domains/GatesPythonIntroBook/Chapter9Gates.pdf).

   Option A:
   Create a new function named **UserChoice**. This function will take the dataframe as a parameter, and outfile.txt as a parameter. This function will *print out for the user the names of the variables* (the column names) for your data. Recall that your data is in your dataframe.

   Next, this function will *ask the user to choose any variable name* that represents numerical data.

   Based on the user choice:
   a) Print the mean, median,, max and min for the chosen variable.
   b) Write the mean, median, max and min into output.txt (APPEND).

   Option B
   Create a function named **MakeGraphs**. This function will take the dataframe as a parameter. It will use the data to generate four graphs, both printing them out and writing each to its own .jpg file.

   You will need to import the matplotlib.pyplot library with the following import statement:
   import matplotlib.pyplot as mpp

   Specific types of graphs required:
   - Scatterplot
   - Boxplot
   - Line plot (which is a basic plot)
   - Pie or bar graph
     - Note that, to make a pie or bar graph, you need to properly prepare the variables, etc. Google this and/or view the examples in the textbook. This can take a little time as you need to think about and understand what you are doing.

   EXAMPLE. (Note that this example shows a function and *some* code for *one* graph.)

```
def MakeGraphs(df):
    imagefilename="FinalGRAPH1.pdf"   #Note: this example creates a .pdf file
    fig=mpp.figure()
    mpp.scatter(df["YearHigh"], df["YearLow"])
    title ="Scatterplot for YearHigh and YearLow"
    fig.suptitle(title, fontsize=20)
    mpp.xlabel("Year High", fontsize=18)
    mpp.ylabel("Year Low", fontsize=16)
    mpp.savefig(imagefilename)
    mpp.show()
```

IV. How to submit

- Submit on Canvas
- Upload your Python Program as ***LastNameFirstName.py.*** (Of course, you will replace "LastNameFirstName" with your actual last and first names.)
- Upload your input file as a .csv file, named ***LastNameFirstName.csv***
- Upload your Project Report as ***LastNameFirstName.pdf***
- ** DOUBLE CHECK your submission. If the grader needs to email you, the penalty is -30%.