Programming Project 07

This assignment is worth 50 points (5.0% of the course grade) and must be **completed and turned in before 11:59 PM on Monday, October 29, 2018.**

Assignment Overview
(learning objectives)
This assignment will give you more experience on the use of:

1. Lists
2. iteration
3. string

The goal of this project is to analyze TSA claims data.

Assignment Background

Did you know that claims can be filed against TSA? Sometimes the US Terminal Security Agency (TSA) makes mistakes. People can get hurt and property can be damaged, lost, or stolen. Claims are generally filed against TSA for personal injuries and lost or damaged property during screenings and they keep records of every claim!

In this project you will be analyzing the claims made to TSA.

The `tsa_claims.csv` file contains a lot of data, but we are interested in Date Received (column 1), Airport Name (column 4), Airport Claim Amount (column 9), Status (column 10), and Close Amount (column 11). Note: column count starts at zero.

We will be calculating total number of approved, settled, and denied application against the TSA. Take a look at the data: what is the difference between 'approved' and 'settled' cases? We will also calculate average settlement amount (close amount). Next we will find the airport with the largest claim against the TSA. Finally, we will plot the data for accepted, settled and denied cases for each year from 2002 to 2009.

Project Description

Your program must meet the following specifications:

1. At program start prompt the user for a file to be analyzed
2. You have to use the six functions in the provided proj07.py skeleton. You have to implement and use the
    a. `open_file()`: Returns the file pointer to the file opened by asking user for the file name. Error checking is required.
    b. `read_file(fp)`: Takes a file pointer as an argument and returns a list of tuples of data in the file. You need to make sure that data for each column of interest (1, 4, 9, 10, 11) exists for each row. If any of that data is missing in a row, ignore that row. Also, we are only interested in data from 2002 to 2009 so ignore data from other years. Your data should only contain data of interest from columns (1, 4, 9, 10, 11), in that order, with the two amounts converted to floats. Other values should be strings with leading and trailing white space stripped.
    IMPORTANT: look at amounts that are greater than $1000. What do you notice? You

need to fix that. (Why do you think they did that?) That is, you need to clean up the amounts before you convert to a float. (Note that all valid amounts have a dollar sign, decimal point and cents—you do not need to check for that.)

c. `process(data)`: Takes the data from `read_data` function as an argument. You need to calculate
   i. Total: The total number of applications that are approved, settled, or denied. Note that there are other types of applications and they don't get counted in this total.
   ii. Average: The average close amount, but **only** for **non-zero** close amounts in **approved or settled** applications (not denied).
   iii. Max_claim: The largest claim amount (it is amazingly big!)
   iv. Max_claim_airport: The aiport that had the largest claim amount

   You will also create three lists
   i. List1: total cases (approved + settled + denied) for each year from 2002 to 2009.
   ii. List2: total settled + approved cases for each year from 2002 to 2009
   iii. List3: total denied cases for each year from 2002 to 2009

   Return a tuple: (List1, List2, List3, Total, Average, Max_claim, Max_claim_airport)

b. `display_data(tup)`: Takes the tuple of data calculated in the `process` function. Returns nothing. See the sample output for the expected formatting. The header is in the skeleton code provided to help with formatting. Finally, the average settlement is placed in a field of size 10, left justified, including both commas and a decimal point.

c. `plot_data(List1, List2, List3)`: This function is written for you. You just have to pass the correct lists for accepted, settled and denied cases. There should be 8 entries in each list corresponding to total accepted, settled and denied cases for each year.

d. `main()`: Takes no input. Returns nothing. Call the functions from here. Only call `plot_data` if the prompt returns "yes".

Assignment Deliverable

The deliverable for this assignment is the following file:

`proj07.py` – the source code for your Python program

Be sure to use the specified file name and to submit it for grading via the **Mimir** before the project deadline.

Assignment Notes

1. To clarify the project specifications, sample output is appended to the end of this document.
2. Items 1-9 of the Coding Standard will be enforced for this project.
3. We provide a proj07.py program for you to start with.
4. You do not need to use dictionaries for this project, but you are allowed to.
5. If you "hard code" answers, you will receive a grade of zero for the whole project. An example of hard coding is to simply print an average rather than calculating an average and then printing the calculated average.

Test Cases

Test 1 (tsa_claims_small.csv)

```
Please enter a file name: tsa_claims_small.csv
TSA Claims Data: 2002 - 2009

N = 52

            2002    2003    2004    2005    2006    2007    2008    2009
Total          8       5       1       7       9       2       9      11
Settled        8       4       1       5       6       1       7       6
Denied         0       1       0       2       3       1       2       5

Average settlement: $191.63
The maximum claim was $20,000.00 at Tri-Cities Airport

Plot data (yes/no): no
```

Test 2 (tsa_claims.csv)

```
Please enter a file name: xxxx

File not found. Please enter a valid file name: yyyy

File not found. Please enter a valid file name: z

File not found. Please enter a valid file name: tsa_claims.csv
TSA Claims Data: 2002 - 2009

N = 125,286

            2002    2003    2004    2005    2006    2007    2008    2009
Total        690  15,847  24,816  22,084  17,640  17,376  15,245  11,588
Settled      536  11,257  16,944  11,522   8,054   5,502   3,190   2,673
Denied       154   4,590   7,872  10,562   9,586  11,874  12,055   8,915

Average settlement: $196.79
The maximum claim was $3,000,000,000,000.00 at John F. Kennedy International
Airport

Plot data (yes/no): no
```

Test 3 (tsa_claims.csv)

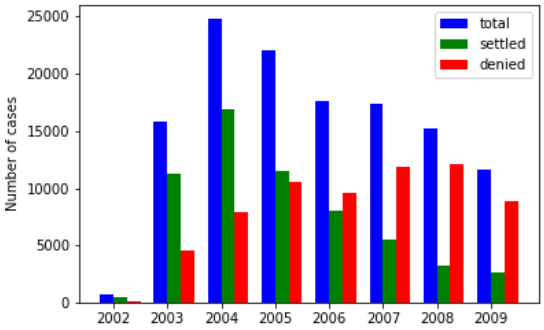
```
Please enter a file name: tsa_claims.csv
TSA Claims Data: 2002 - 2009

N = 125,286

            2002    2003    2004    2005    2006    2007    2008    2009
Total        690  15,847  24,816  22,084  17,640  17,376  15,245  11,588
Settled      536  11,257  16,944  11,522   8,054   5,502   3,190   2,673
Denied       154   4,590   7,872  10,562   9,586  11,874  12,055   8,915

Average settlement: $196.79
The maximum claim was $3,000,000,000,000.00 at John F. Kennedy International
Airport

Plot data (yes/no): yes
```



Grading Rubric

```
Computer Project #07                                  Scoring Summary
General Requirements:
  ( 5 pts) Coding Standard 1-9
             (descriptive comments, function headers, etc...)

Implementation:
  ( 5 pts) open_file function (no Mimir test)
           -2 Did not use try/except
  ( 7 pts) read_file function
  ( 8 pts) process function
  ( 10 pts) Pass Test1 tsa_claims_small.csv (implicitly tests display_data)
  ( 10 pts) Pass Test2 tsa_claims.csv(implicitly tests display_data)
  ( 5 pts) Pass Test3 Draws Bar Chart (no Mimir test)

Note: hard coding an answer earns zero points for the whole project
      -15 points for not using main()
```