Problem 1: Endings

Find on the D2L two datafiles, "namesBoys" and "namesGirls", containing the 1000 most popular baby names in the United States in 2010. Define a function that reads the two datafiles into memory storing them in two lists. For each letter, a through z, count the number of times a names ends in a particular letter. Finally, print the counts to the console in well formatted and sorted columns, so that the data line up nicely. For example...

```
Ending      Boys        Girls
a             12           19
b              7            3
c             15           26

…

…
```

Note that those numbers are made up, and only provided to offer an example of formatting. Finally, include as a comment at the end of your code a paragraph discussing any interesting differences you found in the comparison of the endings of boys and girls names.

Submission: Submit a single .py file containing all the code to the D2L. Do not zip or archive the file. Your code must include comments at the top including your name, date and the honor statement, "I have not given or received any unauthorized assistance on this assignment." Each function must include a docstring and be commented appropriately.

Grading: There is no partial credit for this problem. The code must compile, run without errors and satisfy all requirement to receive full points.

Problem 2: Stem and Leaf

```
                  Stem and Leaf Plot
        4 │ 1
        5 │ 2   7   8
        6 │ 5   6
        7 │ 0   5   8   8   8
        8 │ 0   0
        9 │ 5
```

"A stem-and-leaf display or stem-and-leaf plot is a device for presenting quantitative data in a graphical format, similar to a histogram, to assist in visualizing the shape of a distribution. They evolved from Arthur Bowley's work in the early 1900s, and are useful tools in exploratory data analysis. Stemplots became more commonly used in the 1980s after the publication of John Tukey's book on exploratory data analysis in 1977. The popularity during those years is attributable to their use of monospaced (typewriter) typestyles that allowed computer technology of the time to easily produce the graphics. Modern computers' superior graphic capabilities have meant these techniques are less often used."

> --Wikipedia (retrieved 9/23/2018)

Find on the D2L three "StemAndLeaf" datafiles. Write code that….

a) Greets the user.
b) Asks the user to input a 1, 2 or 3.
c) Given the input, reads in the appropriate datafile and displays a stem and leaf plot. (Note: This will require you to make several design decisions. I am not looking for a specific format, but the final format should be well thought out.)
d) Loops until the user wishes to exit.

Submission: Submit a single .py file containing all the code to the D2L. Do not zip or archive the file. Your code must include comments at the top including your name, date and the honor statement, "I have not given or received any unauthorized assistance on this assignment." Each function must include a docstring and be commented appropriately.

Grading: There is no partial credit for this problem. The code must compile, run without errors and satisfy all requirement to receive full points.

Problem 3: Avocados

Find on the D2L a datafile named "avocados.csv" -- Retrieved from Kaggle (9/26/2018). This data was downloaded from the Hass Avocado Board website in May of 2018 & compiled into a single CSV.

   a) Define a function that takes a variable name in the form of a string (e.g. "Total Volume"), reads into memory the values for that variable (but just that variable) and computes the mean using the statistics module.

```
mean_SM = readAndComputeMean_SM("Total Volume")
```

   b) Define a function that takes a variable name in the form of a string (e.g. "Total Volume"), reads into memory the values for that variable (but just that variable) and computes the standard deviation using the statistics module.

```
sd_SM = readAndComputeSD_SM("Total Volume")
```

   c) Define a function that takes a variable name in the form of a string (e.g. "Total Volume"), reads into memory the values for that variable (but just that variable) and computes the median using the statistics module.

```
median_SM = readAndComputeMedian_SM("Total Volume")
```

   d) Repeat a-c, but instead of using the statistics module write your own "homegrown" code to compute the mean, standard deviation and median.

```
mean_HG = readAndComputeMean_HG("Total Volume")
sd_HG = readAndComputeSD_HG("Total Volume")
median_HG = readAndComputeMedian_HG("Total Volume")
```

   e) Repeat a-c, but your functions must be memoryless – **you can hold in memory only a single value from the file at any given time**.

```
mean_MML = readAndComputeMean_MML("Total Volume")
sd_MML = readAndComputeSD_MML("Total Volume")
median_MML = readAndComputeMedian_MML("Total Volume")
```

   f) Write test code to demonstrate that the means, standard deviations and medians are the same across all three techniques.

Submission: Submit a single .py file containing all the code to the D2L. Do not zip or archive the file. Your code must include comments at the top including your name, date and the honor statement, "I have not given or received any unauthorized assistance on this assignment." Each function must include a docstring and be commented appropriately.

Grading: There is no partial credit for this problem. The code must compile, run without errors and satisfy all requirement to receive full points.

Problem 4: 7s and 12s

Consider this game. The user begins with an account value of $100. She may place a bet of an amount of her choosing, but not more than her current balance. She then rolls two six-sided dice. Her score is the sum of the faces of the dice. If she rolls a 7 or a 12 she wins and doubles her bet. If she does not roll a 7 or 12, she has the option to double her bet and roll a third die. If all three dice sum to 7 or 12 she wins three times the total value of her bet.

Implement this game. Consider all the ways the user might interact with the system. Use exception handling when appropriate. Your program should be robust to all possible user inputs.

Submission: Submit a single .py file containing all the code to the D2L. Do not zip or archive the file. Your code must include comments at the top including your name, date and the honor statement, "I have not given or received any unauthorized assistance on this assignment." Each function must include a docstring and be commented appropriately.

Grading: There is no partial credit for this problem. The code must compile, run without errors and satisfy all requirement to receive full points.