Homework 3 (due on Sep 14)

hw3-1. Book 3-4. 50pts. Guest List: If you could invite anyone, living or deceased, to dinner, who would you invite? Make a list that includes at least three people you'd like to invite to dinner. Then use your list to print a message to each person, inviting them to dinner.

(Remember to add you name as a comment!)
Turn in your code for this.

hw3-2. Book2. 4-3. 50pts. Counting to Twenty: Use a for loop to print the numbers from 1 to 20, inclusive.

(Remember to add you name as a comment!)
Turn in your code for this.

hw3-3. 50pts. Modify the code from 4-3 to calculate the average of the list. Sum up the list then divide by the length of the list. This is the "mean" for the list. Print this value out.

(Remember to add you name as a comment!)
Turn in your code for this.

hw3-4. Book4. 4-5. 50pts. Summing a Million: Make a list of the numbers from one to one million, and then use min() and max() to make sure your list actually starts at one and ends at one million.

(Remember to add you name as a comment!)
Turn in your code for this.

Homework 4 (due on Sep 21)

hw4-1. 50pts. Use the sample code, `read-csv.py` , and add to it a search for a name in phone-book.csv. This is the example from class lectures. Save a copy of phone-book.csv for later and copy big-book.csv over it (or rename it). Then run the search with a large data set.

(Remember to add you name as a comment!)
Turn in your code for this.

hw4-2. 50pts. Use the image, `image-of-binary-search.png` and type in the code. Get it to work. Get it to work. Do a search for a name (from the original phone-book.csv. Then test with the big data set.

hw4-3. 50pts. Modify the "//" to "/" when mid is calculated. What error do you get?

(Remember to add you name as a comment!)
Turn in your code for this.

hw4-4. 50pts. Add automated tests to the binary search code that test each case of in the function. (We will cover how to do this in lecture 7 - next week)