Homework 8: Functions

L435/L555

Due **Friday**, November 18 (by midnight)

1. From Chapter 6 of *Think Python*, `http://greenteapress.com/thinkpython2/html/thinkpython2007.html`

   > As an exercise, use incremental development to write a function called `hypotenuse` that returns the length of the hypotenuse of a right triangle given the lengths of the two legs as arguments. Record each stage of the development process as you go.

   For information on how to calculate the length of a hypotenuse, see, e.g.: `http://en.wikipedia.org/wiki/Hypotenuse`

2. Take previous code and convert it into a function-based program. The user decides at the beginning of the program which of the two options they want, and then the appropriate function is called:

   (a) Function 0 is for printing and takes whatever values you need to print. Functions 1 & 2 should not print directly, but should call function 0 instead.

   (b) Function 1 takes a sentence and splits it into vowels and consonants and outputs how many of each it found, in addition to reporting the number of other characters. For example, if the sentence is "the big-cat sleeps!", the output should be "It has 5 vowels, 10 consonants, and 4 other characters."

   (c) Function 2 does the same on the word level: it splits the sentence into words and reports the number of vowels, consonants, and other characters per word. (A word here is defined as anything between whitespace.) e.g.,

   ```
   the has 1 vowel(s), 2 consonant(s), and 0 other character(s)
   big-cat has 2 vowel(s), 4 consonant(s), and 1 other character(s)
   sleeps! has 2 vowel(s), 4 consonant(s), and 1 other character(s)
   ```

   Note that it is okay for function 1 to call function 2 or vice versa.

3. (a) Take a list of strings and, using `pop()`, define a function that **recursively** pulls the first letter off each list item and creates a string of "first letters." For example, if the list is `['gamma','alpha','nu','delta','alpha','lambda','frodo']`, the function would return the string `'gandalf'`.

   (b) Now, do the same thing, but accumulating the letters in reverse order. In this same example, you would return `'fladnag'`. Do not use python slicing to reverse the string, but rather accumulate the letters in reverse order as you go through the recursion.

4. (a) Using the `random` module (`https://docs.python.org/3/library/random.html`), write a function which randomly selects a card from a (virtual) deck of cards. Treat the cards as 1–13 (i.e., 13=Ace, 12=King, etc.), and think of the problem as selecting both a number and a suit (spades, hearts, diamonds, clubs).

   As a parameter to this function, take a list of cards which have already been selected. If the drawn card is the same as one of these, you should re-select a card.

   (b) In a separate program, import this function. Write a function within this new program which deals a 5-card hand of poker (no repeats).