LAB #7
1-D Arrays/Lists in Python

1. In this course, all our labs involve paired programming. You do not have to keep the same partner for each lab, but you MUST work with someone in each lab. Two of the ten points for each lab is based on following the paired programming model, as specified in the student handout, and turning in the pair programming evaluation with each lab.

2. At this time, you need to pair with someone in the lab, and finish the rest of the lab as a pair.

3. To begin the lab, you are going to do a few exercises with lists to make sure you understand the concepts of a 1-D array. This will require reading the documentation:
   - Python 3 Documents: http://docs.python.org/3
   - Built-in functions: http://docs.python.org/3/library/functions.html
   - Tutorial w/ Examples: http://docs.python.org/3/tutorial/index.html

Let’s begin by thinking about a 1-D array we already know about, i.e. a string of characters (or strings such as the case in Python!!). I’m sure many of you have thought about wanting to catch the user when he/she enters something invalid, such as characters when are reading an integer or floating point number. There are several ways to prevent this, but one way is to manually take care of by taking the input as a string and checking that it is good. You know that input will provide you a string from the input function. So, let’s use this string as an array.

1. First, ask the user for a positive integer: num = input("Enter a positive integer: ")

2. Print the first and last digit of the number entered, i.e. len() might be useful.

3. Check that the input entered was a positive whole number using isdecimal(). If it is, then turn it into an integer (import locale and use locale.atoi(num)) Else print an error to the user.

4. Now, how could you ensure that num is a positive or negative whole number? What do you know about a negative number? What do you know about whole numbers? Use isdigit() for each character, which just happens to be a string in Python©. How would this change for a floating point number? What would we be looking for in this case? (Recognizing floating point can be design right now!)

5. Now, try to assign num a new first digit, i.e. num[0] = ‘2’. What happens? Why? (Does this help explain why we can’t change their values in a function?)

At this point, you need to get checked off by a TA for 2 points.
Let’s **make a list**, which is mutable. Ask the user for the number of scores he/she will be entering, and then, create an array to hold the scores and initialize all the elements to 0. If you need to, review Friday’s slides.

Now, prompt the user for all the scores.

Whoops, the user forgot about another score!!! **Prompt the user for another score and add it to the list.** Hint: Tutorial, 5.1.

Sort all the scores for the user and print the sorted list to the screen.

Sum all the scores, and print the average to the screen.

**At this point, you need to get checked off by a TA for 2 points.**

Since your Assignment #7 is a larger program, let’s begin by using incremental programming and write the first function of your program. Remember the requirements: You must have a main function, i.e. `def main();`, which creates any variables you need for the program and calls other functions using variables created inside of the main function. The **only call or variable creation you are allowed to have outside of main (or any other function) is the call to main()**, at the very bottom of your program. At this point, let’s implement the `get_initial_input()` based on your design. This function will do the following (and feel free to define and call other functions from inside `get_initial_input()`):

- Ask the user for the number of tests, assignments, quizzes, and labs in their course.
- Ask the user if there is a final with a separate weight from the tests above, e.g. a course has 2 tests, each weighing 12.5%, and 1 final weighing 15%.
- For each category having a number > 0
  - Prompt the user for the weighted percent, out of 100%, which should total 100% for all categories!!!

**At this point, you need to get checked off by a TA for 4 points.**

You will have to turn in your **pair programming evaluations** through the TEACH website. At this point, you need to download and fill out the survey.

Now, upload your survey and presentation file to the **TEACH** website for the last 2 points.

Make sure you sign-up with a TA for demoing/explaining your Assignment #6 next week. This is how all assignments are graded in the course, and if you sign-up and do not make your appointment without rescheduling, then you will be penalized 50 points!!!