Week 4 Practice Problems

#1. Write a function that takes an integer input and returns an integer equal to the number of digits it has. The function should work with all integers representable on your computer.

You cannot use strings, math.log(), or an if-statement. (Hint: use a loop.)

#2. Write a new flipper() function from Q1 of Week 3 by modifying your answer from Q1 above. Write a function that takes an integer as input and returns that integer flipped. Unlike Week 3, the code must work with any integer and a single digit flipped is the number itself. If a number ends in a zero, the flipped number will not have a leading 0. Some examples: 2 - > 2, 34 -> 43, 123450 -> 54321

As above, you cannot use strings, math.log(), or an if-statement.

#3. Write a function that takes in an odd number and returns the sum of odd numbers from 1 to N:

$$1 + 3 + 5 + 7 + 9 + ... + N$$

Write "main" code (i.e., outside of a function) that asks the user for an odd, positive integer. If the user does not enter an odd, positive integer, your code should keep asking until he/she does. Then call your function, have your function return the result, and then print it out.

Your function can be a one-liner (Hint: see the function `sum`).

#4. Write a program that repeatedly asks a user for integers, *until* the user enters a zero and then prints the number of positive and negative integers that were entered. For example:

```
Enter an integer:  2
Enter an integer:  14
Enter an integer:  -3
Enter an integer:  0

You entered 2 positive and 1 negative values.
```

#5. (This one is a bit of a challenge)

Write a function that evaluates $\tanh^{-1}(x)$ from the series

$$\tanh^{-1}(x) = x + x^3/3 + x^5/5 + x^7/7 + x^9/9 + ...$$

where $x < 1.0$ and $x > -1.0$. Notice that each successive term in the series is smaller than the previous one.

The function should take x and epsilon, a small number. Your function should return the estimate of $\tanh^{-1}(x)$ once the change in the estimate from adding a new term would be less than or equal to epsilon. Do not use the math module.

If x is not between -1.0 and 1.0, the program should repeatedly ask for input until it is.

When the input is valid, the program should call your function to estimate $\tanh^{-1}(x)$. Your function returns the approximation and then your code prints the value of $\tanh^{-1}(x)$.