Kerry Ojakian's CSI 31 Class
Due Date: Monday November 20 by 11pm

HW #3

General Instructions:

- Homework must be put in a your dropbox folder; if there are multiple parts, create a single folder for the assignment. Make sure you give clear names to your files and folders.
- Remember that you must work on your own without help from anyone (that includes classmates and tutors).

The Assignment

Each of the 5 programs should be a separate .py file, written with a main() which runs or tests your program.

1. Copy in our Fraction class. Then write a function that takes a list of Fractions as input and returns the sum of all the fractions (this should be a short function!)
2. Write a function that takes a sentence a input (i.e. something like "hi my name is bob" that is a bunch of words separated by spaces). It returns a dictionary which contains a key for each word length that appears in the sentence, and the associated value is the number of times such a word length appears.

   For example in the above sentence, it should reeturn $\{2 : 3, 3 : 1, 4 : 1\}$
3. From Goldwasser and Letscher do Exercise 6.18 (page 235) BUT WITH THE FOLLOWING MODIFICATIONS:
   (a) adding: takes the max of each digit where a missing digit is viewed as zero (ex: '10100' + '101' = '10101')
   (b) subing: takes the min of each digit where a missing digit is left out of result (ex: '10100' + '101' = '100')
4. From Goldwasser and Letscher do Exercise 12.5 (page 433), writing a complete program. Note that this is written in Python 2. So change print so that there are parentheses around the string it prints. And instead of *raw_input* we just use *input*.

5. Consider the following procedure, which starts with some given positive integer. The procedure repeatedly does the following in order to produce a sequence of integers:

   - If the last integer is even, divide it by 2 to get the next integer.
   - If the last integer is odd, multiple it by 3 and then add 1 to get the next integer.

   The above operations are applied till the integer 1 is reached. For example, if you start with 3, then since 3 is odd, your next number is $3 \cdot 3 + 1 = 10$. Then since 10 is even, your next number is $10/2 = 5$. Continuing this process till we reach 1, yields the following sequence: $3, 10, 5, 16, 8, 4, 2, 1$. As another example, if you start with 4, you would produce the sequence: $4, 2, 1$. Write the code for a function `oneSequence`(n) which takes a positive integer as input and outputs this sequence as a list; for example `oneSequence`(4) would produce the output $[4, 2, 1]$.

   Question: For what integers do you think this procedure terminates with 1? Use your program to run some experiments to support your guess.