CS116 – Accumulate a value from a list
Due Date: October 1

Purpose: A common task in computer science is to accumulate the values of a list in some way. The ways values can be accumulated is quite extensive so this is an extremely useful technique. This module will present various ways of accumulating values with various types of lists.

Skills: After completion of this module you should be able to

1. Write a method that returns a value

Knowledge: This module will help you become familiar with the following content knowledge:

1. accumulating a value from a list

Schema:

```
resultSoFar = an initial value
for item in list:
      resultSoFar = resultSoFar combined with processed item
return resultSoFar
```

or

```
resultSoFar = an initial value
for item in list:
     if item has a property:
         resultSoFar = resultoFar combined with processed item
return resultSoFar
```

Activity: With your group perform the following tasks and answer the questions. You will be reporting your answers back to the class in 30 minutes.

1. Look at the following Picture method which counts the number of pixels of a given color:

```
#count the number of pixels of a given color in a picture
 def countPixelsOfColor(self,color):
       count = 0
       for pixel in self.getPixels():
          if pixel.getColor() == color:
             count = count + 1
       return count
```

   Every time we see an item in the list that is the given color we add 1 to the counter. Use this method on a picture. For exampe you can count the number of pixels of your warhol picture that are magenta. You will see that you get a number as the "returned" result.

   Predict what would happen if we move the `count = 0` initialization inside the for loop right before the `if` statement. Try it and explain what happens and why.

   Move the initialization of `count` back where it belongs and then predict what would happen if we indent the `return` statement so that it is in the `for` loop. Try it and explain what happens and why.

2. Here is a Word method which returns a new word which doubles every letter:

```
def double(self):
    resultingWord = ""
    for letter in self.getLetters():
        resultingWord = resultingWord + letter + letter
    return resultingWord
```

   Consider applying this method to a Word("spam") object. What is the initial value of the accumulation variable `resultingWord`?

   How is the accumulation variable `resultingWord` being updated for the first letter of the word 's'?

   How is the accumulation variable `resultingWord` being updated for the second letter of the word 'a'?

3. How can we accumulate the reverse of a word? Write a method `reverse` for the Word class which does this. So this method applied to a word "spam" would return "maps"? As you work on this consider what the accumulation update should look like for each letter.

4. Suppose we want to accumulate the maximum number in a list of numbers. What would be an appropriate initial value for the accumulation variable?

   What would the accumulation update look like?

   An application of this accumulation is to normalize a sound to make it as loud as possible without causing noise:

```
#normalize a sound to maximum possible volume
# subgoal 1: accumulate the maximum sound sample
# subgoal 2: change the volume using 32767 / max as the factor of change.
# (note 32767 is the largest possible sample value
def normalize(self):
      # accumulate the maximum sound sample value
      max = 0
      for s in self.getSamples():
         if s.getValue() > max:
            max = s.getValue()
      # change the volume to maximum possible
      for s in self.getSamples():
          value = s.getValue()
          s.setValue(value * 32767.0 / max)
```

Complete each of the following assignments to be submitted for grading. Each should be done individually but you can consult with a classmate to discuss your strategies or if you get an error message that you do not understand.
For each of these accumulation problems it will help if you ask yourself the following questions:

1. How should I initialize the accumulation variable?
2. How should the accumulation variable be updated each time?

Assignment 1:
Again you have written an essay for school, and it has to be at least five pages long. But your essay is only 4.5 pages long! Now you decide to add n spaces between each of the words. Write a Text method `spaceItOut(self,n)` which accumulates the text into one in which you have added the spaces.
Hint: `" "*n` is n spaces in python.

Criteria of Success: The output should look something like:
`It was a dark and stormy night. It really was.`

Assignment 2:

We will turn a picture into a really boring scene by changing all the pixel colors to the average color. We can get the average by summing up the red, green, and blue components and then dividing by the number of pixels. (Be sure to use integer division operator `\\` since color values must be integers!) You can get the length of the pixel list by using the `len` function on the list.
Your method will need three subgoals:

1. Accumulate the sum of red, green, and blue values for the entire picture
2. Compute the new color value as the averages.
3. Change all the pixel colors to this boring average color

Criteria for Success: The average of the waterlilies is shown below:

Assignment 3:

In a grayscale image the red, green and blue colors are equal (some shade of gray) for each pixel. This value is the luminance. But some of these images may have very low contrast as in this image:

To increase the contrast we can stretch the luminance values so they cover the entire range from 0 to 255 using this formula:

$$newluminance = (oldluminance - minluminance)/(maxluminance - minluminance)) * 255$$

Write a Picture method `increaseContrast` Your method will need three subgoals:

1. Accumulate the minimum luminance for the entire picture
2. Accumulate the maximum luminance for the entire picture
3. Change all the pixel colors to the new luminance value using the formula above. (Hint: The formula may not give you an integer value so you will need to use the `int` function to turn it into an integer)

Criteria for Success: The increased contrast is shown below:

Submit your python files with your methods in Canvas for grading.