This project regards the Fibonacci numbers $F_1, F_2, F_3, \ldots, F_n, \ldots$, defined as follows: we define the first two Fibonacci numbers to be equal to one, and we define any other Fibonacci number to be the sum of the previous two. In math talk:

$$F_1 = F_2 = 1, \qquad F_n = F_{n-1} + F_{n-2} \text{ for } n \geq 3.$$

For our purposes, it's also convenient to define a "zeroth" Fibonacci number by $F_0 = 0$.

For this project you will need the Sage worksheet Fibonacci.sws, which you can download from The Sage Page on our course page, and upload to your account Sage account.

1. Open the worksheet Fibonacci.sws. There are two cells to evaluate: a large first cell, which generates the Fibonacci numbers F_0 through F_{10}, and then plots them; and a smaller second cell below, which writes out these numbers in a list.

   Evaluate both cells to make sure your program is working.

2. Make a minor change to the code in the first cell, so that it produces the Fibonacci numbers F_0 all the way up to F_{30}, instead of just going up to F_{10}, and plots them. Hint: there's only one place in the code where the number "10" appears.

   Once you've done this, execute both cells to make sure everything is working, and to get an idea for how quickly Fibonacci numbers grow.

3. The nth Fibonacci *ratio*, which we'll denote by R_n, is defined as follows: we define R_0, R_1, and R_2 to equal 1, and for $n \geq 3$, we define R_n to be the ratio of the nth Fibonacci number to the previous one. In symbols:

   $$R_0 = R_1 = R_2 = 1, \qquad R_n = F_n/F_{n-1} \text{ for } n \geq 3.$$

   Fibonacci ratios have some interesting properties, which we now wish to investigate, by modifying your program Fibonacci.sws.

   The next few exercises involve doing just that. Note: for the most part, were are going to *add* to the existing code. Actually, we will make a few *minor* modifications to the existing code too. But we won't be *deleting* any entire lines of code.

4. We need a list in which to store our Fibonacci ratios. To accomplish this, type in, just below the command that reads `FibNumbers=[]`, the folllowing:

   ```
   FibRatios=[]
   ```

5. Note that there are existing lines in our code that assign values to the zeroth, first, and second Fibonacci numbers. These are the lines that look like `FibNumbers.append(0)` and so on.

   Now *add* to our code lines that similarly assign values to the zeroth, first, and second Fibonacci ratios. These will look like `FibRatios.append(whatever)`, where "`whatever`"

denotes the appropriate value that you want to assign. You will need to add three lines – one for R_0, one for R_1, one for R_2. Add them in whatever part of the code you think is best.

6. It's easy to finish up now; there are only two three things left to do. The first thing is to compute each successive Fibonacci ratio and store it in the list. We can do both at once: just add, right below the line that says `FibNumbers.append(nextfib)`, this line:

   ```
   FibRatios.append(FibNumbers[i]/FibNumbers[i-1])
   ```

   This says: compute the Fibonacci ratio R_i/R_{i-1}, and store it as the ith entry in the list FibRatios.

   WARNING: code lines within a loop, like the ones here, need to be indented, or Sage will have a fit. Make sure you indent any new lines that appear within the loop – that is, lines that are going to be executed repeatedly.

7. Make *one small change* to the line

   ```
   graph=list_plot(FibNumbers,marker='o',color='blue')
   ```

   which creates the graph of Fibonacci numbers, so that it instead plots the Fibonacci ratios. What do you need to change? I leave it to you.

   Run your program to make sure it does what you want it to.

8. Similarly, modify the line in the second cell that simply says `FibNumbers`, so that, when you execute this line, it outputs the list of Fibonacci *ratios*.

   (You'll note that it outputs these as fractions, not decimals. That's OK. If you want, you can search Sage tutorials online to learn how to convert these numbers to decimal form.)

9. In computer programming, it's considered good form to include *comment lines* – lines that don't *do anything* in the program, but that help *explain* to the reader what the program does. In Sage, comment lines are indicated with a "#."

   Now that you have modified this program, the existing comment lines no longer necessarily apply. Please modify and/or add to the comment lines, so that they accurately reflect what the program does now.

10. Finally, rename your worksheet to `Fibonacci_lastname`, and share it with me. You're done!!