LARSON—MATH 353–CLASSROOM WORKSHEET 12
Recursion—Density of Primes.

1. Log in to CoCalc.

   (a) Start the Chrome browser.
   (b) Go to `https://cocalc.com`
   (c) Login (**your VCU email address** is probably your username).
   (d) You should see an existing Project for our class. Click on that.
   (e) Click "New", then "Worksheets", then call it **c12**.

   ### Recursion

   A **recursive** function is a function that calls itself. It must always have a *base case* so that the recursion eventually stops.

2. Here is an example of a recursive definition of the *factorial* function. The base case here is the case where the input is 0 or 1.

```
def factorial(n):
    if n==0 or n==1:
        return 1
    else:
        return n*factorial(n-1)
```

3. The Fibonacci numbers are $0, 1, 1, 2, 3, 5, 8, 13, \ldots$ where the i^{th} Fibonacci number $F[i] = F[i-1] + F[i-2]$ (for $i \geq 2$ and $F[0] = 0, F[1] = 1$). Write a recursive function to compute $F[i]$ for $i \geq 0$.

   Recursive functions can be very natural to code. **Why** can recursive functions be extremely inefficient (slow)?

   ### Scatter Plots

   Given a list L of pairs (x, y) you can plot the *scatter plot* that consists just of those points with `scatter_plot(L)`.

4. Try: `scatter_plot([(0,1),(2,4),(3.2,6)])`

   To get two scatter plots on the same graph use "+".

5. Try: `scatter_plot([(x,x) for x in [5..20]])+scatter_plot([(x,x**2) for x in [5..20]], facecolor="red")`

Primes & Gauss

6. Write a function density_of_primes(n) that **returns the number of primes** in the integers up to n, divided by n. Test it.

7. Use `scatter_plot` to *visualize* the density of the primes as values of n get larger and larger.

   What we see is that the density appears to go down as n increases. Gauss conjectured a function or formula that approximated the density of primes.

8. $f(x) = \frac{1}{x}$ is a decreasing function. Is it a good approximation? Plot it on the same graph as your graph of the density of primes. Try:
   `scatter_plot([(x,density_of_primes(x)) for x in [1..100]])` +
   `scatter_plot([(x,1/x) for x in [1..100]], facecolor="red")`

9. **Experiment** with other decreasing functions $f(x)$ to try to find a "better" approximation of the density of primes. Choose and define an $f(x)$ and then try: Try:
   `scatter_plot([(x,density_of_primes(x)) for x in [1..100]])` +
   `scatter_plot([(x,f(x)) for x in [1..100]], facecolor="red")`.

10. **Getting your classwork recorded**

    When you are done, before you leave class...

    (a) Click the "Make pdf" (Adobe symbol) icon and make a pdf of this worksheet. (If Cocalc hangs, click the printer icon, then "Open", then print or make a pdf using your browser).

    (b) Send me an email with an informative header like "Math 353—c12 worksheet attached" (so that it will be properly recorded).

    (c) Remember to attach today's classroom worksheet!