LARSON—MATH 255–CLASSROOM WORKSHEET 16
Timing—Experiments.

1. (a) Start the Chrome browser.
   (b) Go to `http://cocalc.com`
   (c) Login using **your VCU email address** .
   (d) Click on our class Project.
   (e) Click "New", then "Worksheets", then call it **c16**.
   (f) For each problem number, label it in the Sage cell where the work is. So for Problem 2, the first line of the cell should be `#Problem 2`.

Recursion

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

2. The Fibonacci sequence F_n is defined as follows $F_0 = 0$, $F_1 = 1$ and $F_n = F_{n-1} + F_{n-2}$ for $n > 1$. Here is a recursive function `fib(n)` that computes the nth Fibonacci number.

```
def fib(n):
    if n==0 or n==1:
        return n
    else:
        return fib(n-1)+fib(n-2)
```

3. Define a non-recursive (iterative) function `fib2(n)` that computes the nth Fibonacci number.

Timing

For large programs or calculations that are at the edge of what's possible. It is crucial to optimize and test the speed of your code. One simple first step is simply to *time* your program using Sage's built-in `timeit()` function.

It is often intuitive to define a function recursively, but usually the same function can be defined without recursion.

4. Evaluate and write down what you get for `timeit("fib(10)")`, `timeit("fib(20)")` , and `timeit("fib(25)")` .

5. Now evaluate and write down what you get for `timeit("fib2(10)")`, `timeit("fib2(20)")`, and `timeit("fib2(25)")` .

6. The recursive `fib(n)` function we defined takes a very long time to respond for $n = 30$ and may never respond for $n = 40$. Now try `fib2(40)` and `fib2(400)`. Why does the iterative function work while the recursive function does not?

7. Solve the equation $\frac{a+b}{a} = \frac{a}{b}$, for a and b. Find $\frac{a}{b}$. Get a 10-digit approximation for this quantity (this is the Golden Ratio).

8. Define a function `fib_ratio(n)` which returns the ratio of the $(n+1)^{th}$ Fibonacci number to the n^{th}. find `fib_ratio(10)` and `fib_ratio(100)`. Compare this answer to your previous answer. What can you conjecture?

- **Coin Flip Questions** When you flip a coin 100 times would you expects to see 6 heads or tails in a row at some point? We can investigate this question too by simulating coin flips and repeating our *experiment* a number of times.
- If you flip a coin 100 times, you would expect about 50 heads. Its possible that you could get 100 heads. But this would be rare. How rare? We can *simulate* flipping a coin a hundred times, write down how many heads we got, and then repeating this experiment. This will give us a *distribution* of various possible outcomes.

9. Use `random()` to define a function `coin_flip()` which randomly *returns* the string "H" (for heads) half the time and returns the string "T" (for tails) half the time. Check that it works.

10. Use your `coin_flip()` to define a function `coin_flips(n)` which *returns* a list of n random H's or T's (representing the result of n coin flips).

```
def coin_flips(n):
  flip_results = []
  for i in [1..n]:
      flip_results.append(coin_flip())
  return flip_results
```

    Check that it works.

11. Define a function `number_of_heads(n)` that counts and *returns* the number of heads you get after flipping a coin n times.

12. Write a function `flip_data(n)` which *prints* the numbers of both heads and tails you get after flipping a coin n times.

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 255—c16 worksheet attached" (so that it will be properly recorded).

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