Last name ______________________

First name ______________________

LARSON—MATH 305—CLASSROOM WORKSHEET 20
Sage LAB!—Riemann Hypothesis, Wilson's Theorem, Extended GCD

Organizational Notes

1. A Zoom recording link and class notes will be sent out after each Zoom class.
2. Remember to send your answers to the classroom worksheets, and pdfs of your Lab work. Title your email with enough to help me record your "participation".
3. Homework #4 is due next Thursday, Oct. 8. Its the Test Review—so this can't be late
4. Test 1 is Friday, Oct. 9.

Logging into Sage/CoCalc

1. Start the Chrome browser.
2. Go to `http://cocalc.com` and sign in.
3. You should see an existing Project for our class. Click on that.
4. Click "New", call it **c20**, then click "Sage Worksheet".
5. For each problem number, label it in the Sage cell where the work is. So for Problem 1, the first line of the cell should be `#Problem 1`.
6. When you are finished with the worksheet, click "make pdf", email me the pdf (at `clarson@vcu.edu`, with a header that says **Math 305 c20 worksheet attached**).

Riemann Hypothesis

In the Lagarias paper I sent he proves that the Riemann Hypothesis is equivalent to:

$$\sum_{d|n} d \leq H(n) + e^{H(n)} * log(H(n),$$

for any positive integer n, where $H(n) = \sum_{j=1}^{n} \frac{1}{j}$ is the n^{th} *harmonic number.*

1. To run some numerical tests we need little more than to code a `harmonic_number` function. Write a function—and make sure it works for small n. We can start with the `H`, `LHS`, and `RHS` functions we wrote last week.

2. What kind of tests can we run for the Riemann Hypothesis?

3. How else can we investigate this?

From our Book

Wilson's Theorem

Wilson's Theorem says: p is prime if and only if $(p-1)! \equiv -1 \pmod{p}$.

4. Use this to code a test for the primality of an input integer p. (Our book's author calls this the least efficient primality test imaginable).

5. Now test all integers from $p = 2$ to $p = 20$. How gar can we go? $(p-1)!$ grows *fast*.

Extended GCD

Given integers a, b with $\gcd(a, b) = g$, there are integers x and y such that $g = ax+by$. The *extended GCD algorithm* follows the steps of the GCD algorithm while maintaining enough extra information to construct x and y.

6. How can we find x and y? Write an `extended_gcd` function that returns g, x, y. Start by considering the case where $\gcd(a, b) = 1$ and noting that we can just test small values of x and y. How small?

7. Now test `extended_gcd` to see if it works!

8. Can we use the theory we've developed so far to write a better test?

Extras

9. If p and $p+2$ are both prime, they are called *twin primes*. Find the first several pairs of twin primes.