Last name ________________________

First name ________________________

LARSON—MATH 353–CLASSROOM WORKSHEET 28
$x = n^2 + 1$ Primes Investigation.

Set up.

1. Start the Chrome browser.
2. Go to `https://cocalc.com`
3. Log in to your account.
4. You should see an existing Project for our class. Click on that.
5. Make sure you are in your Home directory (if you put files in the Handouts directory they could be overwritten.)
6. Click “New”, then “Jupyter Notebook”, then call it **353-c28**.
7. Make sure you have SAGE as the *kernel*.
8. Look in your `Home` directory. You should see a `conjecturing.py` file and an `expressions` file **AND** today’s Jupyter notebook.
9. Copy the latest version of `number_theory.sage` from the Handouts directory to your Home directory.
10. `load("number_theory.sage")`.

The **research question** is: are the infinitely many primes of the form $x = n^2 + 1$?

Methodology

1. Here’s the big picture for today’s experiments:
   (a) Upper-bound conjectures for `count_prime_divisors_base`, using **prime** $x = n^2 + 1$ integers.
   (b) Lower-bound conjectures for `count_prime_divisors_base`, using **prime** $x = n^2 + 1$ integers.
   (c) If any conjectures hold up, we can use these as booleans/properties as input to the properties conjectures.

2. Start with the following initial run for a **upper-bound** for `count_prime_divisors_base`, using **prime** $x = n^2 + 1$ integers as the data/objects/input, and where we will interpret produced conjectures as being true for these integers.

   What conjectures do you get?

```
objects = [5, 17, 65, 901, 325, 101, 4625, 197, 1025, 4357, 485, 6085]

invariants = [digits10, digits2, count_divisors,
count_prime_divisors, number, euler_phi, sigma, base,
    count_divisors_base,
count_prime_divisors_base, euler_phi_base, sigma_base, one,
    largest_prime_factor, largest_prime_factor_base]

theorems = []
inv_of_interest = invariants.index(count_prime_divisors_base)

conjs = conjecture(objects, invariants, inv_of_interest, upperBound =
    True, theory = theorems, debug = True)

for conj in conjs:
    print(conj)
```

3. For each produced conjecture, test whether it is true for all the prime $x = n^2 + 1$ integers in the $Sp1$ list. If you find a counterexample, report the smallest integer which is a counterexample.

4. If you found any counterexamples, add these to your `objects` list, and then re-run the conjecturing program (do that in a new cell so you have a full history of your investigations.

5. When you have a run of conjectures, all of which are true for all the prime $Sp1$ integers, then choose a conjecture that interests you, write the conjecture and all relevant definitions in a new cell.

6. Can you prove it? If so, add it as a theorem and generate new conjectures.

7. We can push our investigation forward by any of the following:

   (a) Finding a counterexample to a conjecture and adding it to the examples/objects list.

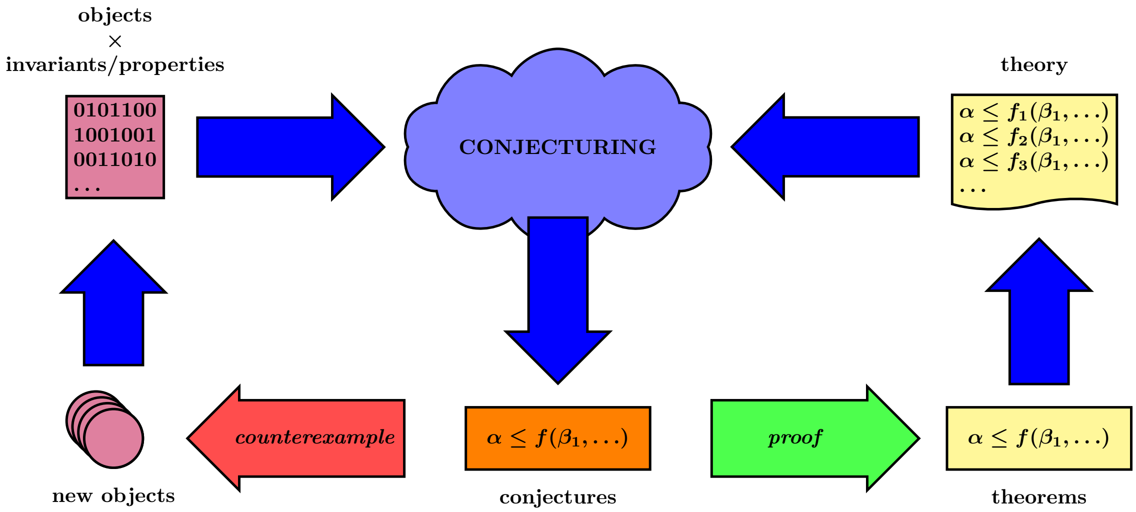
   (b) Proving a conjecture and adding it to the theorems list.

   (c) Coding/adding new invariants.

8. When these investigations no longer seem promising, we can try lower-bound conjectures.

9. If we ever have a vetted (tested) open conjecture (which defines a property), it might be useful to define a procedure corresponding to that conjecture and try that as a property to add to our list of properties.

10. Let's **experiment**!



Getting your classwork recorded

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

1. Click the "Print" menu choice (under "File") and make a pdf of this worksheet (html is OK too).

2. Send me an email (`clarson@vcu.edu`) with an informative header like "Math 353 - c28 worksheet attached" (so that it will be properly recorded).

3. Remember to attach today's classroom worksheet!