Last name ________________________

First name ________________________

LARSON—MATH 353–CLASSROOM WORKSHEET 18
$x = n^2 + 1$ 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-c20**.
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.

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

1. **Open Conjectures**.

   ```
   count_prime_divisors(x) <= digits10(x)
   ```

   Are any of these resolved?

2. Here's an idea for new invariants. For any $x = n^2 + 1$ integer, we can also grab onto the n (the "base") and define any invariants for *that* integer. These are also invariants for n. There are no wrong invariants—and also you can't tell ahead of time what invariants will show up in useful conjectures.

   Some of these are now defined and added to `number_theory.sage`. Let's test these.

   Methodology

   (a) We need `conjecturing.py` loaded for our investigation. I added the command to the `number_theory.sage` file. Every time that file is loaded, the `Conjecturing` program will get loaded too.

   (b) We need *invariants*.

To use the `Conjecturing` program, we'll need some *invariants*. The ones we've coded in class are in the *number_theory.sage* file in your Handouts folder. We don't want to keep re-coding those. We can use this file as a permanent record of everything we've coded for this research. Copy or move this file to your Home directory. We'll do this every class as I update them. If you have/use your own invariants, please name your file something else.

(c) Start with the following initial run for a **lower-bound** for `count_prime_divisors`, using $x = n^2 + 1$ integers as the data/objects/input, and where we will interpret produced conjectures as being true for these integers. (Note that the theorems we proved in class are not recorded here—those are proved upper bounds).

What conjectures do you get?

```
objects = [5,17,65,901,325,170,2210]

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]

theorems = []

inv_of_interest = invariants.index(count_prime_divisors)

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

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

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

(e) 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.

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

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

(h) Read the Wikipedia number theory page, or the Sagemath number theory page, or find a research article or book with a number theory invariant that we are not currently using. If its not built-in you'll have to code it, or ask ChatGPT to code it, or see me and get help. Explain your invariant and give some examples. (And give a reference for your source).

(i) What is your code for this invariant? Test it to make sure that it works.

(j) Add it to your list of invariants (again, make a new cell, so you have your entire complete lab record) and rerun the conjecturing program. Do you get anything new (it will only be new if one of the conjectures includes your invariant. If not, they will be the same conjectures as before)?

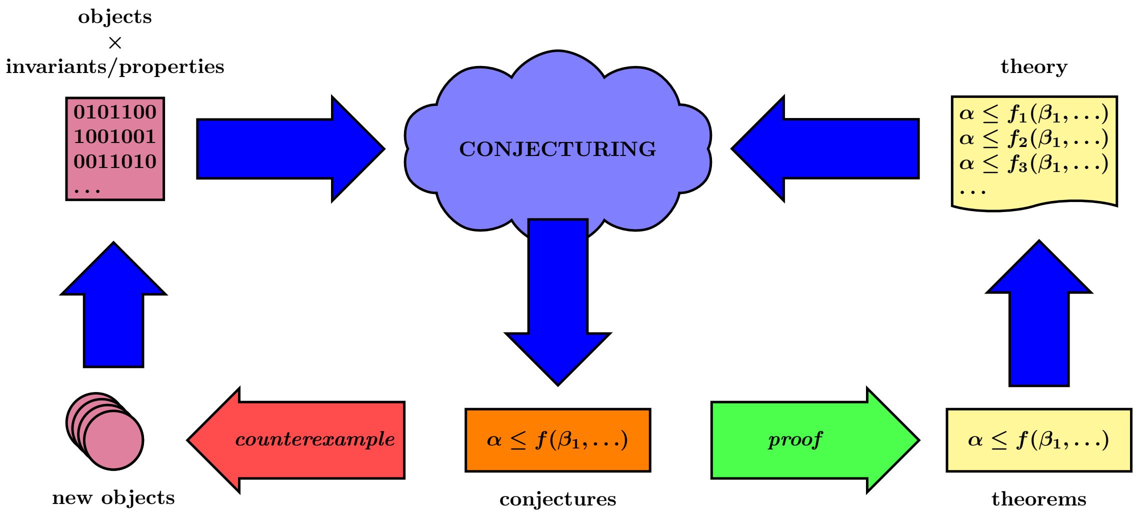
(k) We can push our investigation forward by any of the following:

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

ii. Proving a conjecture and adding it to the theorems list.

iii. Coding/adding new invariants.

3. Now 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 - c20 worksheet attached" (so that it will be properly recorded).

3. Remember to attach today's classroom worksheet!