Last name _____

First name _____

LARSON—MATH 353–CLASSROOM WORKSHEET 17
$n^2 + 1$ Primes Investigation.

1. Sign in to your CoCalc account.
   (a) Start the Chrome browser.
   (b) Go to `https://cocalc.com`
   (c) Log in to your account.
   (d) You should see an existing Project for our class. Click on that.
   (e) Make sure you are in your Home directory (if you put files in the Handouts directory they could be overwritten.)
   (f) Click "New", then "Jupyter Notebook", then call it **353-c17**.
   (g) Make sure you have SAGE as the *kernel*.
   (h) Look in your `Home` directory. You should see a `conjecturing.py` file and an `expressions` file **AND** today's Jupyter notebook.

Review

2. 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.

3. **Open Conjectures**.

```
count_prime_divisors(x) <= digits10(x)
count_prime_divisors(x) <= 1/2*count_divisors(x)
count_prime_divisors(x) <= count_divisors(x) - 1

count_prime_divisors(x) >= digits10(x) - 1
count_prime_divisors(x) >= (1/digits10(x))
```

   Are any of these resolved?

Experiments

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

4. 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.

5. We started with a few prime and non-prime integers of the form $n^2 + 1$.

```
#lets try to make conjectures using a *few* of the numbers from N
#lets get a couple primes and non-primes
objects = [5,17,65,901,325]

#need invariants for integers
invariants = [digits10, digits2, count_divisors, count_prime_divisors,
    number, euler_phi]

#maybe count_prime_divisors is worth investigating
#Note: this invariant is 1 for primes and bigger than 1 for non-primes

invariant_of_interest = invariants.index(count_prime_divisors)

conjs = conjecture(objects, invariants, invariant_of_interest,upperBound
     = True, debug=True)
for conj in conjs:
    print(conj)
```

6. How should we proceed? Here are 2 new ideas: (1) we can add theorems using the (optional) `theory` parameter for the conjecture procedure; and (2) we can add new invariants.

7. If it is true, for instance, that for any $n = x^2 + 1$ integer that `count_prime_divisors(x) <= 1/2*count_divisors(x)` is true, we can define that upper bound using a procedure-name, and then adding that theorem to a list of theorems that we can input to the program:

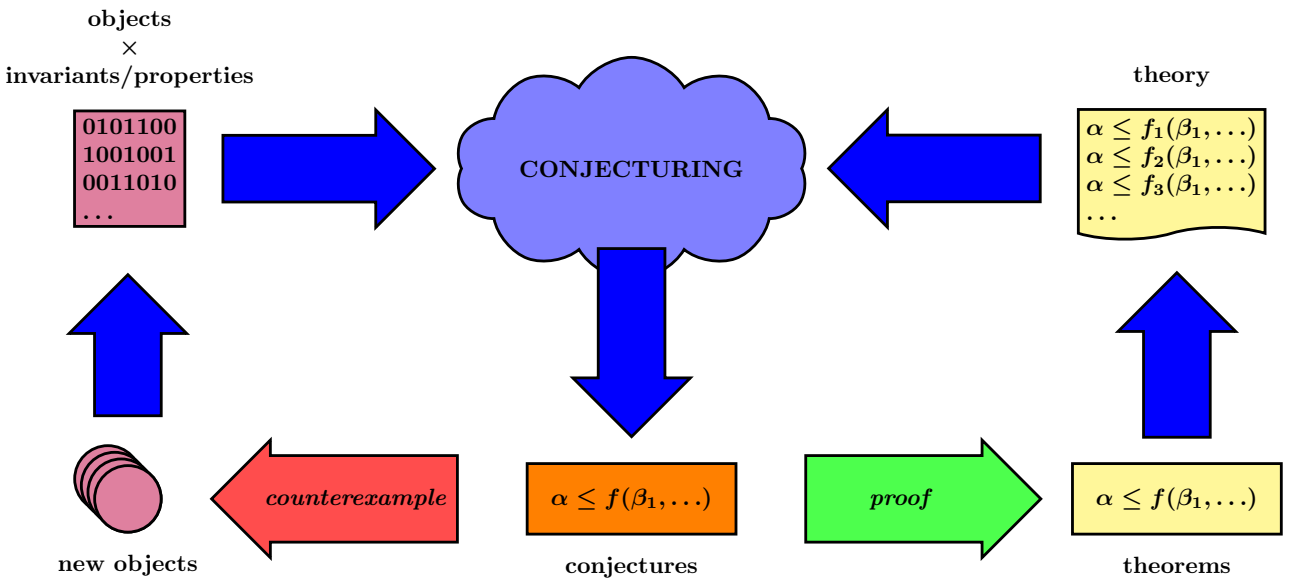
```
def half_count_divisors(x):
    return (1/2)*count_divisors(x)

theorems = [half_count_divisors]

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

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

   Let's define these, add them to our invariants list, and then re-run `conjecture`.



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 - c17 worksheet attached" (so that it will be properly recorded).
3. Remember to attach today's classroom worksheet!