LARSON—MATH 353–CLASSROOM WORKSHEET 37
Conjecturing Invariant Bounds

1. Log in to CoCalc.

   (a) Start the Chrome browser.

   (b) Go to `https://cocalc.com`

   (c) Login (**your VCU email address** is probably your username).

   (d) You should see an existing Project for our class. Click on that.

   (e) Click "New", then "Worksheets", then call it **c37**.

   We can use the CONJECTURING program to conjecture upper and lower bounds for an *invariant* of an mathematical object (number, matrix, graph, etc). An *invariant* in this context means any number associated with that object. So, for instance, the determinant of a matrix is a matrix-invariant.

   Inequalities show up everywhere in mathematics; famous ones include the Cauchy-Schwartz inequality. Investigating bounds can be of enormous practical importance: bounds are useful when we want to reduce a *search space* where the answer to some question may be (for instance optimizing a discrete function).

   We **noted** that the posotive integers can be *partitioned* into three classes: the perfect numbers, the deficient numbers and the abundant numbers. There aren't many perfect numbers so it might be a useful idea to better understand the deficient and abundant numbers.

   We **started** by generating conjectures for the deficient numbers. All of the integers in:

   ```
   objects = [5,7,8,9,10,11,13,14,15,16,17,19,4, 344, 1]
   ```

   are deficient. Thus any generated conjectures might be *interpreted* as being about deficient numbers.

   What invariant should we investigate? Our **idea** was to generate lower bound conjectures for the divisor sum—the defining invariant for being deficient.

2. Load "perfect_numbers.sage" (which will load "conjecturing.py") so that we have access to the integer invariants defined there.

3. Rerun our last example from last class(which is also an example that you can imitate.) Remember that a you can perform a *lower bound* investigation by flipping the `upperBound` switch to `False`. Any time you run conjectures you should ask: Are these conjectures true? If not, can you find a counterexample?

```
objects = [5,7,8,9,10,11,13,14,15,16,17,19,4, 344, 1]

invariants = [divisor_sum, number_of_divisors, number_of_prime_divisors,
number, value_pi, harmonic_mean, number_of_distinct_prime_factors,
number_of_prime_divisors, number_of_ones_in_binary, digits10, digits2,
value_e, value_golden_ratio]

invariant_of_interest = invariants.index(divisor_sum)

conjs = conjecture(objects,invariants,invariant_of_interest,upperBound=False)

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

4. This first conjecture is true. Why?

5. Is the last conjecture true? How can we investigate?

6. Now imitate our previous CONJECTURING experiment to see if you can get any interesting conjectures that might be relevant to investigating perfect numbers.

   Add any new invariants you can think of. The more invariants we have the more ingredients we have for our conjectures!

7. **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 353—c37 worksheet attached" (so that it will be properly recorded).

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