LARSON—MATH 353–CLASSROOM WORKSHEET 17
Perfect Numbers—Conjecturing.

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 **c17**.

Perfect Numbers & Files

A number (integer) n is *perfect* if the sum of its proper divisors (the divisors less than n) equals n (or equivalently that the sum of all divisors equals $2n$). 6 is the smallest perfect number.

2. Now it is the case on any larger program that you will want to use functions you have previously defined. These are called *tools*. Instead of copying and pasting from your old code. You can save them as *files* and load them as needed.

   (a) Click "New". Type `perfect_numbers.sage` and then click "file".

   (b) Define the function:

```
def is_perfect(n):
    if sum(n.divisors())==2*n:
        return True
    else:
        return False
```

   (c) Click "Save" and then go back to your **c17** worksheet.

   (d) Type `load("perfect_numbers.sage")` and evaluate.

   (e) Now try `is_perfect(n)` for a few values of n. You never need to write this function again. You have a tool!

The Conjecturing program

3. Move the `conjecturing-0.13.CoCalc.zip` file from your Handouts CoCalc directory to your Home/Root directory.

4. Now we will install the program. Click "New" and then "Terminal" to get a Sage terminal window (call it anything you want. I often call mine "terminal").

5.
   - unzip conjecturing-0.13.CoCalc.zip
   - cd conjecturing-0.13.CoCalc
   - make

6. Switch back to your $c17$ Sage worksheet and test your installation. First `load` the CONJECTURING program by running the command: `load("conjecturing.py")`.

7. Now let's see if we can get a conjecture with the "properties" version of the program. Type the following lines into a $c17$ worksheet cell and then run it:

```
Integers = [6,28]

def is_perfect(n):
    return sum(n.divisors())==2*n

Properties = [is_perfect, Integer.is_prime, Integer.is_square,
Integer.is_squarefree]

Prop_of_interest = Properties.index(is_perfect)

propertyBasedConjecture(Integers,Properties,Prop_of_interest,sufficient=True)
```

   If a conjecture is true, the only way to be certain is to *prove* it. If it is false, the only way to be certain of that is to find an example that demonstrates falsity (a *counterexample*).

8. What conjectures do you get? (Are they true? If not find a counterexample and add it to Sage. Then re-run to get new conjectures.)

9. To be maximally useful, we need *lots* of integer properties (these will go in the Properties list). What else can we add?

10. Two related properties would be having a divisor sum which is more than $2n$ and having a divisor sum which is less than $2n$. Call these `has_divisor_surplus(n)` and `has_divisor_deficiency(n)`. Test them.

11. Find all the integers up to 100 with divisor sum less than $2n$.

12. If both functions work, add them to your file `perfect_numbers.sage`.

13. Now add the names of your two new properties to the Properties list and re-run CONJECTURING.

14. What other integer properties do we already know—or that we can cook up—to add to our Properties list (and, after suitable testing, to `perfect_numbers.sage`)?

15. **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—c17 worksheet attached" (so that it will be properly recorded).

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