LARSON—MATH 353–HOMEWORK WORKSHEET h06
Perfect Numbers & the `conjecturing` Program.

1. Log in to your Sage Cloud account.

   (a) Start the Chrome browser.

   (b) Go to `http://cloud.sagemath.com` and sign in.

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

   (d) Click "New", call it **h06**, then click "Sage Worksheet".

   Perfect Numbers Investigation

   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.

   As we go we are storing all tested definitions and properties in `perfect_numbers.sage`. Several others have been added there, as well as the command `load("conjecturing.py")`—so that program will be loaded every time `perfect_numbers.sage` is loaded.

2. Start-up. `load("perfect_numbers.sage")` to load what we have so far (my current copy is in the Handouts folder. Your copy may have different functions. It should be in your Root/Home directory.

3. Create a new integer property. If your name is "Benilda", call it `benilda(n)`. It should input an positive integer n and return True if n has some specified property, and False otherwise. Find a property on Wikipedia—or just make something up (for instance, the property might be "has at least two 2's in the base-10 digits"). Test your property to make sure it does what you expect it to do. Then add it to your `load("perfect_numbers.sage")`.

   The Conjecturing program

   By the design of the CONJECTURING program, all produced conjectures are guaranteed to be true for all *input* integers. We interpret the conjectures as being true for *all* integers. This claim may be true or may be false. If it is true, we must *prove* it; and if it is false, we must find an example that shows or demonstrates that the conjecture is false. Such an example is called a **counterexample**. Every time the CONJECTURING program produces a conjecture that is false and we find a counterexample we will add that counterexample.

4. **The 8th Run of our Classroom Investigation**. In class the CONJECTURING program produced the following conjecture:

   ```
   ((~(has_divisor_deficit))^(has_divisor_surplus))->(is_perfect)
   ```

   (Recall, the caret symbol represents *exclusive or* or XOR). Explain how to *interpret* this conjecture. What does it *mean*? You should explain enough so that it's clear how to check if the conjecture is true or false.

5. I claimed in class that the last conjecture is true. Explain why the conjecture is indeed true. You'll have to dig into the definitions.

6. **Necessary Conditions**. We can also investigate *necessary* conditions for a number to be perfect. Try:

```
#Necessary Condition Conjectures. 1st Run.

Integers = [6, 28, 496, 8128, 8, 15, 7, 25, 12, 140]

Properties = [is_perfect, Integer.is_prime, Integer.is_square,
Integer.is_squarefree, has_divisor_surplus, has_integral_harmonic_mean,
is_odd, is_even, benilda]

Prop_of_interest = Properties.index(is_perfect)

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

What conjectures did you get?

7. For each conjecture produced on your last run, try to find a counterexample if you think it is false, or any explanation of why it is true if you think it might be true. If you don't find a counterexample or convincing explanation, at least write sufficiently about your analysis to support what you think *might* be the case about truth or falsity of the conjectures.

Getting your homework recorded

When you are done...

(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—h06 worksheet attached" (so that it will be properly recorded).

(c) Remember to attach your homework worksheet!