

# Perfect Numbers!

CSCI 141, Lab # 3

Fall 2013

**Due:** Your program must be submitted to Canvas before midnight, Friday, October 18.

**Problem:** Perfect numbers are defined as positive integers that are equal to the sum of their proper positive divisors. For example, the first two perfect numbers are

$$6 = 1 + 2 + 3$$

$$28 = 1 + 2 + 4 + 7 + 14$$

The next two are 496 and 8128 (perfect numbers are rather rare).

Perfect numbers have fascinating relationships to prime numbers and the history of mathematics. You can read about them here: [http://en.wikipedia.org/wiki/Perfect\\_number](http://en.wikipedia.org/wiki/Perfect_number)

**Program:** Write up a program to check if a number is perfect by first generating all the proper positive divisors of the number, and then adding up all the numbers in that list, and checking for equality.

Here, for example, is my `divisors` program in action:

```
> (map (lambda (n) (list n (divisors n)))
      '(0 1 2 3 4 5 6 7 8 9 10))
'((0 ())
  (1 ())
  (2 (1))
  (3 (1))
  (4 (2 1))
  (5 (1))
  (6 (3 2 1))
  (7 (1))
  (8 (4 2 1))
  (9 (3 1))
  (10 (5 2 1)))
```

And here is the `perfect?` predicate that uses it:

```
> (filter perfect? (ints 1000))
'(6 28 496)
```

Use good Scheme style!