CSCI 301, Labs # 6 and # 7

Winter, 2017

Due dates:

- Lab #6 must be submitted to Canvas before midnight Tuesday, February 28.
- Lab #7 must be submitted to Canvas before midnight Tuesday, March 7.

Overall Problem: These labs are taken from an exercise found in another online scheme book: `http://www.eecs.berkeley.edu/~bh/ss-toc2.html`. Both of them together will create a procedure `number-name` that takes a positive integer argument and returns a sentence containing that number spelled out in words. You do not have to worry about negative numbers, nor about the "and" that some people throw in before the last bunch.

```
> (number-name 5513345)
(five million five hundred thirteen thousand three hundred forty five)
> (number-name (factorial 20))
(two quintillion four hundred thirty two quadrillion nine hundred two
 trillion eight billion one hundred seventy six million six hundred
 forty thousand)
> (number-name 0)
'(zero)
```

Note that the responses are lists of symbols, not strings. You may want to examine the difference in the documentation: `file:///usr/share/doc/racket/reference/symbols.html`

Once again, **no string processing!**

Lab #6: For the first part of this problem, write an intermediate version of the solution, that behaves like this:

```
> (number-name 1234567890)
'(1 billion 234 million 567 thousand 890)
> (number-name 1000000)
'(1 million)
> (number-name 1000000001)
'(1 billion 1)
> (number-name 2423000000000000002344)
'(2 sextillion 423 quintillion 2 thousand 344)
> (number-name 0)
'()
```

You will have to write a procedure to break up a big number into its three-digit pieces (use `quotient` and `remainder`). Note that if any slot is a zero, both it and its name are omitted.

Lab #7: This lab has two parts. First, write a procedure `name<1000` that converts a number less than 1000 into words, as shown at right. Note that `zero` only shows up if the whole number is zero.

Second, put this together with the solution to lab #4, and you should solve the whole problem, with behavior as in the first box, above.

As a final wrinkle, note that `zero` only shows up if the entire number is zero. We never say, for example, "one million zero thousands two," we just say "one million two."

```
> (name<1000 100)
'(one hundred)
> (name<1000 101)
'(one hundred one)
> (name<1000 15)
'(fifteen)
> (name<1000 111)
'(one hundred eleven)
> (name<1000 222)
'(two hundred twenty two)
```