CSCI 141, Lab # 2

Fall 2013

Due: Your program must be submitted to Canvas before midnight, Friday, October 11.

Problem: The binomial coefficients are usually defined as

$$\left(\begin{array}{c} n \\ k \end{array} \right) = \frac{n!}{k!(n-k)!}$$

These are also the elements in the nth row and kth column of Pascal's triangle.

It is straightforward to code up this expression in Scheme, assuming an implementation of `factorial` has already been written:

```
(define (binom-slow n k)
  (/ (factorial n)
     (* (factorial k)
        (factorial (- n k)))))
```

However, this program is needlessly slow, since a large number of coefficients cancel. In fact, only the smaller of the two terms in the denominator need be calculated.

Program: Write up a version of `binom` that only computes the smaller term in the denominator. You will have to write an auxiliary function that is a modified version of the `factorial` procedure to calculate the new numerator. This function will take two parameters instead of one, and can be used for both the numerator and the denominator in your new version.

Use good Scheme style!

- No assignment statements!
- No global variables!
- Your auxiliary function should be local to the `binom` procedure.
- Scope of variables should be appropriate.

Also, write up a loop to give a single row of Pascal's triangle. It should take both the row number and the function used to compute binomial coefficients as parameters. Here are some examples:

```
> (pascal-row 5 binom)
1 5 10 10 5 1
> (pascal-row 5 binom-slow)
1 5 10 10 5 1
> (pascal-row 10 binom)
1 10 45 120 210 252 210 120 45 10 1
```

Finally, write a procedure to give Pascal's triangle:

```
> (pascal-triangle 6 binom)
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
```