Objectives

- Work with recursive functions
- Work with conditionals `if` and `cond`

Activities

1. Young Jeanie knows she has two parents, four grandparents, eight great grandparents, and so on.
   (a) Write a recursive function to compute the number of Jeanie's ancestors in the n^{th} previous generation. The number of ancestors in each generation back produces a sequence that may look familiar:
   $$2, 4, 8, 16, \ldots$$
   For each generation back, there are twice the number of ancestors than in the previous generation back. That is, $a_n = 2a_{n-1}$. Of course, Jeanie knows she has two ancestors, her parents, one generation back.
   (b) Write a recursive function to compute Jeanie's total number of ancestors if we go back n generations. Specifically, `(num-ancestors n)` should return:
   $$2 + 4 + 8 + \cdots + a_n$$
   Use your function in part (a) as a "helper" function in the definition of `(num-ancestors n)`[1].

2. Perhaps you remember learning at some point that $\frac{22}{7}$ is an approximation for π, which is an irrational number. In fact, in number theory, there is a field of study named Diophantine approximation, which deals with rational approximation of irrational numbers.
   (a) In 1910, Srinivasa Ramanujan, an Indian mathematician discovered several infinite series that rapidly converge to π. The series Ramanujan discovered form the basis for the fastest modern algorithms used to calculate π. One such series is
   $$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$
   This series computes an additional eight decimal places of π for each term in the series. Write a Scheme function to calculate π using this series. Your function, `(pi-approx k)`, should take k as a parameter and produce the approximation of π produced by the first k terms in the series. You may use the following skeleton to complete your solution. All you will need to add is the helper function to compute the summation defined above:

[1] Of course, we can use the closed-form solution for the geometric progression to compute `num-ancestors` ($ancestors(n) = 2^{n+1} - 2$) but that doesn't give us any experience with recursive functions. However, this is a useful fact we can use when testing our functions to ensure they are correct.

```scheme
(define (pi-approx k)
  ;Recall the definition of factorial from the lecture slides
  (define (factorial k)
    (if (= k 0)
        1
        (* k (factorial (- k 1)))))

  ;Define a helper function to compute the summation of the terms in the series
  (define (pi-aux k)
    ;Take care with the base case, k=0 is a term in the series

  )

  ;Body of the pi-approx function
  (/ 1 (*  (/ (* 2 (sqrt 2))
              9801)
           (pi-aux (- k 1)))))
)
```

(b) The Pell numbers are an infinite sequence of integers which correspond to the denominators of the closest rational approximations of $\sqrt{2}$. The Pell numbers are defined by the following recurrence relation (which looks very similar to the Fibonnacci sequence):

$$P_n = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ 2P_{n-1} + P_{n-2} & \text{otherwise} \end{cases}$$

Use this recurrence relation to write a recursive function, `pell-num`, which takes one parameter, n, and returns the n^{th} Pell number.

The numerator for the rational approximation of $\sqrt{2}$ corresponding to a particular Pell number is **half** of the corresponding number in the sequence referred to as the *companion Pell numbers* (or Pell-Lucas numbers). The companion Pell numbers are defined by the recurrence relation:

$$Q_n = \begin{cases} 2 & \text{if } n = 0 \\ 2 & \text{if } n = 1 \\ 2Q_{n-1} + Q_{n-2} & \text{otherwise} \end{cases}$$

(c) Use this recurrence relation to write a function, named `comp-pell-num`, which returns the n^{th} companion Pell number.

(d) Finally write a function that uses the Pell number and companion Pell number functions to compute the n^{th} approximation for $\sqrt{2}$. Use your new function to compute the approximation for $\sqrt{2}$ for the sixth Pell and companion Pell numbers.

3. It is an interesting fact the the square-root of any number may be expressed as a *continued fraction*. For example,

$$\sqrt{x} = 1 + \cfrac{x-1}{2 + \cfrac{x-1}{2 + \cfrac{x-1}{\ddots}}}$$

Write a Scheme function called *new-sqrt* which takes two formal parameters x and n, where x is the number we wish to find the square root of and n is the number of continued fractions to compute recursively. Demonstrate that for large n, *new-sqrt* is very close to the builtin *sqrt* function.