Scheme

1. Special Forms

(a) \textit{define} - \texttt{(define (name arg1 arg2 ... body)}

\hspace{1cm} \text{Syntactic sugar for the following: (define name (lambda (arg1 arg2 ... body))}

(b) \textit{cond} - \texttt{(cond (test consequent) (test consequent) ... (else alternative))}

\hspace{1cm} \text{Alternative to if when there are more than two cases. The value returned is the consequent where the first test evaluates to true (anything but \#f). If no tests are true, evaluate and return the alternative, if any. The alternative \texttt{else} is optional. If a consequent is omitted, the value of the test is returned.}

Problems

1. Consider the following definitions:

\begin{verbatim}
(define (our-display x)
  (display x) ;this prints x to the screen
  x) ;this returns x as the value

(define (count1 x)
  (cond ((= x 0) 0)
       (else (our-display x)
          (count1 (- x 1))))

(define (count2 x)
  (cond ((= x 0) 0)
       (else (count2 (- x 1))
          (our-display x))))
\end{verbatim}

What will \texttt{(count1 4)} and \texttt{(count2 4)} display?
2. Write a procedure `fact` that computes the factorial of a number `n`.
   Plan:

3. Write a procedure that computes $e$.
   Plan:

4. Write an iterative procedure that computes $e$.
   Plan:
5. Write a procedure \texttt{fib} that computes the \(n^{th}\) fibonacci number.
   Plan:

6. Write a procedure that computes the golden ratio, \(\phi\).
   Plan: