Scheme

1. Special Forms

   (a) \textit{and} - (\texttt{and} \ arg_1 \ arg_2 \ldots)
   
   Evaluates arguments from left to right, stopping at the first one that evaluates to false and returning false. Should all the arguments evaluate true-ishly, returns the value of the last argument.

   (b) \textit{or} - (\texttt{or} \ arg_1 \ arg_2 \ldots)
   
   Evaluates arguments from left to right, stopping at the first one that evaluates to true-ish and returns that value. Should all the arguments evaluate to false, returns false.

Higher Order Procedures

\begin{verbatim}
(define sum
    (lambda (f x y dx)

(\texttt{define sum
    (lambda (f x y dx)}
\end{verbatim}
Types

Problems

For each expression, write the type of the value that results from evaluating the expression. Ignore define expressions.

4
(+ 1 1)
(lambda (x) (+ x 1))
(lambda (x) (= x 1))
(define square
  (lambda (x) (* x x)))
square
(square 5)
(define a
  (lambda (f) (+ (f 5) 1)))
a
(a square)
(define b
  (lambda (x y)
    (+ (a x) y)))
b
(b square 4)
(define c
  (lambda (x)
    (lambda (y)
      (+ x y))))
c
(c 5)