2. Define a function `atom-depth` which takes a list `ls` as its argument and returns a list where every atom is replaced by its depth in the list. Here, depth is defined to be the number of left and right parentheses which enclose the atom. For example,

   > (atom-depth '(a b c)) returns
     (1 1 1)
   > (atom-depth '(a (b) c))
     (1 (2) 1)
   > (atom-depth '(((foo))))
     ((((3))))

   Hint: Use a helper function with an extra parameter.

3. Define a function `integer->natural` which takes an integer `i` as an argument and returns a natural number represented as a nested list of depth `n`. For example,

   > (integer->natural 0)
     ()
   > (integer->natural 1)
     ()
   > (integer->natural 2)
     ()

4. Define a function `natural->integer` which takes a natural `n` as its argument (defined as above) and returns the corresponding integer `i`. For example,
5. Without using any arithmetic functions, define a function \textit{times} which takes two natural numbers \( n \) and \( m \) as arguments (defined as above) and returns a list of nested parentheses of depth \( n \times m \): For example,

\[
\begin{align*}
> & \ (\text{times} \ (\text{integer->natural} \ 0) \ (\text{integer->natural} \ 37)) \\
> & \ () \\
> & \ (\text{natural->integer} \ (\text{times} \ (\text{integer->natural} \ 2) \ (\text{integer->natural} \ 2))) \\
& \ 4 \\
> & \ (\text{natural->integer} \ (\text{times} \ (\text{integer->natural} \ 4) \ (\text{integer->natural} \ 2))) \\
& \ 8
\end{align*}
\]

6. Define a function \textit{iota-iota} which takes an integer \( i \) as its argument and returns a list of pairs of integers such that

\[
\begin{align*}
> & \ (\text{iota-iota} \ 1) \\
& \ ((1 \ . \ 1)) \\
> & \ (\text{iota-iota} \ 2) \\
& \ ((1 \ . \ 1) \ (1 \ . \ 2) \ (2 \ . \ 1) \ (2 \ . \ 2)) \\
> & \ (\text{iota-iota} \ 3) \\
& \ ((1 \ . \ 1) \ (1 \ . \ 2) \ (1 \ . \ 3) \ (2 \ . \ 1) \ (2 \ . \ 2) \ (2 \ . \ 3) \ (3 \ . \ 1) \ (3 \ . \ 2) \ (3 \ . \ 3))
\end{align*}
\]

All helper functions should be tail-recursive and should be defined within the body of \textit{iota-iota} using \textit{letrec}.

7. The definition of \textit{append} given in class requires \( O(n^2) \) calls to \textit{cons}. Give a definition for \textit{append} which uses only \( O(n) \) calls to \textit{cons}. All helper functions should be tail-recursive and should be defined within the body of \textit{append} using \textit{letrec}. Warning: This problem is challenging.