1. Functional vs. Imperative Languages:
   Provide at least two differentiating factors between functional and imperative languages.

2. Nested scopes (Bindings):
   Consider the following fragment of code in Scheme:

   ```scheme
   (let ((a 3))
     (let ((a 4)
           (b a))
       (+ a b)))
   ```

   What does this function evaluate to and why?

3. List manipulation:
   What is the car of the cdr of the following list in Scheme?

   ```scheme
   +-----+ +-+-----+ ++-----+ +-----+
   bar | * +--- | * | ++ +--- | * | * |
        +-----+ +-----+ +-----+ +-----+
           \|/ \|/ \|/
        22 15  6
   ```

4. A simple program in Scheme:

   Recall that the fibonacci numbers are a set of numbers formed by adding the last two numbers in the series to get the next.

   \[ f(0) = 1, f(1) = 1, \text{and } f(n) = f(n-1) + f(n-2) \text{ for } n > 1 \]

   Implement a function to return the \( n \)th fibonacci number in the series in Scheme.