Problem 1: Implement the following procedures for handling lambda-calculus expressions.

The constructors:
- \( \text{var-exp} : \ Var \rightarrow \text{Lc-exp} \)
- \( \text{lambda-exp} : \ Var \times \text{Lc-exp} \rightarrow \text{Lc-exp} \)
- \( \text{app-exp} : \ \text{Lc-exp} \times \text{Lc-exp} \rightarrow \text{Lc-exp} \)

The predicates:
- \( \text{var-exp?} : \ \text{Lc-exp} \rightarrow \text{Bool} \)
- \( \text{lambda-exp?} : \ \text{Lc-exp} \rightarrow \text{Bool} \)
- \( \text{app-exp?} : \ \text{Lc-exp} \rightarrow \text{Bool} \)

The extractors:
- \( \text{var-exp>var} : \ \text{Lc-exp} \rightarrow \text{Var} \)
- \( \text{lambda-exp>bound-var} : \ \text{Lc-exp} \rightarrow \text{Var} \)
- \( \text{lambda-exp>body} : \ \text{Lc-exp} \rightarrow \text{Lc-exp} \)
- \( \text{app-exp>rator} : \ \text{Lc-exp} \rightarrow \text{Lc-exp} \)
- \( \text{app-exp>rand} : \ \text{Lc-exp} \rightarrow \text{Lc-exp} \)

---

1. EOPL, p.42-43
Problem 2\textsuperscript{2}: Implement the four required operations for bigits. Then use your implementation to calculate the factorial of 10. How does the execution time vary as this argument changes? How does the execution time vary as the base changes? Explain why.

Problem 3\textsuperscript{3}: Consider the data type of stacks of values, with an interface consisting of the procedures empty-stack, push, pop, top, and empty-stack?. Write a specification for these operations in the style of the example above. Which operations are constructors and which are observers?

Problem 4\textsuperscript{4}: We can use any data structure for representing environments, if we can distinguish empty environments from non-empty ones, and if we can can extract the pieces of a non-empty environment. Implement environments using a representation in which the empty environment is represented as the empty list, and in which extend-env builds an environment that looks like

This is called an a-list or association-list representation.

Problem 5\textsuperscript{5}: Add to the environment interface an observer called has-binding? that takes an environment env and a variable s and tests to see if s has an associated value in env. Implement it using the a-list representation.

\textsuperscript{2} EOPL, p.34, Exercise 2.1
\textsuperscript{3} EOPL, p.37, Exercise 2.4
\textsuperscript{4} EOPL, p.39, Exercise 2.5
\textsuperscript{5} EOPL, p.39, Exercise 2.9