Answer all of the questions below. The organization and neatness of your answers, as well as the legibility of your writing will count for 10% of your grade.

Name and section: _____

Instructor's name: _____

1. (a) Describe the three fundamental programming paradigms and relate them to the notions of an algorithm described by *Post grammars*, the λ-*calculus*, and *Turing machines*.
   (b) For each of the languages studied in class (Prolog, Lisp/Scheme, and Go), identify which paradigm it's an exemplar of, and the key ideas which informed its development.
2. Define a relation in Prolog called `slice(L, M, N, S)` which is read as follows: "S is the slice of the list L consisting the elements L[M], ..., L[N]". Assume 0-based indexing. Note that a slice of a list is itself a list. If the interval defined by M and N is emtpy, then S will be an empty list.
3. A digraph may be described by a collection of the facts defined by the following pair of relations:
   - `node(X)` - read as "X is a node."
   - `edge(X, Y)` - read as "there is an edge from X to Y."

   (a) Define a relation `path(X, Y)` which is satisfied if and only if there exists a path in the digraph from X to Y.
   (b) Define a relation `cyclic` which is satisfied if and only if the graph contains a cycle.
4. Describe how the list operations of `car` and `cdr` in Lisp/Scheme led to the development of the linked-list data structure.
5. Define a function `iota` in Scheme which satisfies the following claim: If N is a natural number then `(iota N)` is the list of integers from 0 to N non-inclusive.
6. Define a function `f` in Scheme which satisfies the following claim: If N is a natural number then `(f N)` is the sum of the digits of N!.
7. (a) Define a function `sum` in Go which satisfies the following claim: If `ch` is a bounded channel of integers then `sum(ch)` is the sum of all of the integers that are sent to the channel.
   (b) The function `sum` from the previous part, implement a solution in Go to Problem 6. Structure your algorithm to make use of channels and goroutines.
8. What is the primary distinction between the procedural/imperative programming paradigm and the declarative/functional programming paradigms?
9. Describe the notion of *tail recursion* and what distinguishes it from standard recurion. Detail an example of the benefits of recursion by writing a tail recursive, and standard recursive definition of a function in Scheme and comparing how their corresponding invocations are evaluated.