Programming Language Principles
Homework 7

1. No, they are not contradictory. When there are consecutive identical operators within an expression, associativity determines which subexpressions are arguments of which operators. It does not determine the order in which those subexpressions are evaluated. For example, left associativity for subtraction determines that \( f(a) - g(b) - h(c) \) groups as \( (f(a) - g(b)) - h(c) \) (rather than \( f(a) - (g(b) - h(c)) \)), but it does not determine whether \( f \) or \( g \) is called first.

2. Let \( \sim \) represent unary negation (yes, it’s needed).
   - Postfix: \( b \sim b b * 4 a * c * - sqrt + 2 a * / \).
   - Prefix: \( / + \sim b sqrt - * b b * * 4 a c * 2 a \).

3. Answer: Without parentheses, would \( -2 3 * 4 5 6 \) evaluate to \( -(2 3 (* 4 5) 6) \) = -27 or to \( (-2 3 (* 4 5 6)) \) = -121?
   More accurately, issues of precedence and associativity do not arise with prefix or postfix notation in which each operator takes a fixed number of operands.

4. It’s not entirely clear whether this statement was meant to be serious or humorous. Economy of syntax at this level does not appear to be a compelling concern. More relevant, arguably, is the likelihood of confusion: can the wrong operator easily be used by mistake, and can a human reader of the code misinterpret the programmer’s intent? The problem in C is not so much the specific choice of operators for assignment and equality testing, but rather the similarity of those operators and the fact that both are permissible in an expression context.

5. Yes, it’s more than a coincidence. Where a language with a value model of variables tends to propagate copies of objects, a language with a reference model tends to propagate references to the same object. This proliferation of references makes it very difficult for the programmer to keep track of the number of references to any given object, and specifically of whether a given object is no longer referenced at all. Garbage collection becomes more or less essential.