1. Which of the following Hoare triples are correct?
   (a) \{x<y\} t:=x; x:=y; y:=t; \{y<x\}. Can you write a stronger postcondition?
   (b) \{false\} x:=2; \{x=4\}
   (c) \{x=1\} x:=2; \{x=4\}
   (d) \{x>0\} while x!=0 do x++; \{x=0\}
   (e) \{x>0\} while x!=0 do x++; [x=0]
   (f) \{x<0 and x\in\text{Integer}\} while x!=0 do x++; \{x=0\}
   (g) \{x<0 and x\in\text{Integer}\} while x!=0 do x++; [x=0]

2. Compute the following weakest preconditions, and simplify the result where possible.
   (a) WP(t:=x; x:=y; y:=t, y<x)
   (b) WP(x:=2, x=4)
   (c) WP(x:=2, x=2)

3. Compute (and simplify where possible) VC(while x!=0 do x++, x=0)
   (a) using the loop invariant false
   (b) using the loop invariant x<0 and x\in\text{Integer}

4. Show how to compute a satisfying assignment for
   \[(a \lor b \lor c) \land (\neg b \lor c) \land (a \lor b) \land (a \lor c) \land (\neg b \lor \neg c) \land (\neg a \lor c)\]
   using the Davis-Putnam procedure. (Hint: Plug the resulting satisfying assignment in to the
   formula to check your work.)

5. In class, we considered how to compute weakest preconditions for a language with assignment,
   sequential composition, if expressions, and a throw construct that throws an exception to the
   nearest enclosing try s1 catch s2 construct.
   Suppose we extended this language with a “stopwhen E” statement that stops program com-
   putation if the expression E is true. Show how to to compute weakest preconditions for this
   language in a way that supports reasoning about the final program state.
   Hint: Use a wp function that takes three postconditions.