CSE 331 Autumn 2017 Homework 1

Notes:

- You may use any standard symbols for logical operators (e.g., V for "or").
- Assume that:
  - all numbers are integers
  - integer overflow will never occur
  - division is truncating integer division (as in Java)

1. **Hoare triples.** State whether each Hoare triple is valid. If it is invalid, give a counterexample.

   a. {{ x >= 0 }}
   y = 5*x;
   {{ y > 0 }}

   b. {{ x > y }}
   z = x – y;
   {{ z > 0 }}

   c. {{ }}
   if (x > 15)
   y = x % 12;
   else
   y = x+1;
   {{ y < 17 }}

   d. {{ x > 10 }}
   if (x > 0)
   x = 100;
   else
   x = -1;
   {{ x > 10 }}

2. **Weakest conditions.** Circle the weakest condition in each set.

   a. {{ a < 6 }} {{ a = 2 }} {{ a <= 6 }}

   b. {{ b < 20 }} {{ b != 20 }} {{ b = 0 }}

   c. {{ x < 0 and y > 0 }} {{ x < 0 or y > 0 }}

   d. {{ |a+b| < c }} {{ a+b < c }}

3. **Forward reasoning with assignment statements.** Write an assertion in each blank space indicating what is known about the *program state*, given the precondition and the previously executed statements. The first assertion in part (a) is supplied as an example.

   Additional rules for this problem:

   - Simplify your assertions but **do not weaken them**.
   - Rewrite your assertions to only refer to the current state of variables (no subscripts).

   a. {{ }}
   `x = 5;`
   {{ x = 5 }}
   y = -2 * x;
   {{ _____ }}
   z = y - 4;
   {{ _____ }}
   x = z / 2;
   {{ _____ }}
   y = 1;
   {{ _____ }}

   b. {{ x != 0 }}
   y = x;
   {{ _____ }}
   y = y - 2;
   {{ _____ }}

   c. {{ |x| > 20 }
   x = -x;
   {{ _____ }}
   x = x / 5;
   {{ _____ }}
   x = x - 4;
   {{ _____ }}

   d. {{ y > 4*x }}
   y = y * 2;
   {{ _____ }}
   x = x + 3;
   {{ _____ }}

4. **Backward reasoning with assignment statements.** Find the weakest precondition for each sequence using backward reasoning, and write the appropriate assertion in each blank space.

   a. {{ _____ }}
   x = x * 2;
   {{ _____ }}
   y = 3 + x;
   {{ y > 15 }

   b. {{ _____ }}
   y = w * 6;
   {{ _____ }}
   x = 2 - x;
   {{ x >= y }}

   c. {{ _____ }}
   t = 4 * s;
   {{ _____ }}
   r = w - 6;
   {{ _____ }}
   s = w + 3*s;
   {{ r >= s and s >= t }}

5. **Backward reasoning with if/else statements.** Find the weakest precondition for the following conditional statement using backward reasoning, inserting the appropriate assertion in each blank.

   {{ _____ }}
   if (x >= 0)
   {{ _____ }}
   z = -x;
   {{ _____ }}
   else
   {{ _____ }}
   z = x + 2;
   {{ _____ }}
   {{ z != 0 }

6. **Verifying correctness.** For each block of code, fill in the intermediate assertions (working either forward or backward or some combination), then use them to state whether the code is correct. I.e., whether the Hoare triples for the pre- and post-condition are valid.

   a. {{ x >= 9 }}
   y = x – 10;
   {{ _____ }}
   z = 2 * y;
   {{ _____ }}
   z = z + 2;
   {{ z > 0 }}

   b. {{ 4x >= w }}
   y = w – 1;
   {{ _____ }}
   x = 4*x;
   {{ _____ }}
   z = x – 1;
   {{ z >= y }}

   c. {{ s >= 0 }}
   if (s == t)
   {{ _____ }}
   s = -1;
   {{ _____ }}
   else
   {{ _____ }}
   s = t - 2;
   {{ _____ }}
   {{ s < t }}