CS2205: Theory of Computation
Problem Class X (Revision)

1. Consider this equivalence:
$$W_1 \Leftrightarrow (W_1 \wedge W_2) \equiv W_1 \Rightarrow W_2$$
   (a) Demonstrate that it holds by drawing a truth-table.
   (b) Demonstrate that it holds using the laws of the algebra of propositions.
   (c) Demonstrate that it holds by showing the following:
      i. $\{W_1 \Leftrightarrow (W_1 \wedge W_2)\} \vdash W_1 \Rightarrow W_2$
      ii. $\{W_1 \Rightarrow W_2\} \vdash W_1 \Leftrightarrow (W_1 \wedge W_2)$

2. Show whether this inference rule is sound or not:
$$\frac{W_1 \Rightarrow W_2, \neg W_2}{\neg W_1}$$

3. Determine the weakest precondition P that satisfies the following. Simplify as much as you can.
   (a) $(\!| P |\!)\, x := x - 1\, (\!| x^2 + 2x + 1 = 0 |\!)$
   (b) $(\!| P |\!)\, x := y - 1; y := x - 1\, (\!| x > 0 |\!)$
   (c) $(\!| P |\!)$ **if** $x \geq 1$ $x := x + 1$ **else** $x := x - 1\, (\!| x \neq 1 |\!)$

4. Prove that $(\!|$ **True** $|\!)\, ProgA\, (\!| 0 \leq x \wedge x \leq 5 |\!)$ where *ProgA* is as follows:

```
if x > 5
{    x := 5;
}
else
{    if x < 0
     {    x := 0;
     }
}
```


5. Prove that $(\!| 0 \leq n |\!)\, ProgB\, (\!| x^2 \leq n \wedge n < (x+1)^2 |\!)$ where *ProgB* is below. In other words, prove that *ProgB* computes the largest integer that does not exceed the square root of n (i.e. the square root of n, rounded down). It's your job to come up with the invariant.

```
x := 0;
while (x + 1)^2 ≤ n
{    x := x + 1;
}
```

6. Prove that $(\!| 0 \leq n |\!)\, ProgC\, (\!| x^2 \leq n \wedge n < (x+1)^2 |\!)$ where *ProgC* is below. In other words, prove that *ProgC* also computes the largest integer that does not exceed the square root of n (i.e. the square root of n, rounded down). It's your job to come up with the invariant.

```
x := n;
while x^2 > n
{    x := x - 1;
}
```

7. *ProgB* and *ProgC* above both compute the largest integer that does not exceed the square root of n. Which, if either, do you think is the more efficient and why?