Exercise 4F-2. VCGen for Let [6 points].

The bug is that the substitution of x with e is not scoped to be only inside of the let statement. This can be fixed with variable renaming of x within c before substitution. So the proper procedure would be to first rename x to x' (where x' is assumed to be a variable that is not used elsewhere) within c, then compute the verification condition of c, then substitute x' with e. A correct rule is:

$$\text{VC}(\mathsf{let}\ x = e\ \mathsf{in}\ c, B) = \ \text{VC}([x'/x]c, B)[e/x']$$

Question assigned to the following page: 3

Exercise 4F-3. VCGen Mistakes [6 points].

1. command c:
$$\mathsf{let}\ x = 1\ \mathsf{in}\ (y := x + 1)$$

2. Post-condition B:
$$x < y$$

3. State σ:
$$\{x = 5, y = 0\}$$

4. $\sigma \models \mathrm{VC}(c, B)$
Since the buggy verification condition (VC) substitutes e (or in this case 1) for x everywhere, the VC will always compute to:
$$1 < 2$$
which is always true.

5. $\langle c, \sigma \rangle \Downarrow \sigma'$
Execution of c results in:
$$\langle c, \sigma \rangle \Downarrow \sigma'$$
where $\sigma' = \{x = 5, y = 2\}$.

6. $\sigma' \not\models B$
In σ', the condition $x < y$ evaluates to:
$$5 < 2$$
which is false. Therefore the actual state after execution of c does not satisfy the post-condition B.

Question assigned to the following page: 4

Exercise 4F-4. Axiomatic Do-While [6 points].

$$\frac{\vdash \{A\}c\{C\} \quad \vdash \{C \wedge b\}c\{C\} \quad \vdash C \wedge \neg b \Rightarrow B}{\vdash \{A\}\mathsf{do}\ c\ \mathsf{while}\ b\{B\}}$$

Where C is the loop invariant.