Handout #12
Analysis of Programs, Spring 2003

Sample Solutions to Test #2

1. (a) The set of program states $wp(S, q)$ is the largest set of program states such that starting from a program state in $wp(S, q)$ and if $S$ terminates, then the resulting program state is in $q$. (b) Compute $wp(S, q)$, then prove $p \Rightarrow wp(S, q)$.

2. (a) Let $S$ be $x := e$, where $x$ is a variable and $e$ is an expression, then the rule to compute $wp(S, q)$ is: replace every free occurrence of $x$ in $q$ by $e$. (b) $wp(x := x + y, x > 2y) = x + y > 2y = x > y$.

3. (a) $wp(S_1, wp(S_2, q))$ (b) $wp(S_1, wp(S_2, wp(S_3, q)))$ (c)

\[ wp(S, x > y) = \{\text{part (b)}\} \]
\[ \quad wp(z := x + y, wp(x := y + z, wp(y := x + z, x > y))) \]
\[ = \{wp \text{ for assignments}\} \]
\[ \quad wp(z := x + y, wp(x := y + z, 0 > z)) \]
\[ = \{wp \text{ for assignments}\} \]
\[ \quad wp(z := x + y, 0 > z) \]
\[ = \{wp \text{ for assignments}\} \]
\[ \quad 0 > x + y \]

4. (a) $wp(\text{IF}, q) = (C \land wp(S_1, q)) \lor (\neg C \land wp(S_2, q))$ (b)

\[ wp(\text{IF}, x > y) = \{\text{part (a)}\} \]
\[ \quad (x \leq y \land wp(x := y, x > y)) \lor (x > y \land wp(x := x + y, x > y)) \]
\[ = \{wp \text{ for assignments}\} \]
\[ \quad (x \leq y \land y > y) \lor (x > y \land x > 0) \]
\[ = \{\text{simplification}\} \]
\[ \quad x > y \land x > 0 \]

5. Find the loop invariant $I$; prove $p \Rightarrow wp(S_1, I)$; prove $I \land C \Rightarrow wp(S_2, I)$; prove $I \land \neg C \Rightarrow q$.

6. Recall that total correctness means partial correctness plus termination. The example program is while (true) skip, the precondition is true, and the postcondition is true. It is easy to show that this program is correct with respect to the given pre- and post-conditions, i.e., the program is partially correct. The program, however, is not totally correct, because it does not terminate.

7. To show that the program terminates, we need to show that $f(m, n)$ is always non-negative, and each loop iteration strictly decreases $f(m, n)$. Since $0 \leq m \leq n$ is a loop invariant, we have $f(m, n) \geq 0$ by simple arithmetic. If the if statement takes the first branch, then $f(m, n)$ decreases by $1$ because $m$ increases by $1$; if the if statement takes the second branch, then $m = n$, because $0 \leq m \leq n$ is a loop invariant and $m \geq n$ is the negation of the condition of the if statement. Hence, $f(m, n)$ decreases by $(n + 1)(n + 2)/2 - m - ((n - 1 + 1)(n - 1 + 2)/2 - 0) = n + 1 - n = 1$. Therefore, each loop iteration strictly decreases $f(m, n)$, and since $f(m, n) \geq 0$, the loop terminates eventually.