SECTION 4.5 Program Correctness

2. There are two cases. If $x \geq 0$ initially, then nothing is executed, so $x \geq 0$ at the end. If $x < 0$ initially, then x is set equal to 0, so $x = 0$ at the end; hence again $x \geq 0$ at the end.

4. There are three cases. If $x < y$ initially, then min is set equal to x, so $(x \leq y \wedge min = x)$ is true. If $x = y$ initially, then min is set equal to y (which equals x), so again $(x \leq y \wedge min = x)$ is true. Finally, if $x > y$ initially, then min is set equal to y, so $(x > y \wedge min = y)$ is true. Hence in all cases the disjunction $(x \leq y \wedge min = x) \vee (x > y \wedge min = y)$ is true.

6. There are three cases. If $x < 0$, then y is set equal to $-2|x|/x = (-2)(-x)/x = 2$. If $x > 0$, then y is set equal to $2|x|/x = 2x/x = 2$. If $x = 0$, then y is set equal to 2. Hence in all cases $y = 2$ at the termination of this program.

8. We prove that Algorithm 8 in Section 4.4 is correct. It is clearly correct if $n = 0$ or $n = 1$, so we assume that $n \geq 2$. Then the program terminates when the **for** loop terminates, so we concentrate our attention on that loop. Before the loop begins, we have $x = 0$ and $y = 1$. Let the loop invariant p be "$(x = f_{i-1} \wedge y = f_i) \vee (i$ is undefined $\wedge\, x = f_0 \wedge y = f_1)$." This is true at the beginning of the loop, since i is undefined and $f_0 = 0$ and $f_1 = 1$. What we must show now is $p \wedge (1 \leq i < n)\{S\}p$. If $p \wedge (1 \leq i < n)$, then $x = f_{i-1}$ and $y = f_i$. Hence z becomes f_{i+1} by the definition of the Fibonacci sequence. Now x becomes y, namely f_i, and y becomes z, namely f_{i+1}, and i is incremented. Hence for this new (defined) i, $x = f_{i-1}$ and $y = f_i$, as desired. We therefore conclude that upon termination $x = f_{i-1} \wedge y = f_i \wedge i = n$; hence $y = f_n$, as desired.

10. We must show that if p_0 is true before S is executed, then q is true afterwards. Suppose that p_0 is true before S is executed. By the given conditional statement, we know that p_1 is also true. Therefore, since $p_1\{S\}q$, we conclude that q is true after S is executed, as desired.

12. Suppose that the initial assertion is true before the program begins, so that a and d are positive integers. Consider the following loop invariant p: "$a = dq + r$ and $r \geq 0$." This is true before the loop starts, since the equation then states $a = d \cdot 0 + a$, and we are told that a (which equals r at this point) is a positive integer, hence greater than or equal to 0. Now we must show that if p is true and $r \geq d$ before some pass through the loop, then it remains true after the pass. Certainly we still have $r \geq 0$, since all that happened to r was the subtraction of d, and $r \geq d$ to begin this pass. Furthermore, let q' denote the new value of q and r' the new value of r. Then $dq' + r' = d(q+1) + (r-d) = dq + d + r - d = dq + r = a$, as desired. Furthermore, the loop terminates eventually, since one cannot repeated subtract the positive integer d from the positive integer r without r eventually becoming less than d. When the loop terminates, the loop invariant p must still be true, and the condition $r \geq d$ must be false—i.e., $r < d$ must be true. But this is precisely the desired final assertion.