In Class Exercises – Axiomatic Semantics
Sample Solutions

1. (a) INVALID; $3 \neq 4$.
   (b) INVALID; statement ends, but postcondition is false.
   (c) VALID; precondition is false.
   (d) INVALID; if y is negative and $|y| > x$ the postcondition will be false.
   (e) VALID; $y > x \wedge x > 0$, so $y > 0$. Therefore $x + y > y$.
   (f) VALID; statement never terminates, so postcondition irrelevant (for partial correctness)
   (g) VALID; statement terminates and postcondition is true.
   (h) VALID; precondition is false.

2.

	Result of assignment rule	Simplifies to
(a)	$Q : x * 3 = 15$	$Q : x = 5$
(b)	$Q : 14 = 14$	$Q : true$
(c)	$Q : (r-1)^3 + sin(r-1) - 12 = 4$	same
(d)	$Q : x + a[y+1] = \sum_1^{y+1} a[k]$	$Q : x = \sum_{k=1}^{y} a[k]$

3. $\{Q : (w = z \wedge z = 40)$ **or** $(w \neq z \wedge (w = 0 \vee z = 10))\}$
   if $w = z$ **then begin**
   $\{z = 40\}$
   $\{z = 10 * 4\}$
   $w := 4;$
   $\{z = 10 * w\}$
   $\{z \div 2 = 5 * w\}$
   $z := z \div 2$
   $\{z = 5 * w\}$
   end
   else begin
   $\{w = 0 \vee z = 10\}$
   $\{w * z = 10 * w\}$
   $z := w * z;$
   $\{z = 10 * w\}$
   $\{z = 5 * 2 * w\}$
   $w := 2 * w$
   $\{z = 5 * w\}$
   end
   $\{z = 5 * w\}$

4. • $\{Q\}$ Initialization $\{P\}$

   $\{0 < n\}$
   $i := 1;$
   $\{$`inv` P`:` $(0 < i \leq n) \wedge (i$ `is a power of` $2)\}$

   1 is assigned to i. From the precondition $0 < n$, so the first clause is satisfied. 1 is a power of 2, so the second clause is satisfied.

   • $\{P \wedge B\}$ S $\{P\}$

$\{(0 < i \le n) \wedge (i \texttt{ is a power of 2}) \wedge (2 * i \le n)\}$
$i := i * 2$
$\{\texttt{inv } P\texttt{:} \quad (0 < i \le n) \wedge (i \texttt{ is a power of 2})\}$

The precondition tells us that $2 * i \le n$, so assigning $2 * i$ to i will result in $i \le n$. $i > 0$ initially, and doubling i will maintain that. i is a power of 2 before the assignment statement, so doubling i will maintain that as well.

- $P \wedge \neg B \Rightarrow R$

  $(0 < i \le n) \wedge (i \texttt{ is a power of 2}) \wedge \textbf{not}(2 * i \le n)$
  $(0 < i \le n) \wedge (i \texttt{ is a power of 2}) \wedge (2 * i > n)$
  $(0 < i \le n) \wedge (n < 2 * i) \wedge (i \texttt{ is a power of 2})$
  $(0 < i \le n < 2 * i) \wedge (i \texttt{ is a power of 2})$

- Prove bound function t decreases

  i doubles each time through the loop. The bound function is $n - i$. If the loop guard was true, $2 * i < n$, so doubling i decreases $n - i$ in each iteration.

- $P \wedge B \Rightarrow t > 0$

  $(0 < i \le n) \wedge (i \texttt{ is a power of 2}) \wedge (2 * i \le n)$

  Since $2 * i \le n$ it is the case that $i \ne n$. Therefore $n - i > 0$.

5.
- Find a loop guard B.

  $\{\texttt{pre } Q\texttt{:} \quad n > 0\}$
  $\{\texttt{inv } P\texttt{:} \quad (1 \le i \le n) \wedge (a = F_i) \wedge (b = F_{i-1})\}$
  $\{\texttt{post } R\texttt{:} \quad a = F_n\}$

  The answer to the question "When are we done?" is "When P and R mean the same thing." That happens when $i = n$; when that happens, $a = F_n$. Therefore, we are *not* done when $i \ne n$, and that is our loop guard.

- Find initialization to establish invariant P.

  At the beginning we only know that $n > 0$. To establish P, we need to initialize a, b, and i. If $i = 1$, the job is easy; $a = 1$ and $b = 0$. So our initialization is

  $\{\texttt{pre } Q\texttt{:} \quad n > 0\}$
  $i := 1; a := 1; b := 0$
  $\{\texttt{inv } P\texttt{:} \quad (1 \le i \le n) \wedge (a = F_i) \wedge (b = F_{i-1})\}$

- Guess bound function t and find ways to decrease it.

  As a first guess, let's try $n - i$. (We got this by looking at the invariant and seeing that i ranges between 1 and n.) We'll come back to this after our loop is finished to make sure it works.

  We can decrease t by incrementing i:

  $i := i + 1;$

- Ensure that P is reestablished.

  We need to add code to ensure that the following triple is valid.

  $\{\texttt{inv } P\texttt{:} \quad (1 \le i \le n) \wedge (a = F_i) \wedge (b = F_{i-1})\}$
  while $i \ne n$ **do begin**
  $\quad i := i + 1$ **end;**
  $\{\texttt{inv } P\texttt{:} \quad (1 \le i \le n) \wedge (a = F_i) \wedge (b = F_{i-1})\}$

After i is incremented, $a = F_{i-1}$ and $b = F_{i-2}$. Applying the formula from the problem, we get

$\{$`inv` P: $(1 \le i \le n) \wedge (a = F_i) \wedge (b = F_{i-1})\}$
while $i \ne n$ **do begin**
  $i := i + 1$;
  $temp := a$;
  $a := a + b$;
  $b := temp$;
  end;
$\{$`inv` P: $(1 \le i \le n) \wedge (a = F_i) \wedge (b = F_{i-1})\}$

Lastly, we check that $n - i$ is okay as a bound function $(P \wedge B \Rightarrow t > 0)$. Since $t = n - i$, and the loop guard is $i \ne n$, and i is initialized to 1 and is incremented by 1 in each iteration, i is always $\le n$. If there are iterations left, therefore, $i < n$, so $0 < n - i$.