Task 1.1

		$[n > 0]$
$i := n;$		$[n > 0 \wedge i = n]$
$\{\textbf{inv } i > 0\}$		
$\{\textbf{dec } i\}$		
$\text{while}(i > 1)\{$		$[i > 0 \wedge i > 1 \wedge i = i_0] \Rightarrow [i/2 > 0 \wedge i/2 < i_0]$
$\quad i := i/2$		$[i > 0 \wedge i < i_0]$
$\}$		$[n > 0 \wedge i = 1]$

Task 1.2

$$[|a| \geq 1 \wedge i = 0 \wedge a[i] \geq 1]$$

$x := 1;$

$k := 0;$

$\{\textbf{inv } x = 2^k \wedge x \leq a[i]\}$

$\{\textbf{dec } a[i] - x * 2\}$

$\text{while}(x * 2 \leq a[i])\{$

$\quad k := k + 1$

$\quad x := x * 2$

$\}$ $\quad [x = 2^k \wedge x \leq a[i] \wedge a[i] < 2^{k+1}]$

Task 2.1

state σ	P_1	P_2
$\sigma \models e_1 \wedge e_2$	Executes s_1	Executes $s_1 \,\square\, s_2$
$\sigma \models e_1 \wedge \neg e_2$	Executes s_1	Executes s_1
$\sigma \models \neg e_1 \wedge e_2$	Executes s_2	Executes s_2
$\sigma \models \neg e_1 \wedge \neg e_2$	Executes s_2	$\perp_e$

Task 2.2

(a) $M(\text{havoc } i; a[i] = 1, \sigma)$

$= \bigcup_{\sigma' \in \{\sigma[i \mapsto n] \mid n \in \mathbb{Z}\}} M(a[i] = 1, \sigma')$

$= \{\sigma[a[\sigma'(i)] \mapsto 1]\}$

or can just do this?

$= \{\sigma[a[\sigma[i \mapsto n]] \mapsto 1] \mid n \in Z\}$

(b) $M(\text{while } \{x > -10 \wedge x < 10 \rightarrow x := x + 1 \,\square\, x > -10 \wedge x < 10 \rightarrow x := x - 1\}, \{x = 1\})$

$= \{\{x = 10\}, \{x = -10\}\}$

With $\{x = 1\}$, both guards are true, so the program either keeps decrementing or incrementing x until it is either 10 or -10 non-deterministically.

(c) $M(\text{branch } \{x \geq y \rightarrow x := x + 2 \,\square\, y \geq x \rightarrow y := y + 1\}, \{x = 3, y = 3\}$

$= M(x := x + 2, \{x = 3, y = 3\}) \cup M(y := y + 1, \{x = 3, y = 3\})$

$= \{\{x = 5, y = 3\}\} \cup \{\{x = 3, y = 4\}\}$

$= \{\{x = 5, y = 3\}, \{x = 3, y = 4\}\}$

Thus,

$M((\text{branch } \{x \geq y \rightarrow x := x + 2 \,\square\, y \geq x \rightarrow y := y + 1\});\ \text{if } (x < y) \text{ then } \{z := y\}$
else $\{z := x\}, \{x = 3, y = 3\})$

$= \bigcup_{\sigma' \in \{\{x=5,y=3\},\{x=3,y=4\}\}} M(\text{if } (x < y) \text{ then } \{z := y\} \text{ else } \{z := x\}, \sigma')$

$= \{\{x = 5, y = 3, z = 5\}, \{x = 3, y = 4, z = 4\}\}$

Task 2.3

(a) $wlp(\text{branch } \{x \geq y \rightarrow max := x \,\square\, y \geq x \rightarrow max := y\}, max \geq 0)$

$= (x \geq y \rightarrow wlp(max := x, max \geq 0)) \wedge (y \geq x \rightarrow wlp(max := y, max \geq 0))$

$= (x \geq y \rightarrow [x/max](max \geq 0)) \wedge (y \geq x \rightarrow [y/max](max \geq 0))$

$= (x \geq y \rightarrow x \geq 0) \wedge (y \geq x \rightarrow y \geq 0)$

(b) $sp(x \geq y, \text{branch } \{x \geq y \rightarrow y := y + 1 \,\square\, y \geq x \rightarrow x := x + 1\})$

$= sp(x \geq y \wedge x \geq y, y := y + 1) \wedge sp(x \geq y \wedge y \geq x, x := x + 1)$

$= sp(x \geq y, y := y + 1) \wedge sp(x = y, x := x + 1)$

$= ([y_0/y](x \geq y) \wedge y = [y_0/y](y + 1)) \wedge ([x_0/x](x = y) \wedge x = [x_0/x](x + 1))$

$= (x \geq y_0 \wedge y = y_0 + 1) \wedge (x_0 = y \wedge x = x_0 + 1)$

(c) $wlp(\text{havoc } x; y := y + 1, y \geq \frac{x}{|x|})$

$= wlp(\text{havoc } x, wlp(y := y + 1, y \geq \frac{x}{|x|}))$

$= wlp(\text{havoc } x, y + 1 \geq \frac{x}{|x|})$

$= \forall x_0 \in \mathbb{Z}.[x_0/x](y + 1 \geq \frac{x}{|x|})$

$= \forall x_0 \in \mathbb{Z}.y + 1 \geq \frac{x_0}{|x_0|}$

Task 3.1

$s \equiv [y := x/3 || x := x + 6; z := x * 2]$ and $\sigma = \{x = 18\}$

(a)

$$
\begin{array}{ccc}
 & < s, \sigma > & \\
\swarrow & & \searrow \\
< [skip || x := x + 6; z := x * 2], \{x = 18, y = 6\} > & & < [y := x/3 || skip], \{x = 24, z = 48\} > \\
\downarrow & & \downarrow \\
< [skip || skip], \{x = 24, y = 6, z = 48\} > & & < [skip || skip], \{x = 24, z = 48, y = 8\} >
\end{array}
$$

(b) $M(s, \sigma) = \{\{x = 24, y = 6, z = 48\}, \{x = 24, z = 48, y = 8\}\}$

Looking at the evaluation graph, the ending states have $x = 24, z = 48$ and either $y = 6$ or $y = 8$.

This is due to using x in an assignment $(y := x/3)$ and updating it $(x := x + 6)$ as well.

Task 4.1

I spent about 4 hours on this.