MODULE *appex4_1_1*

EXTENDS *Integers*, *TLC*

-wfNext

```
--algorithm test {
variables x = 10, z = 2 * x, y = z + x ;
{
l1: assert x = 10 /\ z = 2 * x /\ y = z + x ;
y := z + x ;
l2: assert x = 10 /\ y = x + 2 * 10 ;

l3: print <x, y, z> ;
 }
 }
```

BEGIN TRANSLATION

VARIABLES x, z, y, pc

$vars \triangleq \langle x, z, y, pc\rangle$

$Init \triangleq$ Global variables
$\land x = 10$
$\land z = 2 * x$
$\land y = z + x$
$\land pc =$ "l1"

$l1 \triangleq \land pc =$ "l1"
$\land Assert(x = 10 \land z = 2 * x \land y = z + x,$
"Failure of assertion at line 11, column 4.")
$\land y' = z + x$
$\land pc' =$ "l2"
$\land$ UNCHANGED $\langle x, z\rangle$

$l2 \triangleq \land pc =$ "l2"
$\land Assert(x = 10 \land y = x + 2 * 10,$
"Failure of assertion at line 13, column 4.")
$\land pc' =$ "l3"
$\land$ UNCHANGED $\langle x, z, y\rangle$

$l3 \triangleq \land pc =$ "l3"
$\land PrintT(\langle x, y, z\rangle)$
$\land pc' =$ "Done"
$\land$ UNCHANGED $\langle x, z, y\rangle$

Allow infinite stuttering to prevent deadlock on termination.

$Terminating \triangleq pc =$ "Done" $\land$ UNCHANGED $vars$

$Next \triangleq\ l1 \lor l2 \lor l3$
$\qquad\qquad \lor\ Terminating$

$Spec \triangleq\ Init \land \Box[Next]_{vars}$

$Termination \triangleq\ \Diamond(pc = \text{“Done”})$

END TRANSLATION

$MAX \triangleq 32768$ 16 bits
$D \triangleq 0\,..\,32768$
$x \leq 32760$

$DD(X) \triangleq\ (X \in D)$

$Safety\_absence \triangleq\ \ DD(x)\ \land DD(y) \land\ DD(z)$

$Inv \triangleq$
$\quad \land\ pc = \text{“l1”} \Rightarrow\ \ x = 10 \land\ \ z = 2 * x \land y = z + x$
$\quad \land\ pc = \text{“l2”} \Rightarrow\ \ \ x = 10 \land y = x + 2 * 10$