

Pre y Post Condiciones e Invariantes.

1. Escribir los estados intermedios en el siguiente programa escrito Dafny. Asumiendo que $x=10$.

```
method MyMethod(x: int) returns (y: int)
{
    var a, b := 0,0;
    a := x + 3;
    if x < 20 {
        b := 32 - x;
    }
    else {
        b := 16;
    }
    y := a + b;
}
```

2. Para el programa del inciso 1, escriba aserciones que capturen los conjuntos de estados a los cuales se llegan despues de cada paso.

3. Muestre que las siguientes ternas de Hoare no se cumplen:

- (a) $\{\{true\}\} x:=2*y \{\{y \leq x\}\}$
- (b) $\{\{true\}\} x:=100 \{\{false\}\}$
- (c) $\{\{0 \leq x\}\} x:=x-1 \{\{0 \leq x\}\}$

4. Justifique por qué se cumplen las siguientes ternas de Hoare:

- (a) $\{\{x = 89\}\} y:=34 \{\{x = 89\}\}$
- (b) $\{\{x = 3\}\} x:=x+1 \{\{x = 4\}\}$
- (c) $\{\{0 \leq x < 100\}\} x:=x+1 \{\{0 < x \leq 100\}\}$

5. Para cada una de las siguientes ternas reemplazar ? con un predicado que haga la terna verdadera:

- (a) $\{\{?\}\} x:=400 \{\{x = 400\}\}$
- (b) $\{\{?\}\} x:=x+3 \{\{x\%2 = 0\}\}$
- (c) $\{\{?\}\} x:=65 \{\{y \leq x\}\}$
- (d) $\{\{?\}\} x,y := 2*x, x+y; \{\{0 \leq x \leq 100 \wedge y \leq x\}\}$

6. En el siguiente programa anote cada paso con una aserción que caracterice el conjunto de estados que se alcanza luego de ejecutar el comando correspondiente:

$$\{\{x = X \wedge y = Y\}\}$$

$x := x + y;$

$y := x - y;$

$x := x - y;$

$$\{\{x = Y \wedge y = X\}\}$$

7. (*) Usando *WP* demostrar que el siguiente programa es correcto, las variables **a** y **b** son booleanos, y **xor** es el o exclusivo:

$$\{\{x = X \wedge y = Y\}\}$$

$x := x \text{ xor } y;$

$y := x \text{ xor } y;$

$x := x \text{ xor } y;$

$$\{\{x = Y \wedge y = X\}\}$$

8. Calcular el *WP* para el siguiente código, considerando la postcondición $\{\{y \geq 0\}\}$.

```
if (x < 0) {  
    y := -x;  
}  
else {  
    y := x;  
}
```

9. Calcular el *WP* para el siguiente código, considerando la postcondición $\{\{x + y = 22\}\}$.

```
if (x < 20) {  
    x := 3;  
}  
else {  
    y := 2;  
}
```

10. (Teórico) Calcular el WP del siguiente programa con la postcondición $\{\{y < 10\}\}$.

```
if (x < 8) {
    if (x < 4) {
        x := x + 1;
    }
    else {
        y := 2;
    }
}
else {
    if (x < 32) {
        y := 1;
    } else { }
}
```

11. Decir cuáles de las siguientes combinaciones de ternas de Hoare son válidas:

- (a) $\{\{0 \leq x\}\} x := x + 1; \{\{-2 \leq x\}\} y := 0 \{\{-10 \leq 0\}\}$
- (b) $\{\{x \geq 0\}\} x := x + 1; x := x + 1; \{\{x \geq 2\}\}$
- (c) $\{\{x < 2\}\} y := x + 5; x := 2 * x; \{\{x < y\}\}$

Este estilo de ejercicio comenzar a analizar desde atrás hacia adelante

12. Calcular las precondiciones para los siguiente fragmentos de código, en donde la postcondición es $x + y \leq 100$:

- (a) $x := x + 1; y := x + y$
- (b) $y := x + y; x := x + 1$

13. Escriba los invariantes y variantes para el siguiente programa:

```
var a := new int[n]

{\{n > 0\}}

s, i := 0, 0;
while i < n {
    s, i := (s + a[i]), i + 1;
}
```

$\{\{s = \text{sum}(a, s.Length)\}\}$

Primero debe definir la función `sum(a:array<int>, j:int)` que suma todos los elementos del arreglo `a` hasta la posición `j` (sin incluir `j`). Trate de verificar la corrección del programa.

14. Escriba el código anterior en Dafny y verifique su solución utilizando el compilador de Dafny.

15. Escriba los invariantes y variantes (función de decremento o cota) para el siguiente código:

```
var a := new int [n]
 $\{\{n > 0\}\}$ 
```

```
s, i := 0, n;
while i > 0 {
    s, i := (s + a[n-1]), i-1;
}
```

```
 $\{\{s = \text{sum}(a, i, s.Length)\}\}$ 
```

Para esto deberá definir una función `sum(a:array<int>, i:int, j:int)` que dado un arreglo `a`, y dos índices `i, j` con `i <= j` calcula la suma de todos los elementos del arreglo desde `i` hasta `j-1`.

16. Escriba el código anterior en Dafny y verifique su solución utilizando el compilador de Dafny.