

1 Ejercicio 1.

(a)

```
proc buscar (in l: seq( $\mathbb{R}$ ), in elem:  $\mathbb{R}$ , out result:  $\mathbb{Z}$ ) {  
  Pre {elem  $\in$  l}  
  Post {0  $\leq$  result < |l|  $\wedge_L$  l[result = elem]}  
}
```

(b)

```
proc progresionGeometricaFactor2 (in l: seq( $\mathbb{Z}$ ), out result: Bool) {  
  Pre {True}  
  Post {result = True  $\longleftrightarrow$  ( $\forall i : \mathbb{Z}$ )(1  $\leq i < |l| \longrightarrow_L$  l[i] = 2 * l[i - 1])}  
}
```

(c)

```
proc minimo (in l: seq( $\mathbb{Z}$ ), out result:  $\mathbb{Z}$ ) {  
  Pre {True}  
  Post {( $\forall i : \mathbb{Z}$ )(0  $\leq i < |l| \longrightarrow$  l[i]  $\geq$  result)}  
}
```

2 Ejercicio 2.

3 Ejercicio 3.

- (a) I. {0} II. {1, -1} III. $\{\sqrt{27}, -\sqrt{27}\}$
(b) I. {3} II. {0, 2} III. {0, 1, 2, 3, 4, 5}
(c) I. {3} II. {0} III. {0}
(d) Para aquellas listas que tengan un único valor máximo

4 Ejercicio 4.

- (a) Está mal por ese \wedge que se encuentra a la mitad de la post condición. El valor 'a' no puede ser menor a cero y a la vez ser mayor o igual.
(b) Está mal porque no se contempla el caso en que a sea cero.
(c) Está correcto.
(d) Está correcto.
(e) Está mal. Por ejemplo puedo tener un $a < 0$, mi result $\neq 2 * b$ (lo que estará mal) y no estaría incumpliendo esa post-condición.
(f) Está correcto.

5 Ejercicio 5.

El algoritmo propuesto para esa especificación solo sirve para aquellos valores que no están en el rango [0, 1]. Entonces la pre-condición nueva sería la siguiente:

```
proc unoMasGrande (in x:  $\mathbb{R}$ , out result:  $\mathbb{R}$ ) {  
  Pre {x  $\notin$  [0, 1]}  
  Post {result > x}  
}
```