

**Tener en cuenta:**

- Cada ejercicio debe entregarse en **hojas separadas** numeradas y con el nombre y apellido al lado del número de ejercicio.
- Una vez terminadas las derivaciones de un ejercicio, escribir el programa resultante.
- Utilice el formato de derivación usado en clase.
- Sea prolijo.

1. a) Derivar una definición recursiva para la función especificada como

$$f.xs.n = \langle \exists as, bs, cs : xs = as ++ bs ++ cs \wedge \#bs = n : pares.bs \rangle$$

a donde *pares* está definida por:

$$\begin{aligned} pares.[] &\doteq True \\ pares.(x \triangleright xs) &\doteq (x \bmod 2 = 0) \wedge pares.xs \end{aligned}$$

b) ¿Qué calcula la función *f*?

2. Derivar el siguiente programa

```
Const N : Int;
Var a : array [0, N) of Int;
    r : Int;
{N > 0}
S
{r = ⟨Max i : 0 ≤ i < N : a.i * ⟨∑ j : 0 ≤ j < i : a.j⟩⟩}
```

**Nota:** No se puede usar  $\infty$  ni  $-\infty$  en el programa. Se puede usar el operador binario *max*.

**Ayuda:** Sean sistemáticos al realizar el ejercicio, i.e. deriven.

3. **(Ejercicios para libres)** Derivar el programa del Ejercicio 2 cambiando la postcondición por

$$\{r = \langle \text{Max } i : 0 \leq i < N \wedge a.i > 0 : a.i * \langle \sum j : 0 \leq j < i : a.j \rangle \rangle\}$$

y teniendo en cuenta lo siguiente:

- No se puede usar  $\infty$  ni  $-\infty$  en el programa y sí se puede reforzar la precondición.
4. Especificar con pre y poscondición los siguientes problemas. No olvidarse de declarar las variables y constante que aparecen en las especificaciones (Ver ejercicio 2):
- a) Hay tres elementos de un arreglo, en posiciones distintas, cuya multiplicación es igual a 24523.
  - b) Existe un elemento en el array tal que éste es menor al sucesor dividido 2.