

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 resultado final.
- Utilice el formato de derivación usado en clase.
- Sea prolijo.

1. Derivar una definición recursiva para la función especificada como

$$f.xs = \langle \text{Min } i : 0 \leq i < \#xs \wedge \text{sum}.(xs \uparrow i) = \text{sum}.(xs \downarrow i) : i \rangle$$

2. Derivar el siguiente programa

```
Const N : Int;
Var a : array [0, N) of Num;
    r : Num;
```

```
{N > 1}
```

S

```
{r = ⟨Min i, j : 0 ≤ i < j < N : a.i * a.j⟩}
```

Nota: (1) no se puede usar ∞ ni $-\infty$ en el programa.

Nota: (2) El algoritmo debe recorrer solo una vez el arreglo.

Ayuda: Recordar que el mínimo no distribuye con respecto al producto, a menos que el multiplicador sea positivo.

3. Especificar con pre y poscondición los siguientes problemas:

- a) Hay tres elementos en posiciones distintas de un arreglo cuya suma es igual a una constante C .
- b) Dado un arreglo calcular en otro (de igual tamaño) los promedios de los segmentos finales del arreglo. Esto es, en la posición i -ésima del arreglo resultado debe estar el promedio de los números en las posiciones mayores o iguales a i del arreglo de inicial.

4. (Ejercicio para libres)

Derivar un programa imperativo en base a la siguiente especificación:

```
Const N : Int;
Var a : array [0, N) of Int;
    r : Bool;
```

```
{N ≥ 0}
```

S

```
{r ≡ ⟨∀ i : 0 ≤ i < N : ⟨∑ j : 0 ≤ j < i : a.j⟩ ≤ i + a.i⟩}
```