

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

$$f.xs = \langle \exists i : 0 \leq i < \#xs : \text{prod.}(xs \uparrow i) + xs ! i = 1 \rangle$$

- b) Dar una lista de 5 elementos xs que cumpla $f.xs = \text{True}$. Justificar **usando la especificación**.
 c) Dar otra lista de 5 elementos que cumpla $f.xs = \text{False}$. Justificar **usando la especificación**.

2. Considere el problema de, dado un arreglo, calcular la suma del producto de todos los segmentos del arreglo, especificado de la siguiente manera:

Const $N : \text{Int}, A : \text{array}[0, N) \text{ of } \text{Int};$

Var $r : \text{Int};$

$\{P : N \geq 0\}$

S

$\{Q : r = \langle \sum p, q : 0 \leq p \leq q \leq N : \text{prod.}p.q \rangle$

$\llbracket \text{prod.}p.q = \langle \prod i : p \leq i < q : A.i \rangle \rrbracket\}$

- a) Calcular el resultado para $A = [3, -2, 1]$. Justificar, enumerando todos los elementos del rango.
Ayuda: El resultado es -8 .
 b) Derivar un programa imperativo que resuelva este problema. El programa **debe recorrer una sola vez el arreglo** (sin ciclos anidados).

3. **Especificar con pre y post condición** los siguientes problemas. Declarar constantes y variables.

- a) Dados dos arreglos A y B , de $N \geq 0$ y $M \geq 0$ elementos respectivamente, calcular si existen elementos de A que son menores que todos los de B .

Ejemplo: Con $A = [0, 8, 10]$ y $B = [-1, 7, -8]$ la respuesta es negativa.

- b) Dado un número X y un arreglo A de $N \geq 0$ elementos, calcular el largo del segmento más largo con valores todos mayores o iguales que X .

Ejemplo: Con $X = 6$ y $A = [6, 11, 9, 5, 8]$ la respuesta es 3 por el segmento $[6, 11, 9]$.

4. **(Ejercicio para libres:)** Considere la siguiente especificación:

Const $N : \text{Int}, A : \text{array}[0, N) \text{ of } \text{Int};$

Var $r : \text{Bool};$

$\{P : N \geq 0\}$

S

$\{Q : r = \langle \exists i : 0 \leq i \leq N : \langle \sum j : 0 \leq j < i : A.j \rangle = 2^i \rangle\}$

- a) Calcular el resultado para $A = [34, -10, -16, 12]$. Justificar, enumerando todos los elementos del rango.
 b) Derivar un programa imperativo que resuelva este problema. El programa **debe recorrer una sola vez el arreglo** (sin ciclos anidados).