

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

$$f.xs = \langle \exists as, bs : xs = as ++ bs : prod.as = sum.bs \rangle$$

b) Dar una lista de 4 elementos xs que cumpla $f.xs = True$. Justificar.

c) Dar otra lista de 4 elementos xs que cumpla $f.xs = False$. Justificar.

2. Considere el problema de, dado un arreglo, sumar los productos de todos los pares de elementos que dan > 0 . especificado de la siguiente manera:

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

Var $r : Int;$

{ $P : N \geq 0$ }

S

{ $Q : r = \langle \sum i, j : 0 \leq i < j < N \wedge A.i * A.j > 0 : A.i * A.j \rangle$ }

a) Calcular el resultado para $A = [3, -2, 1, 0, -2]$. 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).

3. **Especificar con pre y post condición** los siguientes problemas. Declarar constantes y variables. **No derivar.**

a) Dados dos arreglos A y B , de $N \geq 0$ y $M \geq 0$ números respectivamente, calcular la suma de los elementos de A que aparecen en B .

b) Dado un arreglo A de $N \geq 0$ elementos, calcular si todos los valores aparecen una única vez en el arreglo.

4. **(Ejercicio para libres:)** Derivar un programa imperativo que calcule si existe algún segmento inicial del arreglo A cuya suma sea -1 , especificado de la siguiente manera:

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

Var $r : Bool;$

{ $P : N \geq 0$ }

S

{ $Q : r = \langle \exists i : 0 \leq i \leq N : \langle \sum j : 0 \leq j < i : A.j \rangle = -1 \rangle$ }

a) Calcular el resultado para $A = [4, -1, -6, 2]$. 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).