

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

$$f.xs = \langle \forall i : 0 \leq i < \#xs : xs!i = 2^i \rangle$$

- b) Dar una lista de 5 elementos xs que cumpla $f.xs = True$. Justificar.
 c) Dar otra lista de 5 elementos que cumpla $f.xs = False$. Justificar.
2. Considere el problema de, dado un arreglo de **números enteros**, calcular la cantidad de pares de elementos cuyo producto da 1, especificado de la siguiente manera:

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

Var $r : Int;$

{ $P : N \geq 0$ }

S

{ $Q : r = \langle N p, q : 0 \leq p < q < N : A.p * A.q = 1 \rangle$ }

- a) Calcular el resultado para $A = [-1, 2, 1, -1, -1]$. 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 $M \geq 0$ elementos respectivamente, calcular cuántos elementos de A son mayores que todos los de B .

Ejemplo: Con $A = [0, 8, 10]$ y $B = [-1, 7, -8]$ la respuesta es 2.

- b) Dado un número X y un arreglo A de $N \geq 0$ elementos, calcular el largo del segmento más corto cuya suma da mayor que X .

Ejemplo: Con $X = 18$ y $A = [6, 11, 9, 6, 8]$ la respuesta es 2 por el segmento $[11, 9]$.

4. **(Ejercicio para libres:)** Derivar un programa imperativo que calcule la cantidad de segmentos iniciales del arreglo A cuyo producto es 8, especificado de la siguiente manera:

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

Var $r : Int;$

{ $P : N \geq 0$ }

S

{ $Q : r = \langle N i : 0 \leq i \leq N : \langle \prod j : 0 \leq j < i : A.j \rangle = 8 \rangle$ }

El programa **debe recorrer una sola vez el arreglo** (sin ciclos anidados).