1. Considere la siguiente especificacíon:

Const N : Int, A : array[0, N) of Int;
Var r : Int;
$\{P : N \geq 1\}$
S
$\{Q : r = \langle N\ i : 0 < i < N : A.i > A.(i - 1)\rangle \}$

a) Explicar en palabras qué problema se calcula de acuerdo a la especificación.

b) Derivar un programa imperativo que satifaga la especificación.
Ayuda: agregar $1 \leq n \leq N$ al invariante.

c) ¿Cuál es el resultado para el arreglo A = [2, 3, 2, 1, 0, 8]?

a) Cantidad de veces en la cual un elemento es mayor estricto que el anterior

b) Derivación:

Paso 1 (Invariante)
Cambio de variable por constante
$INV = r = \langle N\ i : 0 < i < n : A.i > A.(i - 1)\rangle \wedge 1 \leq n \leq N$
$B = n < N$

Luego vale $INV \wedge \neg B \rightarrow Q$ ya que
$INV \wedge \neg B \equiv r = \langle N\ i : 0 < i < n : A.i > A.(i - 1)\rangle \wedge 1 \leq n \leq N \wedge n \geq N$
≡{ lógica
$r = \langle N\ i : 0 < i < n : A.i > A.(i - 1)\rangle \wedge 1 \leq n \leq N \wedge (n > N \vee n = N)$
≡{ distributiva
$r = \langle N\ i : 0 < i < n : A.i > A.(i - 1)\rangle \wedge (1 \leq n \leq N \wedge n > N) \vee (1 \leq n \leq N \wedge n = N)$
≡{ principio de no contradicción
$r = \langle N\ i : 0 < i < n : A.i > A.(i - 1)\rangle \wedge false \vee (1 \leq n \leq N \vee n = N)$
≡{ neutro disyunción, lógica
$r = \langle N\ i : 0 < i < N : A.i > A.(i - 1)\rangle \wedge (1 \leq n \leq N)$
≡{ luego es equivalente a Q
true

Paso 2 (Inicializamos) Proponemos inicializar r,n := E,F
asumimos P
wp.(r,n := E,F).(INV)
≡{ def de wp
$E = \langle N\ i : 0 < i < F : A.i > A.(i - 1)\rangle \wedge 1 \leq F \leq N$
≡{ rango vacío con F=1
$E = \langle N\ i : 0 < i < 1 : A.i > A.(i - 1)\rangle \wedge 1 \leq 1 \leq N$
≡{ logica, hipotesis, rango vacío
$E = 0$
≡{ elegimos convenientemente
true

Paso 3 (cota) Proponemos $t = N - n$ ya que N es constante y podemos hacer crecer n para que la cota decrece en cada paso del ciclo y en caso de ser 0 la guarda se habría hecho falsa.

$INV \wedge B \rightarrow t \geq 0$
asumimos $INV \wedge B$
$N - n \geq 0$
≡{ aritmética
$N \geq 0$
≡{ lógica
$N \geq 1 \vee N = 0$
≡{ hipótesis
true

Paso 4 (cuerpo del ciclo) Proponemos $r,n := E, n+K$
asumimos $INV \wedge B \equiv r = \langle N\, i : 0 < i < n : A.i > A.(i - 1)\rangle \wedge 1 \leq n \leq N \wedge n < N$
$1 \leq n \leq N \wedge n < N$
≡{
$(1 \leq n < N \vee n = N) \wedge n < N$
≡{
$(1 \leq n < N \wedge n < N \vee n < N \wedge n = N)$
≡{ lógica
$1 \leq n < N$
luego
$INV \wedge B \equiv r = \langle N\, i : 0 < i < n : A.i > A.(i - 1)\rangle \wedge 1 \leq n < N$
ahora veamos
$wp.(r,n:=E,n+K).(INV)$
≡{ def de wp
$E = \langle N\, i : 0 < i < n+K : A.i > A.(i - 1)\rangle \wedge 1 \leq n+K \leq N$
≡{ Proponemos aumentar n en 1 con K = 1
$E = \langle N\, i : 0 < i < n+1 : A.i > A.(i - 1)\rangle \wedge 1 \leq n+1 \leq N$
≡{ logica e hipotesis
$E = \langle N\, i : 0 < i < n \vee i = n : A.i > A.(i - 1)\rangle$
≡{ partición de rango
$E = \langle N\, i : 0 < i < n : A.i > A.(i - 1)\rangle + \langle N\, i : i = n : A.i > A.(i - 1)\rangle$
≡{ hipótesis, rango unitario
$E = r + (A.n > A.(n - 1)) \rightarrow 1$
$\neg(A.n > A.(n - 1)) \rightarrow 0$

Luego planteamos un condicional separando por casos
Caso 1 $(A.n > A.(n - 1))$:
$E = r + 1$
≡{ elegimos convenientemente
true
Caso $\neg(A.n > A.(n - 1))$
$E = r$
≡{ elegimos convenientemente
true

Veamos que la cota decrece al ejecutar el cuerpo del ciclo $\{INV \wedge B \wedge t = X\}$ if…fi $\wedge$ $\{t < X\}$
asumimos $INV \wedge B \wedge t = X \equiv r = \langle N\, i : 0 < i < n : A.i > A.(i - 1)\rangle \wedge 1 \leq n < N \wedge N - n = X$

$\underline{A.n > A.(n - 1) \vee \neg(A.n > A.(n - 1))}$
$\wedge$ $\underline{\{A.n > A.(n - 1) \wedge INV \wedge B \wedge t = X\}}$ $r,n := r + 1, n + 1$ $\{t < X\}$
$\wedge$ $\underline{\{\neg(A.n > A.(n - 1))\} \wedge\ INV \wedge B \wedge t = X\}}$ $r,n := r , n + 1$ $\{t < X\}$
≡{ tercero excluido, asumimos
$wp.(r,n := r + 1, n + 1).(t < X)$
$\wedge$ $wp.(r,n := r , n + 1).(t < X)$
≡{ def de wp
$N - (n+1) < X \wedge N - (n+1) < X$
≡{ aritmética
$N - n - 1 < X \wedge N - n - 1 < X$
≡{ hipótesis
$X - 1 < X \wedge X - 1 < X$
≡{ lógica
true

Finalmente

```
Const N : Int, A : array[0, N) of Int;
Var r : Int;
{P : N ≥ 1}
r,n := 0,1
do(n < N) →
   S1
   if(A.n > A.(n − 1)) →
      r,n := r + 1, n + 1
   else(¬(A.n > A.(n − 1))) →
      r,n := r, n+1
   fi
od
   S2
{Q : r = ⟨ N i : 0 < i < N : A.i > A.(i − 1)⟩ }
```

c) ¿Cuál es el resultado para el arreglo A = [2, 3, 2, 1, 0, 8]? es 3

r	n	estado
0	1	S1
1	2	S1
1	3	S1
1	4	S1
2	5	S1
3	6	S1

r = 3

2. Derivar un programa imperativo que calcule si existe alǵun segmento inicial del arreglo A cuya suma sea −1, especificado de la siguiente manera:

Const N : Int, A : array[0, N) of Int;
Var r : Bool;
{P : N ≥ 0}
S
{Q : r = ⟨ ∃ i : 0 ≤ i ≤ N : ⟨ summ j : 0 ≤ j < i : A.j ⟩ = −1 ⟩ }

summ.i ≡ ⟨ summ j : 0 ≤ j < i : A.j ⟩
summ.0 ≡ 0   rango vacío
summ.i+1 ≡ summ.i + A.i

Luego
{Q : r = ⟨ ∃ i : 0 ≤ i ≤ N : summ.i = −1 ⟩ }

Paso 1 (Invariante)
INV = r = ⟨ ∃ i : 0 ≤ i ≤ n : summ.i = −1 ⟩ ^ 0 ≤ n ≤ N
B = n < N
Luego vale INV ^ ¬B → Q

Paso 2 (inicializamos)
asumimos P
wp.(r,n := E,F).(INV)
≡{ def de wp
E = ⟨ ∃ i : 0 ≤ i ≤ F : summ.i = −1 ⟩ ^ 0 ≤ F ≤ N
≡{ F = 0, forzamos rango vacío, logica, hipotesis
E = false
≡{ elegimos convenientemente
true

Paso 3 (cota)
Lo mismo de siempre. Proponemos t = N - n, como N es constante podemos decrecer la cota mientras aumentemos n en cada paso del ciclo, tq INV ^ B → t ≥ 0

Paso 4 (cuerpo del ciclo)
asumimos INV ^ B ≡ r = ⟨ ∃ i : 0 ≤ i ≤ n : summ.i = −1 ⟩ ^ 0 ≤ n < N
wp.(r,n := E,n+1).(INV)
≡{ def de wp
E = ⟨ ∃ i : 0 ≤ i ≤ n+1 : summ.i = −1 ⟩ ^ <u>0 ≤ n+1 ≤ N</u>
≡{ lógica en el rango, lógica e hipótesis
E = ⟨ ∃ i : 0 ≤ i < n+1 v (i = n+1) : summ.i = −1 ⟩
≡{ partición de rango
E = ⟨ ∃ i : 0 ≤ i < n+1 : summ.i = −1 ⟩ v ⟨ ∃ i : (i = n+1) : summ.i = −1 ⟩
≡{ lógica, rango unitario
E = ⟨ ∃ i : 0 ≤ i ≤ n : summ.i = −1 ⟩ v summ.n+1 = −1
≡{ hipótesis, def de summ

$E = r \vee (summ.n + A.n = -1)$
≡{ fortalecemos invariante

$E = r \vee (rsum + A.n = -1)$

$INV' \equiv INV \wedge rsum \equiv \langle\ summ\ j : 0 \leq j < n : A.j\ \rangle$

Inicializamos nuevamente

r,rsum, n := false,0,0 (rango vacío de rsum)

Cuerpo del ciclo nuevamente

asumimos $INV' \wedge B \equiv INV \wedge B \wedge rsum = \langle\ summ\ j : 0 \leq j < n : A.j\ \rangle$
wp.(r,rsum,n:=E,F,n+1).(INV')
≡{ mismos pasos para todo, menos rsum
$F = \langle\ summ\ j : 0 \leq j < n+1 : A.j\ \rangle$
≡{ lógica en el rango
$F = \langle\ summ\ j : 0 \leq j \leq n : A.j\ \rangle$
≡{ lógica, partición de rango
$F = rsum + A.n$
luego el cuerpo del ciclo:
r,rsum,n := $r \vee (rsum + A.n = -1)$, rsum + A.n, n+1

Luego demostramos que la cota decrece al ejecutar el cuerpo del ciclo
$\{INV' \wedge B \wedge t = X\}\ S\ \{t < X\}$

Finalmente

Const N : Int, A : array[0, N) of Int;
Var r : Bool;
Var rsum,r : Int;
$\{P : N \geq 0\}$
r,rsum,n := 0,0,0
<u>do</u>(n < N) →
    r,rsum,n := $r \vee (rsum + A.n = -1)$, rsum + A.n, n+1
<u>od</u>
$\{Q : r = \langle\ \exists\ i : 0 \leq i \leq N : \langle\ summ\ j : 0 \leq j < i : A.j\ \rangle = -1\ \rangle\ \}$

3. Especificar con pre y post condición los siguientes problemas. No derivar.
a) Calcular el máximo y el mínimo de un arreglo no vacío de N enteros A.
b) Calcular si un número entero dado N es un cuadrado perfecto (o sea, el cuadrado de un entero).

Const N : Int, A : array [0,N) of int;
var min, max : Int;
{P: N ≥ 2}
S
{Q: max = ⟨ max i : 0 ≤ i < N : A.i ⟩ ^ min = ⟨ min i : 0 ≤ i < N : A.i ⟩}

Const N : Int;
Var x : Int,;
{P: N ≥ 0}
S
{Q: x^2 = N}