15. Dada la siguiente especificación:
Const M : Int, A : array[0, M) of Int;
Var r : Int;
{P : $M \geq 0$}
S
{Q : $r = \langle N\ p, q : 0 \leq p < q < M : A.p * A.q \geq 0 \rangle$}

Decir en palabras que hace el programa y derivarlo.

Veces en las cuales el producto entre dos elementos diferentes es positivo.
Derivación:

Paso 1 (invariante)
$INV = r = \langle N\ p, q : 0 \leq p < q < m : A.p * A.q \geq 0 \rangle \wedge 0 \leq m \leq M$
B = m < M
luego vale INV ^ ¬B → Q

Paso 2 (inicializamos)
r,n := 0,0 ya que forzamos rango vacío con n=0 y el neutro del conteo es 0

Paso 3 (cota) Proponemos t = M - m ya que N es una constante y podemos hacer crecer n con tal de decrecer la cota en cada paso del ciclo con r,m := E,m+1. Por lo que, una vez que la cota sea 0, la guarda no vale. Es decir, INV ^ B → t ≥ 0.

Paso 4 (cuerpo del ciclo)
asumimos INV ^ B
wp.(r,m := E,m+1).(INV)
≡{ def de wp
$E = \langle N\ p, q : 0 \leq p < q < m+1 : A.p * A.q \geq 0 \rangle \wedge 0 \leq m+1 \leq M$
≡{ lógica e hipótesis
$E = \langle N\ p, q : 0 \leq p < q < m+1 : A.p * A.q \geq 0 \rangle$
≡{ lógica en el rango
$0 \leq p < q < m+1$
≡{
$0 \leq p < q \wedge q < m+1$
≡{
$0 \leq p < q \wedge q \leq m$
≡{
$0 \leq p < q \wedge (q < m \vee q = m)$
≡{ distributiva
$(0 \leq p < q \wedge q < m) \vee (0 \leq p < q \wedge q = m)$
≡{ luego partición de rango y lógica
$E = \langle N\ p, q : (0 \leq p < q < m) : A.p * A.q \geq 0 \rangle + \langle N\ p : 0 \leq p < m : A.p * A.m \geq 0 \rangle$
≡{ hipótesis
$E = r + \langle N\ p : 0 \leq p < m : A.p * A.m \geq 0 \rangle$

Debemos fortalecer dividendo por casos ya que por regla de signos:

$A.p * A.m \geq 0 \equiv (A.p \geq 0 \wedge A.m \geq 0) \vee (A.p < 0 \wedge A.m < 0)$

luego

$E = r + \langle N\ p : 0 \leq p < m : (A.p \geq 0 \wedge A.m \geq 0) \vee (A.p < 0 \wedge A.m < 0)\rangle$

Caso $A.m \geq 0$

$E = r + \langle N\ p : 0 \leq p < m : (A.p \geq 0 \wedge true) \vee (A.p < 0 \wedge false)\rangle$

≡{ lógica

$E = r + \langle N\ p : 0 \leq p < m : A.p \geq 0 \rangle$

luego fortalecemos $rpos = \langle N\ p : 0 \leq p < m : A.p \geq 0 \rangle$

Caso $A.m < 0$

$E = r + \langle N\ p : 0 \leq p < m : (A.p \geq 0 \wedge false) \vee (A.p < 0 \wedge true)\rangle$

≡{ lógica

$E = r + \langle N\ p : 0 \leq p < m : A.p < 0\rangle$

luego fortalecemos $rneg = \langle N\ p : 0 \leq p < m : A.p < 0\rangle$

Por ende, nuestro nuevo invariante

$INV' \equiv INV \wedge rpos = \langle N\ p : 0 \leq p < m : A.p \geq 0 \rangle$
$\qquad\qquad \wedge\ rneg = \langle N\ p : 0 \leq p < m : A.p < 0\rangle$

Inicializamos nuevamente

r,rpos,rneg,m := 0,0,0,0 ya que forzamos rango vacío en todos lados con m = 0

Cuerpo del ciclo nuevamente

asumimos INV'

wp.(r,rpos,rneg,m := E,F,C,m+1).(INV')

≡{ mismos pasos para todo salvo rpos y rneg

$F = \langle N\ p : 0 \leq p < m+1 : A.p \geq 0 \rangle \wedge C = \langle N\ p : 0 \leq p < m+1 : A.p < 0\rangle$

≡{ lógica en el rango

$0 \leq p < m+1 \equiv 0 \leq p \leq m \equiv 0 \leq p < m \vee p = m$

≡{ luego partición de rango, hipótesis y rango unitario

$F = rpos + (A.m \geq 0) \rightarrow 1 \wedge C = rneg + (A.m < 0) \rightarrow 1$
$\qquad \neg(A.m \geq 0) \rightarrow 0 \qquad\qquad \neg(A.m < 0) \rightarrow 0$

Luego planteamos un if para la asignación y finalmente nos queda

```
Const M : Int, A : array[0, M) of Int;
Var r : Int;
{P : M ≥ 0}
r,rpos,rneg,m:= 0,0,0,0
do(m < M) →
   if(A.m ≥ 0) →
      r,rpos,rneg,m := r + rpos, rpos + 1, rneg, m + 1
   else(A.m < 0) →
```

r,rpos,rneg,m := r + rneg, rpos, rneg + 1, m + 1

<u>od</u>

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