

Examen Junio

[1pto.]

1. Expresa los siguientes enunciados en L.P.O.:

a) En el vector $A[1..n]$ hay un único elemento primo.

$$\begin{aligned} EsPrimo(x) &\equiv \neg \exists i (1 < i < x \wedge x \bmod i = 0) \\ \& j (1 \leq j \leq n \wedge EsPrimo(A[j])) &= 1 \end{aligned}$$

b) Todos los números pares del vector $A[1..n]$ están en la sección $A[i..j]$. (Nota: eso no quiere decir que todos los elementos de $A[i..j]$ sean pares)

$$\neg \exists k (1 \leq k < i \wedge A[k] \bmod 2 = 0) \wedge \neg \exists k (j < k \leq n \wedge A[k] \bmod 2 = 0)$$

[1 pto.]

2. Di cuáles de estas implicaciones lógicas son ciertas y cuáles falsas. **Razona las falsas o da un contraejemplo:**

a) $\text{suma} = \sum_{i=1}^k (i + a) \rightarrow \text{suma} = a * k + \sum_{i=1}^k i$

Cierto.

b) $\exists i (1 \leq i \leq k \wedge esPrimo(i)) \rightarrow \& i (1 \leq i \leq n \wedge esPrimo(i)) > 1$

Falso. Existe alguno así que el cardinal será ≥ 1

c) $x \geq y \rightarrow \text{def}(x < y)$

Cierto. $x < y$ no puede dar error, así que $\text{def}(x < y)$ será siempre cierto.

d) $1 \leq i \leq n \rightarrow \text{def}(s = A[i] + A[i+1])$

Falso. $A[i+1]$ sí puede dar error, sólo sabemos que i está dentro del vector.

e) $1 < i < n \rightarrow 1 \leq i \leq n$

Cierto.

[1.5 ptos.]

3. Descompón la representación esquemática del programa siguiente:

```
/* $\Phi$ */
while( $B_1$ )
{
   $A_1$ ;
   $A_2$ ;
}
if( $B_2$ ) $A_3$ ;
else  $A_4$ ;
/* $\Psi$ */
```

(A)/* Φ */ while(B_1) A_1 ; A_2 ; /* Φ_1 */
 (A.1) $\Phi \rightarrow INV$
 (A.2) $\Phi \wedge \neg B_1 \rightarrow \Phi_1$
 (A.3) $INV \rightarrow \text{def}(B_1)$
 (A.4)/* $INV \wedge B_1$ */ A_1 ; A_2 ; /* INV */
 (A.4.1)/* $INV \wedge B_1$ */ A_1 ; /* Φ_2 */
 (A.4.2)/* Φ_2 */ A_2 ; /* INV */
 (B)/* Φ_1 */ if(B_2) A_3 ; else A_4 ; /* Ψ */
 (B.1) $\Phi_1 \rightarrow \text{def}(B_2)$
 (B.2)/* $\Phi \wedge B_2$ */ A_3 ; /* Ψ */
 (B.3)/* $\Phi \wedge \neg B_2$ */ A_4 ; /* Ψ */

[1.25 pts.]

4. Deduce el **invariante** y la **expresión cota E** del programa siguiente:

```
/*i=0 ∧ numNeg=0*/
while(i < n)
{
    i=i+1;
    if(A[i] < 0)
    {
        numNeg=numNeg+1;
        A[i]=-A[i];
    }
}
/* numNeg = ∑ j (1 ≤ j ≤ n ∧ A[j] < 0) ∧ ∀ j (1 ≤ j ≤ n → A[j] ≥ 0) */
```

/* INV */ $\equiv$ /* $0 \leq i \leq n \wedge \text{numNeg} = \sum j (1 \leq j \leq n \wedge A[j] < 0) \wedge \forall j (1 \leq j \leq n \rightarrow A[j] \geq 0)$ */
 $E \equiv n - i$

[1.75 pts.]

5. Verifica el siguiente programa:

```
/* z = x * y */
if(y > 0)
{
    z = z - x;
    y = y - 1;
}
/* z = x * y */
```

[2 pts.]

6. Verifica la **corrección parcial** de la siguiente iteración:

```
/*i = a - 1 ∧ serie = 0*/
while(i < b)
{
    i=i+1;
    serie=serie+i*i;
}
/* serie = ∑_{j=a}^b j^2 */
```

$$/* INV */ \equiv /* a-1 \leq i \leq b \wedge serie = \sum_{j=a}^i j^2 */$$

[1.5 ptos.]

7. Verifica la **corrección total** del siguiente programa recursivo:

```
function foo(int [] A, int i) return int res
/* 1 ≤ i ≤ n */
if (i == n) res = A[n];
else
{
    res = foo(A, i + 1);
    res = res * A[i];
}
/* res = ∏_{j=i}^n A[j] */
```