

Examen – 2.a convocatoria

[1pto.]

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

a) El vector $A[1..n]$ está ordenado de manera creciente a partir de la posición k .

b) El vector $B[1..n]$ contiene múltiplos de los elementos de $A[1..n]$

Ejemplo: $A[1..n] = [2, 3, 2, 5, 7]$
 $B[1..n] = [4, 21, 2, 45, 21]$

[1 pto.]

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

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

b) $x \neq 0 \rightarrow def(x/x == 0)$

c) $1 < i + 1 < n \rightarrow 1 \leq i \leq n$

d) $\forall i (1 \leq i \leq n \rightarrow A[i] < i^2) \rightarrow A[i] < i^2$

e) $def(x/y = z) \rightarrow x = z * y$

[1 ptos.]

3. Descompón la representación esquemática del programa siguiente:

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

[1 ptos.]

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

```
/*  $i = a \leq b \wedge numM2 = 0 \wedge numM3 = 0$  */
while ( $i \leq b$ )
{
    if ( $i \% 2 == 0$ )  $numM2 = numM2 + 1$  ;

    if ( $i \% 3 == 0$ )  $numM3 = numM3 + 1$  ;

     $i = i + 1$  ;
}
/*  $numM2 = \aleph j (a \leq j \leq b \wedge j \bmod 2 = 0) \wedge numM3 = \aleph j (a \leq j \leq b \wedge j \bmod 3 = 0)$  */
```

[2 ptos.]

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

```
/*  $x = a \wedge y = b \geq 1$  */
while( $y > 1$ )
{
     $x = x * a$ ;
     $y = y - 1$ ;
}
/*  $x = a^b$  */
```

$/* INV */ \equiv /* 1 \leq y \leq b \wedge x = a^{(b-y+1)} */$

[2 ptos.]

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

```
function foo(int [] A, int i, int j) return int res
/*  $1 \leq i \leq j \leq n$  */
if( $i == j$ )  $res = A[i] + i * i$ ;
else
{
     $res = foo(A, i, j-1)$ ;
     $res = res + A[j] + j * j$ ;
}
/*  $res = \sum_{k=i}^j A[k] + \sum_{k=i}^j k^2$  */
```

[2 ptos.]

7. Se quiere programar una función que sirva para calcular potencias. **Deriva** formalmente una función recursiva que calcule x^y a partir de dos números enteros x e y .