Departamento de L.S.I.
Modelo de examen
Metodología de la Programación
2008-2009

Escuela Universitaria de Ingeniería Vitoria-Gasteiz
Ingeniaritzako Unibertsitate Eskola Vitoria-Gasteiz

[1 pto.]

1. Expresa los siguientes enunciados en **LPO**:

   a) Todos los elementos de A[1..n] son números primos.

   b) El producto de los números mayores que *i* y menores que *j* tiene *z* divisores.

[1 pto.]

2. Decide cuáles de las siguientes afirmaciones son ciertas y razona aquellas que no los sean:

   a) $\text{num\_ceros} = \aleph\, i\,(1 \le i < k \wedge A[i] = 0) \wedge A[k] = 0 \rightarrow \text{num\_ceros} = \aleph\, i\,(1 \le i < k \wedge A[i] = 0)$

   b) $x \bmod i = 0 \rightarrow \exists i\,(1 < i < x \wedge x \bmod i = 0)$

   c) $1 \le i \le n \wedge A[i]\,! = 0 \rightarrow \text{def}\,(A[i] == 0)$

   d) $x = a/b \rightarrow x - 1 = (a - b)/b$

   e) $\forall i\,(1 < i < x - 1 \rightarrow x \bmod i\,! = 0) \wedge x \bmod (x-1)\,! = 0 \rightarrow \forall\,(1 < i < x \rightarrow x \bmod i\,! = 0)$

[1.5 ptos.]

3. Descompón la siguiente representación esquemática de un programa:

```
/* Φ */

while(B1)
{
if (B2) A1;
else A2;

A3;
}

/* Ψ */
```

[1.25 ptos.]

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

```
/* i=n ∧ num_pares=0 ∧ num_impares=0 */
while (i>0)
{
if (A[i] 2==0) num_pares=num_pares+1;
else num_impares=num_impares+1;

i=i-1;
}
/* num_pares=ℵ i(1≤i≤n ∧ A[i] mod 2=0) ∧
num_impares=ℵ i(1≤i≤n ∧ A[i] mod 2!=0) */
```

[1.75 ptos.]

5. **Verifica la corrección total** del siguiente programa:

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

[2 ptos.]

6. **Verifica la corrección parcial** del siguiente programa:

```
/* i=0 ∧ s=0 */
while (i<x)
{
i=i+1;
s=s+A[i];
s=s+1;
}
/* s=x+∑_{j=1}^{x} A[j] */
```

Sabemos que el invariante es el siguiente:

$$/* \; INV \; */ \equiv /* \; 0 \le i \le x \wedge s = j + \sum_{j=1}^{i} A[j] \; */$$

[1.5 ptos.]

7. **Verifica la corrección total** de la siguiente función recursiva:

```
function SumaCuadrados(int n) return int suma
/* n>0 */
{
if (n==1) suma=1;
else
}
suma=SumaCuadrados(n-1);
suma=suma+n*n;
}
/* suma=∑_{i=1}^{n} i^2 */
```