

Aclaraciones: Se permite tener UNA hoja A4 con anotaciones durante el parcial. Cualquier decisión de interpretación que se tome debe ser aclarada y justificada. Para aprobar se requieren al menos 60 puntos. Entregar cada ejercicio en una hoja separada, numerada y que incluya el nro. de orden.

Ejercicio 1. [25 puntos] Demostrar formalmente la corrección del programa con respecto a su especificación usando la precondition más débil.

Especificación

```
proc procedimientoRaro (inout a: Z, inout b: Z) {
  Pre {a = A0 ∧ b = B0}
  Post {a = B0 ∧ ((b = B0 ∧ B0 ≥ 0) ∨ (b = -B0 ∧ B0 < 0))}
}
```

Implementación en SmallLang

```
a := b;
if (b < 0)
  b := -b
else
  skip
endif
```

Ejercicio 2. [25 puntos] Sea el siguiente ciclo con su correspondiente precondition y postcondición:

$P_c : \{i = 0 \wedge res = 0\}$

```
while (i < length(s)) do
  if (s[i] == i)
    res := res + i
  else
    skip
  endif;
  i := i + 1
endwhile
```

$Q_c : \{res = \sum_{j=0}^{i-1} (if s[j] = j then j else 0 fi)\}$

- [15 puntos] Proponer un invariante que satisfaga las hipótesis del teorema del invariante (no se pide una demostración).
- [10 puntos] Proponer una función variante y demostrar que el ciclo termina.

Ejercicio 3. [25 puntos] Escribir un programa en C++ que dada una secuencia no vacía de enteros positivos (mayores que 0), devuelva la primera "parte izquierda" tal que la suma de sus elementos sea mayor a los de la "parte derecha". Asumir que la suma de los elementos de la secuencia vacía es 0. Algunos ejemplos de ejecución serían los siguientes:

- $\langle 5 \rangle \mapsto \langle 5 \rangle$
- $\langle 1, 7, 5 \rangle \mapsto \langle 1, 7 \rangle$
- $\langle 3, 1, 1 \rangle \mapsto \langle 3 \rangle$
- $\langle 1, 2, 5, 8, 17 \rangle \mapsto \langle 1, 2, 5, 8, 17 \rangle$

Ejercicio 4. [25 puntos] Una matriz se dice de *filas batidas* si todas las filas (i) contienen los mismos valores aunque no necesariamente en el mismo orden y (ii) todos los valores aparecen la misma cantidad de veces. La primera de las matrices que se exhiben a continuación es de *filas batidas* y las restantes dos no lo son. Se pide escribir un programa en C++ que dada una matriz de $n \times m$ determine si es o no de *filas batidas*.

```
1 2 6
6 1 2
2 6 1
1 2 6
```

```
1 1 3
3 1 1
3 3 1
```

```
1 1 3
2 1 1
3 1 1
```