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] Dado el siguiente programa y los siguientes predicados:

```
int multi(int a, int b) {
    int res = a * b;
    return res;
}
```

- $Q_1 : res = a * b$
- $Q_2 : res = a^2$
- $Q_3 : res \geq 0$

a) ¿Cuáles de los predicados pueden ser la postcondición de la especificación para el programa si la precondición es $a \geq 0 \longleftrightarrow b \geq 0$?
b) ¿Cuáles de los predicados pueden ser la postcondición de la especificación para el programa si la precondición es $True$?
c) ¿Cuáles de los predicados pueden ser la postcondición de la especificación para el programa si la precondición es $a = b$? Asumiendo esta precondición, ¿cómo es la relación de fuerza entre las postcondiciones válidas?
d) ¿Cuál es la relación de fuerza entre las 3 precondiciones planteadas?

Ejercicio 2. [25 puntos]

Dada una secuencia de enteros s y un natural n, llamaremos acumulador de orden n a un elemento en la secuencia cuyo valor es exactamente igual a la suma de los n valores anteriores al acumulador. Por ejemplo, dados $n = 2$ y $s = \langle 3, 4, 7, 5, 2, 1, 4, 5 \rangle$, s tiene dos acumuladores de orden 2 en las posiciones 2 y 7.

Especificar el problema de, dada una secuencia de enteros s y un natural n, modificar la secuencia de forma tal que queden las posiciones de los acumuladores de orden n. Además, el procedimiento deberá devolver un booleano que indique si hay al menos dos acumuladores de distinto valor en la secuencia.

Por ejemplo, dados $n = 2$ y $s = \langle 3, 4, 7, 5, 2, 1, 4, 5 \rangle$, la secuencia debería transformarse en $s = \langle 2, 7 \rangle$ y se debería devolver True (ya que los acumuladores son 7 y 5, con distinto valor). Para $n = 2$ y $s = \langle 1, 6, 7, 2, 5, 7, 4, 3, 7 \rangle$, la secuencia debería transformarse en $s = \langle 2, 5, 8 \rangle$ y se debería devolver False (ya que todos los acumuladores tienen el mismo valor).

Ejercicio 3. [25 puntos] Dada la siguiente especificación:

proc hacerCosaRara (inout l: $seq\langle\mathbb{Z}\rangle$) {
  Pre $\{|l| \ mod \ 2 = 0 \wedge l = L_0\}$
  Post $\{|l| = |L_0| \wedge_L (\forall i : \mathbb{Z})\big((0 \leq i < |l| \wedge i \ mod \ 2 = 0) \rightarrow_L l[i] = 0 \big) \wedge (\forall i : \mathbb{Z})\big((0 \leq i < |l| \wedge i \ mod \ 2 = 1) \rightarrow_L l[i] = L_0[i-1] \big)\}$
}

Implementar un programa en C++ que cumpla con dicha especificación y que contenga un ciclo `while` cuyo invariante I sea:

$$\begin{aligned} & |l| = |L_0| \wedge -1 \leq i \leq |l| - 1 \wedge_L i \ mod \ 2 = 1 \\ & \wedge (\forall j : \mathbb{Z})\big(0 \leq j \leq i \rightarrow_L l[j] = L_0[j] \big) \\ & \wedge (\forall j : \mathbb{Z})\big((i < j < |l| \wedge j \ mod \ 2 = 0) \rightarrow_L l[j] = 0 \big) \\ & \wedge (\forall j : \mathbb{Z})\big((i < j < |l| \wedge j \ mod \ 2 = 1) \rightarrow_L l[j] = L_0[j-1] \big) \end{aligned}$$

Ejercicio 4. [25 puntos] Dada la siguiente especificación junto con el siguiente programa:

proc restaExtraña (in x: $\mathbb{Z}$, in y: $\mathbb{Z}$, out res: $\mathbb{Z}$) {
  Pre {True}
  Post {if $0 \leq x < y$ then $res = y - x$ else $res = 0$ fi}
}

```
  int restaExtrana(int x, int y) {
L1    int dif = 0;
L2    if (x > 0) {
L3        while (x <y) {
L4            dif++;
L5            x++;
          }
      }
L6    return dif;
  }
```

a) Describir el diagrama de control de flujo (control-flow graph) del programa.
b) ¿Es posible escribir para este programa un test suite que cubra todas las líneas pero no cubra todas las decisiones? En caso afirmativo, describirlo; en caso negativo, justificarlo.
c) ¿Es posible escribir para este programa un test suite que cubra todas decisiones pero no encuentre el defecto en el código? En caso afirmativo, describirlo; en caso negativo, justificarlo.