

IMPORTANTE:

- Para aprobar es necesario **sumar 60 puntos**.
- Las condiciones de entrega se encuentran en el cuestionario del campus.

Ejercicio 1. [35 puntos]

Sea el siguiente programa con su correspondiente precondition y postcondición:

$$P_c : \{ |s| > 1 \wedge 0 < n < |s| \wedge i = |s| \wedge s = S_0 \}$$

```

while ( i > n ) do
    i = i-1;
    s[i] = s[i-1]
endwhile
    
```

$$Q_c : \{ |s| = |S_0| \wedge_L ((\forall k : \mathbb{Z})(n \leq k < |s| \rightarrow_L s[k] = S_0[k-1]) \wedge (\forall k : \mathbb{Z})(0 \leq k < n \rightarrow_L s[k] = S_0[k])) \}$$

Proponer un invariante para el ciclo y demostrar que cumple los siguientes puntos del teorema del invariante:

- $(I \wedge \neg B) \Rightarrow Q_c$
- $\{I \wedge B\}$ Cuerpo del Ciclo $\{I\}$.

Ejercicio 2. [35 puntos]

En un tablero de ajedrez de tamaño $N \times N$, con $N > 4$, se mueve a los saltos un caballo. El caballo de ajedrez se mueve una casilla en una dirección y otra casilla en diagonal alejándose de la casilla inicial (ver Fig. 1 y 2), sin salirse del tablero. El tablero se representa con una secuencia de secuencias de enteros donde 0 representa una casilla libre y 1 representa una casilla ocupada, sin importar qué pieza específica contiene. Las casillas se representan como pares $\mathbb{Z} \times \mathbb{Z}$ (*fila, columna*).

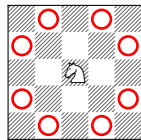


Fig. 1: Saltos posibles desde (2,2)

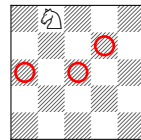


Fig. 2: Saltos posibles desde (0,1)

a) Asumiendo dado el predicado $esSaltoVálido(N : \mathbb{Z}, a : \mathbb{Z} \times \mathbb{Z}, b : \mathbb{Z} \times \mathbb{Z})$, que será *Verdadero* si y sólo si es válido hacer un salto de caballo desde la casilla a hacia la casilla b en un tablero de dimensión N , especificar el problema **saltarYComer**, que toma un tablero y una secuencia de al menos dos casillas, donde la primera casilla es la casilla inicial del caballo (que debe estar ocupada), y las casillas restantes forman un *camino válido* de saltos de caballo en el tablero dado. El procedimiento modifica el tablero realizando los movimientos, *comiendo* cualquier pieza encontrada en casillas ocupadas visitadas, y dejando el caballo en la casilla final. Además, devuelve la cantidad de piezas comidas en la secuencia de saltos realizada.

b) Escribir el *pred* $esSaltoVálido(N : \mathbb{Z}, a : \mathbb{Z} \times \mathbb{Z}, b : \mathbb{Z} \times \mathbb{Z})$ usado en el ítem anterior.

Ejemplo: dado el tablero inicial de la Fig. 3, cuya representación será:

$$t = \langle \langle 0, 1, 0, 0, 0 \rangle \langle 0, 0, 1, 0, 0 \rangle \langle 1, 0, 1, 0, 0 \rangle \langle 0, 0, 0, 0, 0 \rangle \langle 0, 0, 1, 0, 0 \rangle \rangle$$

y la secuencia de movimientos $\langle (2, 2), (0, 1), (2, 0), (4, 1), (2, 2), (3, 4) \rangle$, una implementación correcta de **saltarYComer** debe dejar el tablero en el estado

$$t = \langle \langle 0, 0, 0, 0, 0 \rangle \langle 0, 0, 1, 0, 0 \rangle \langle 0, 0, 0, 0, 0 \rangle \langle 0, 0, 0, 0, 1 \rangle \langle 0, 0, 1, 0, 0 \rangle \rangle$$

que se puede observar gráficamente en la Fig. 4. Además, en este caso, debe informar que fueron comidas **2 piezas**.

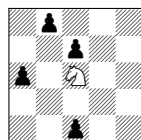


Fig. 3: Tablero inicial t

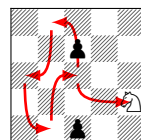


Fig. 4: Posición final del tablero t

Ejercicio 3. [30 puntos] Dados el siguiente programa S en SmallLang y la siguiente especificación:

<pre> if (pisar = true) then res := s[j-1]; s[j-1] := 'z' else skip endif </pre>	<pre> proc colaDeZetas (inout s: seq(Char), in j: Z, in pisar: Bool, out res: Char) { Pre {s = S₀ ∧ 1 ≤ j < s ∧ (∀k : Z)(j ≤ k < s →_L s[k] = 'z')} Post { s = S₀ ∧_L (pisar = true → (res = S₀[j - 1] ∧ (∀k : Z)(j - 1 ≤ k < s →_L s[k] = 'z')))} } </pre>
--	--

- a) Calcular la precondition más débil del programa S con respecto a la postcondición de la especificación: $wp(S; Post)$.
- b) Demostrar que el programa es correcto respecto a la especificación. En caso de no ser posible, justificar detalladamente por qué el programa no es correcto y qué parte de la demostración falla.