



1. Debugging + Invariantes

Importar el proyecto de la clase. En CLion seleccionar File → New CMake Project From Sources ... y seleccionar la carpeta con el nombre *Labo05* que se encuentra en el archivo *Labo05.zip*.

En onlinedb copiar la estructura de los archivos main.cpp, ejercicios.cpp y ejercicios.h.

Ejercicio 1. Los siguientes ejercicios contienen un código con errores junto con casos de prueba. Para los casos que fallan, utilizar el debugger para encontrar los posibles errores y arreglar el código.

- **Está Ordenada:** dada una secuencia de enteros indica si está ordenada ascendente o descendentemente.
- **Pertenece:** dada una secuencia de enteros y un elemento, indica si este pertenece o no a la secuencia.
- **Fibonacci:** calcula el K-esimo número de la serie de fibonacci.

Ejercicio 2. Los siguientes ejercicios contienen un código y un predicado I. Corregir el predicado para que sea un invariante del ciclo. Para cada uno, escribir Pc y Qc.

Pueden utilizar el debugger para ver el valor de las variables tanto al comenzar las iteraciones como al finalizar la ejecución del cuerpo del ciclo.

- **Suma doble:** Dada una secuencia de enteros s , calcular la sumatoria del doble de los elementos que son positivos y pares.

$$I \equiv 1 \leq i < |s| \wedge_L res = \sum_{k=0}^i \text{if } (s[k] \bmod 2 = 0 \wedge s[k] >= 0) \text{ then } s[k] * 2 \text{ else } 0 \text{ fi}$$

- **Existe Pico:** Una secuencia tiene picos si en alguna posición el elemento es mayor tanto del anterior como del siguiente. Decidir si una secuencia dada (de al menos tres elementos) tiene picos.

$$I \equiv 0 \leq i \leq |s| \wedge_L res = true \leftrightarrow (\exists k : \mathbb{Z}) 1 \leq k \leq i \wedge_L s[k] > s[k-1] \wedge s[k] > s[k+1]$$

- **Todos Impares:** Dada una secuencia de enteros positivos s (no vacía), chequea si todos los elementos de la secuencia son impares.

$$I \equiv 0 \leq i \leq |s| \wedge_L res = true \leftrightarrow (\forall k : \mathbb{Z}) (0 \leq k < i \rightarrow_L s[k] \bmod 2 = 1)$$

Ejercicio 3. Dado el Invariante I y un código, ver que I se cumple al principio de cada iteración y al finalizar la ejecución del cuerpo del ciclo para $n = 7$. ¿Qué pasa con $n = 8$? Corregir el código para que I sea un Invariante del ciclo.

- **Suma primos:** Calcular la suma de todos los números primos positivos mayores a 2 hasta n (inclusive).

$$I \equiv 1 \leq i < n \wedge i \bmod 2 = 1 \wedge_L suma = \sum_{k=i+2}^n \text{if } esPrimo(k) \text{ then } k \text{ else } 0 \text{ fi}$$

Ejercicio 4. Resolver los siguientes problemas respetando, para el ciclo principal del programa, el invariante dado.

- **Mínimo de una subsecuencia:** Devuelve el índice del mínimo valor de una subsecuencia.

```

proc indiceMinimoSubsec (in s:seq⟨ℤ⟩, in i,j:ℤ, out res:ℤ) {
  Pre {|s| > 0 ∧ 0 ≤ i, j < |s| ∧ i ≤ j}
  Post {i ≤ res ≤ j ∧ (∀k : ℤ) i ≤ k ≤ j →L s[k] ≥ s[res]}
}

```

$$I \equiv i - 1 \leq l \leq j \wedge i \leq res \leq j \wedge (\forall k : \mathbb{Z}) l < k \leq j \rightarrow_L s[k] \geq s[res]$$

- **Sumatoria de los elementos de una secuencia:** Calcula la suma de todos los elementos de una secuencia s con cantidad de elementos impares.

$$I \equiv 1 \leq i \leq \lfloor |s| \text{ div } 2 \rfloor + 1 \wedge_L suma = s[\lfloor |s| \text{ div } 2 \rfloor] + \sum_{k=1}^{i-1} s[\lfloor |s| \text{ div } 2 \rfloor - k] + s[\lfloor |s| \text{ div } 2 \rfloor + k]$$

- ¿Qué habría que cambiar en el invariante si no hubiera ninguna restricción en la cantidad de elementos de s ?

- **Ordenar (Opcional):** Ordena ascendentemente una secuencia (no vacía) de enteros.

$$I \equiv 0 \leq i \leq |s| \wedge_L |s| = |s_0| \wedge mismos(s, s_0) \wedge_L ordenada(subseq(s, 0, i)) \wedge$$

$$(\forall k : \mathbb{Z}) 0 \leq k < |s| \wedge i > 0 \rightarrow_L ((k < i \wedge s[k] \leq s[i-1]) \vee (k \geq i \wedge s[k] \geq s[i-1]))$$

```

pred ordenada (s:seq⟨ℤ⟩) {
  (∀i : ℤ) 0 ≤ i < |s| - 1 →L s[i] ≤ s[i+1]
}
pred mismos (s:seq⟨ℤ⟩, s0 : seq⟨ℤ⟩) {
  (∀i : ℤ) 0 ≤ i < |s| →L #apariciones(s, i) = #apariciones(s0, i)
}

```

Ayuda: Utilizar la función *indiceMinimoSubsec*.