

Práctica 3

Práctica 3: Sincronización I

1. Considerar el siguiente pseudocódigo para dos threads que ejecutan concurrentemente con una variable compartida x .
¿Cual es el valor final imprimido para x ? ¿Depende del orden de ejecución? En ese caso, mostrar todos los *caminos* (ejecuciones) que terminar por dar outputs distintos. ¿Podría decirse que hay race conditions? ¿Y data races?

Thread A	Thread B
x = 5	x = 7
print(x)	

2. Repetir el mismo analisis del ejercicio anterior pero para el siguiente código y asumiendo que x es inicializada en 0.

Thread A	Thread B
x = x + 1	x = x + 1

3. Considerar el siguiente pseudocódigo para dos threads que ejecutan concurrentemente con una variable compartida x inicializada en 0.
Determinar el valor máximo y mínimo que puede obtener x y dar una ejecución que los realice. ¿Es posible que el valor final de x sea 0? Si es así, justificar.

for(int i = 0; i<2; i++)	for(int i = 0; i<2; i++)
x = x + 2	x = x - 1

4. petir el mismo analisis del ejercicio anterior pero para el siguiente caso. El caso del mínimo es menos evidente que el del máximo.

for(int i = 0; i<100; i++)	for(int i = 0; i<100; i++)
x = x + 1	x = x + 1

5. Resolver los niveles de las secciones *Tutorial* y *Unsynchronized Code* de la página The Deadlock Empiere
6. ¿Qué significa que una instrucción sea atómica a nivel hardware? Considerar todos los ejercicios anteriores y determinar si, en caso de que las operaciones de suma o resta fuesen atómicas, se solucionarían los problemas encontrados previamente.