Exercise 1: Reader-Writer Problem Implementation

Objective: Implement a solution to the classic Reader-Writer problem using pthreads and POSIX semaphores in Linux.

Description:

- Create a shared resource (a file) that multiple reader and writer threads will access.
- Implement the following conditions:
    - Multiple readers can read simultaneously.
    - Writers have exclusive access; no other reader or writer can access the resource while a writer is writing.
- Use semaphores to manage synchronization between the reader and writer threads.
- Test the program by creating multiple threads (e.g., 5 readers and 2 writers) and ensure no race conditions occur.
- The number of readers and writers should be configurable via variables.

Exercise 2: Producer-Consumer Problem with Bounded Buffer

Objective: Solve the Producer-Consumer problem using pthreads and a bounded buffer to explore synchronization with mutexes and condition variables.

Description:

- Implement a bounded buffer using a fixed-size array.
- The size of the buffer should be configurable from a preprocessor macro
- Use two types of threads:
    - **Producer Threads:** Insert items into the buffer.
    - **Consumer Threads:** Remove items from the buffer.
- Implement proper synchronization using mutexes and condition variables to ensure:
    - Producers block if the buffer is full.
    - Consumers block if the buffer is empty.
- Verify the solution by running multiple producer and consumer threads, with each printing the items they produce or consume.