Condition Variables with Predicate Solutions

Multiple Threads

- Modify the reader task to display its thread ID before and after calling wait()
- Modify the main thread so that it starts three reader threads
- Modify the writer task so that it
  - Calls notify_one() once
  - Calls notify_one() three times
  - Calls notify_all() once

Multiple Threads

- Explain the results
  - When the writer thread calls notify_all(), the condition variable will wake up all three reader threads. In each reader, wait() returns, the mutex is locked and the thread can display the modified value before it exits
  - When the writer thread calls notify_one() three times, the condition variable will also wake up all three reader threads
  - Calling notify_one() once will cause one reader thread to be woken up. The other two readers will continue to sleep. If no further notifications are sent to the condition variable, the program will be blocked indefinitely
  - In all three cases, the choice of which thread to wake up, and the order in which to wake them up, is made by the scheduler. Different executions may result in different output