**The Bounded Buffer Problem Using Semaphores:**

A computer screen shot of a program

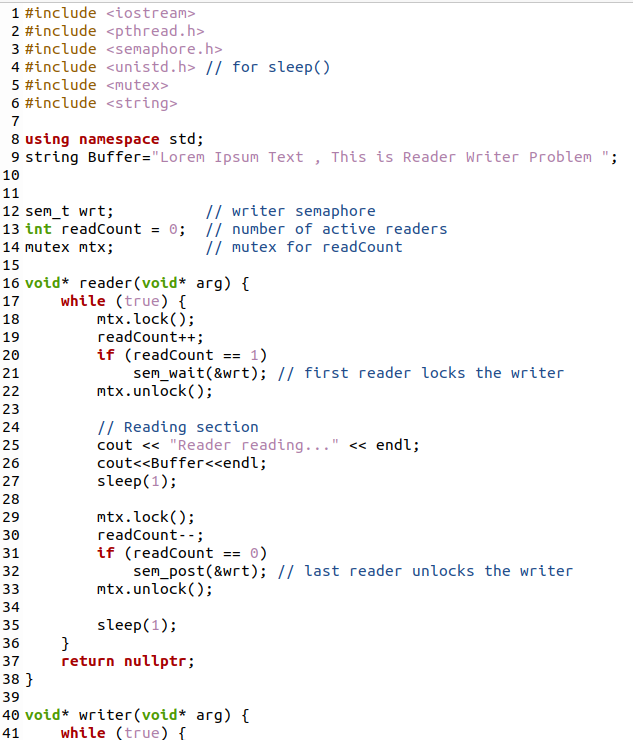
AI-generated content may be incorrect.A computer code with text

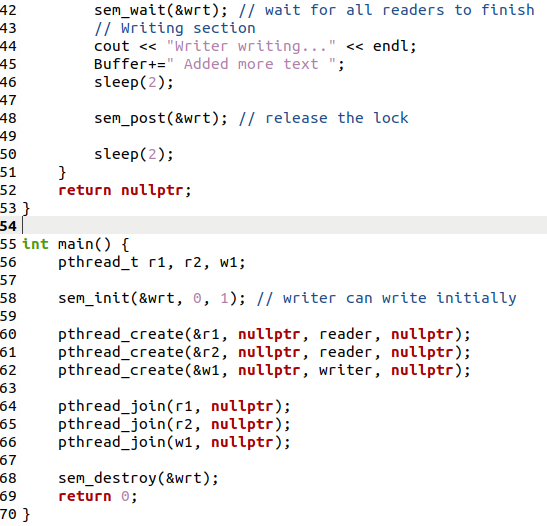
AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**The Reader Writer Problem Using Semaphores:**





A screenshot of a computer program

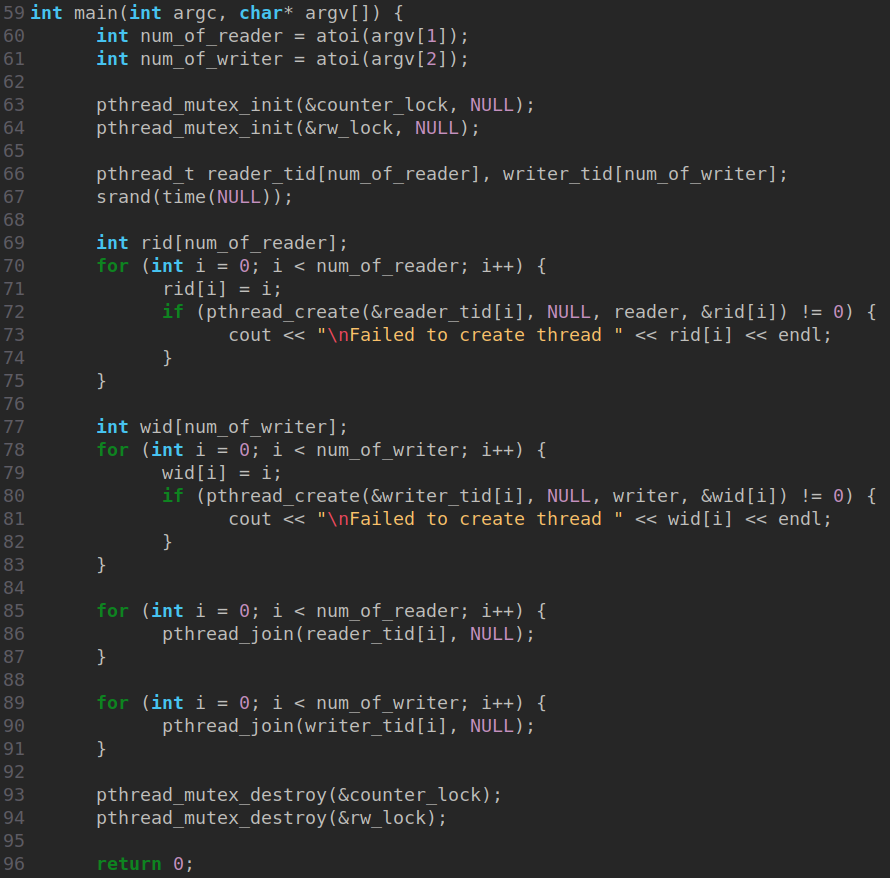
AI-generated content may be incorrect.

A screen shot of a computer program

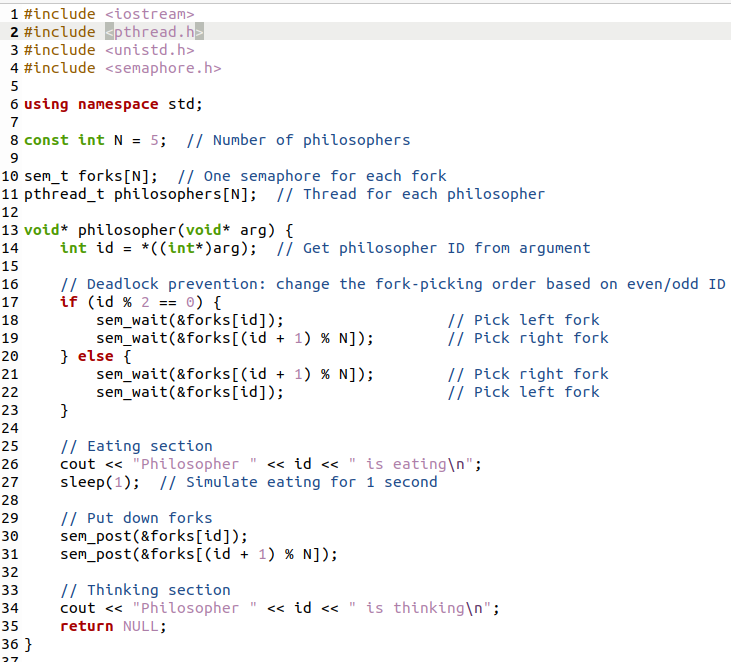
AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.



**The Dining Philosopher Problem:**

****

A computer screen shot of text

AI-generated content may be incorrect.

A computer screen shot of a computer screen

AI-generated content may be incorrect.

Several approaches can solve this problem and prevent deadlock:

1. **Limit the number of philosophers at the table:** Allow at most *N-1* philosophers to pick up their first chopstick simultaneously. This ensures that at least one philosopher can always pick up both chopsticks and eat.
2. **Asymmetric solution:** Make one philosopher (or a subset) pick up chopsticks in the reverse order (e.g., right first, then left). This breaks the circular wait condition.
3. **Arbitrator/Waiter:** Introduce a central "waiter" or monitor. A philosopher must ask permission from the waiter to pick up chopsticks. The waiter grants permission only if both chopsticks are available.
4. **Resource Hierarchy:** Assign an order to the chopsticks (e.g., Chopstick 1, Chopstick 2, ... Chopstick N). Philosophers must always pick up the lower-numbered chopstick first, then the higher-numbered one. This prevents circular waiting.

