CS5103: Assignment 12 // Date: 13-11-24

Title: "The Chronicles of Syncville"

In the village of Syncville, three guardians protect the town's peace and prevent chaos from erupting. These guardians - Mutex, Semaphore, and Condition Variable - each have a unique question to solve. To test your programming skills, you must help them maintain order by writing C code for synchronization problems of varying difficulty.

Question 1: The Bank of Syncville

Story: The townsfolk of Syncville regularly visit the Bank of Syncville to deposit and withdraw money. However, only one person can access the vault at a time to avoid mishandling of funds. Mutex, the first guardian, needs help implementing a lock system to ensure only one villager can access the vault at a time. Can you help Mutex write a program that enforces this rule?

Task: Write a C program that simulates multiple threads trying to access the vault. Use a mutex lock to ensure that only one thread can access it at any given time.

Example Input:

- 5 villagers try to access the vault to perform their transactions.

Expected Output:

Villager 1 is accessing the vault.
Villager 1 has finished and left the vault.
Villager 2 is accessing the vault.
Villager 2 has finished and left the vault.
Villager 3 is accessing the vault.
Villager 3 has finished and left the vault.
Villager 4 is accessing the vault.
Villager 4 has finished and left the vault.
Villager 5 is accessing the vault.
Villager 5 has finished and left the vault.

Question 2: The Syncville Library's Reading Room

Story: The Syncville Library has a special reading room with limited seats. Villagers enter the library in pairs to discuss research. However, if only one person is left, they must wait for a partner before entering the room. The librarian, Condition Variable, must carefully balance the access, ensuring villagers only enter in pairs, and no one is left alone waiting indefinitely.

Task: Write a C program where villagers (threads) enter the library in pairs using condition variables. Implement the logic to ensure:

- Threads only enter in pairs.
- No one is left waiting indefinitely (solve the possible deadlock).
- Each thread notifies when they've completed their discussion so that the next pair can enter.

Example Input:

- 5 villagers attempt to enter the reading room.
- Only pairs can enter the room.

Expected Output:

Villager 1 is waiting for a partner.
Villager 2 is waiting for a partner.
Villager 1 and Villager 2 have entered the reading room.
Villager 3 is waiting for a partner.
Villager 4 is waiting for a partner.
Villager 3 and Villager 4 have entered the reading room.
Villager 5 is waiting for a partner.
Villager 5 has no partner, waiting indefinitely for another villager.