**Develop a C program that implements solutions for the following sync. Problems**

(1) Sleeping Barber: A barbershop consists of a waiting room with n chairs, and the barber room containing the barber chair. If there are no customers to be served, the barber goes to sleep. If a customer enters the barbershop and all chairs are occupied, then the customer leaves the shop. If the barber is busy, but chairs are available, then the customer sits in one of the free chairs. If the barber is asleep, the customer wakes up the barber.

(2) Jurassic Park consists of a dinosaur museum and a park for safari tiding. There are n passengers and m single-passenger cars. Passengers wander around the museum for a while, then line up to take a ride in a safari car. When a car is available, it loads the one passenger it can hold and rides around the park for a random amount of time. If the m cars are all out riding passenger around, the passenger who wants to ride waits; if a car is ready to load but there are no waiting passengers, then the car waits. Use semaphores to synchronize the n passenger threads/processes and the m car threads/processes.

(3) Sleeping TA Problem (exercise of Schilbertz – Synchronzation Chapter)

------------------------------------------------------------

(4) Simulate a Banking Scenario using semaphores for the following description:

Bank shall maintain an overall balance and there shall be three counters supported in the bank for credit or debit operations by customers.

Number of customers accessing counters should be dynamic.  A withdrawal or credit transaction by a customer thread would update both his local account balance and a global overall bank balance.

Synchronize the balance statistic (both local and global ) to allow across account (differing account numbers / customers) and within account transactions. (Two different account withdrawals should not violate the overall balance and the order of credit / debit operations should be synchronised in the case of within account transactions.)

The setup should also allow joint account holders (two transaction id’s over the same customer id) to perform debit (sync issue) and debit operations.

Differentiate multiple transactions by the same customer with a transaction id.

Also support a shared token generator for use by the counters which shall be mapped on to transaction ids/customer threads for performing the transaction.

Adopt standard banking convention such as a minimum balance for each customer and an minimum overall bank balance =  minimum balance of all customers.