The Sleeping Barber Problem: A barbershop contains one barber, one barber chair, and N chairs for waiting customers, if any, to sit in. If there are no customers present, the barber sits down in the barber chair and falls asleep. When a customer arrives, he has to wake up the sleeping barber. If additional customers arrive while the barber is cutting a customer's hair, they either sit down (if there are empty chairs) or leave the shop (if all chairs are full).

Design semaphore-based implementations for processes Barber( ) and Customer( ), viewed as C++ functions, so that a single Barber( ) process and an arbitrary number of Customer( ) processes will interact as described above. Describe your implementations using standard C++ syntax, similar to the solutions shown in class for the Producer-Consumer and Dining Philosophers problems. Be sure to show all declarations, including initializations, and make clear which variables are shared by all processes.

Shared variables:

```cpp
void Customer( ) {
}

void Barber( ) {
}
```