Programming Exercise 3 - Sleeping Barber Problem Revisited

In last week's exercise you wrote a Pthread program to solve a Sleeping Barber problem. Now the barber hired a shop assistant.

The barber shop still has N waiting chairs for customers. When a customer arrives and if there are free waiting chairs, the customer picks up a ticket from the front desk and then sits in a waiting chair, waiting for the ticket number to be called. The tickets are numbered from 1 to N and initially placed in ascending order. Customers always pick up a ticket from one end and return the ticket to the other end after being called. In this way the ticket numbers are always maintained in a perfect cyclic order.

The shop assistant looks after the customers. When the barber becomes available, the assistant will call one waiting customer if any. Customers will be called in a first-come-first-served order.

Revise the program you did last week to adapt to the scenario described above.

You need to write three thread routines, i.e., `assistant_routine`, `barber_routine` and `customer_routine`.

In `main` the program first asks the user to give a few parameters:

- `no_of_seats` – the total number of waiting chairs in the barber shop;
- `no_of_consumers` – the total number of consumers to be created;
- `T1` – the minimum number of time units for barber to service a customer;
- `T2` – the maximum number of time units for barber to service a customer;
- `T3` – the minimum number of time units between two customer arrivals;
- `T4` – the maximum number of time units between two customer arrivals.

You can change these input parameters and see behaviours of the barber, the assistant and the customers accordingly when running the program.

To check whether your program functions properly, your program must print the following statements.

The assistant thread must print the following status messages wherever appropriate:

- "`Assistant: I'm waiting for customers.`" – Wait for customers
- "`Assistant: I'm waiting for barber to become available.`" - Waiting for barber to become available
- "`Assistant: Call one customer with a ticket numbered n.`" - Call the customer with a ticket numbered n to get haircut next
- "`Assistant: Hi Barber, we've finished the work for the day.`" - No more customers and call barber to terminate

The barber thread must print the following status messages wherever appropriate:

- "`Barber: I'm now ready to accept a new customer.`" - Barber is ready to service a new customer

- "`Barber: Hello, customer n.`" - To service one customer with a ticket numbered *n*
- "`Barber: Finished cutting. Good bye, customer n.`" - Finished haircut for customer with a ticket numbered *n*
- "`Barber: Thank Assistant and see you tomorrow!`" - Finished the work for the day and terminate

Each Customer thread must print the following status messages wherever appropriate:"

- `Customer [id]: I have arrived at the barber shop.` " - New customer with customer id arrives
- "`Customer [id]: oh no! all seats have been taken and I'll leave now!`" - All chairs are occupied
- "`Customer [id]: I'm lucky to get a free seat and a ticket numbered n`" - Take a ticket numbered *n* and sit on a free waiting chair
- "`Customer [id]: My ticket numbered n has been called. Hello, Barber!`" - To be serviced next
- "`Customer [id]: Well done. Thank barber, bye!` " - Get haircut done and leave