Programming Exercise 2 - Sleeping Barber Problem

In this exercise you'll write a Pthread program to solve a little bit more complicated problem, i.e., Sleeping Barber problem.

The sleeping barber problem is a classic inter-process communication and synchronization problem. A barber shop has one barber and a limited number of chairs for waiting customers. When there are no customers, the barber waits and sleeps. As soon as a customer arrives, he tries to sit down on one of the free chairs. If all of the chairs are occupied, the newly arrived customer simply leaves (and vanishes). The waiting customers wait to be called by the barber one at a time. Then, the customer vacates the chair and come to sit on the barber chair for haircut. When the customer is sited, the barber serves the customer for a certain period of time. The customer leaves after being served.

Write a program to solve this problem using **condition variables** and **mutexes** to provide synchronization between the barber and customers.

You need to write two thread routines `barber_routine` and `customer_routine`.

In `barber_routine`, you need to add print statements to print out

- "`Barber: The number of free seats is %d. No customers waiting and I'll go to sleep.\n`" - when there are no customers;
- "`Barber: The number of free seats now is %d. Call next customer. \n`" - there are waiting customers, call one customer
- "`Barber: wait for the customer to sit on the barber chair.\n`" - wait for the customer to sit on the barber chair
- "`Barber: Start serving the customer.\n`" - cutting hair
- "`Barber: finished cutting. Bye!\n`" - finished cutting.

In customer_routine, you print out

- "`Customer %d arrives.\n`" - when a new customer arrives;
- "`Customer %d: oh no! all seats have been taken and I'll leave now!\n`" - all chairs are occupied;
- "`Customer %d: I'm lucky to get a free seat from %d\n`" - get a free chair from a total number of free chairs;
- "`Customer %d: I'm to be served.\n`" - being called by the barber
- "`Customer %d: sit on the barber chair.\n`" - sit on the barber chair
- "`Customer %d: I'm being served.\n`" □ getting haircut
- "`Customer %d: Well done. Thank barber, bye!\n`" - done and leave.

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;
- `barber_pace` – barber's working pace, may spend different time (in a range of `0` to `barber_pace` seconds) for different customers;
- `consumer_rate` – consumer inter-arrival time between `0` to `cosumer_rate` seconds.

You can change these numbers and see the waiting queue and behaviours of the barber and the customers accordingly when running the program.