**1-Solution pseudocode**

1- If there is no customer to be served, the barbers are sleeping:

if numCustomer=0

then

barberState := sleeping

2- if the customer enters the barbershop, and the barber is sleeping:

if numCustomer=1 AND barberState=sleeping

then

barberState:= awake AND availableBarber=1

3-If a customer enters the barbershop, and the one of the two barbers is sleeping and the other is unavailable:

if availableBarber=1

then

TheOtherBarberState := awake AND availableBarber :=0

4-If the customer enters the barbershop, and the two barbers are busy:

if available Barber=0 AND numOfWatingChairs=0

then

customerState := waiting AND WaitingChairsCounter--

5-If a customer enters the shop and there are no available chairs:

if availableBarber=0 AND numOfWatingChairs=0

then

customerState := leaving

6- if the two barbares ar done and ther are no more waiting customer :

if numOfCustomer=0 AND unWorkingBarbers = numOfBarbers

then

BarberState :=sleeping

**2-Examples of Deadlock:**

**In general** deadlock is Situation where two (or more) operations need overlapping sets of resources, and neither can complete because they cannot obtain all locks necessary to complete an operation and release their locks.

Deadlock is a situation where multiple operations are waiting for the same resource(s).

**In our problem** the deadlock will occur if the customer ends up waiting for the barber and the barber ends up waiting for the customer to arrive and that would happen if many customer threads entered the critical section all at once

**3-How did solve deadlock?**

Making a customer list to put in any customer who enters the shop following FIFO strategy or handling this problem we used reentrant locks and after a thread acquires a lock it sleeps for few milliseconds and then release the lock. The code ensures that each thread releases the lock after performing the critical section. I have also used try-catch blocks to handle exceptions. Using reentrant locks, the critical section which is inside the locks can only be accessed by one thread at a time.

**4-Examples of starvation:**

the starvation will occur if the customers don’t follow any order for getting a haircut, as some won’t get a haircut even though they've been waiting for a long time.

**5-How did solve starvation?**

To handle this problem, we inserted the customers in a linked list which follows the first in first out property. So, every time a customer sits in a waiting room, they will be selected by the barber in first come first serve basis.

**6-Explanation for real world application and how did apply the problem:**

for the real world application we used the TA as a method where instead the barber it's the TA who's waiting in the teaching hall(shop) for the students(customers) to come and ask if they have any questions so the TA sleeps waiting for the students and the students enter the hall first and if the TA is found to be not busy he goes to him and if he's busy the student wait in the chair and served by order and if the chairs are full the student shall go and return back after a random time to ask.