EOPSY Lab 5

Barber Synchronization

1. Problem Description

The analogy is based upon a hypothetical barber shop with many barbers serving women and man. In a barber shop there are N1 barbers serving only women, N2 barbers serving only men and N3 barbers serving both women and men. Each barber has one barber's chair in a cutting room. In a waiting room there are M chairs. When the barber finishes cutting a customer's hair, he dismisses the customer and goes to the waiting room to see if there are others waiting. If there are, he brings one of them (but only if he is able to cut hair, i.e. the barber brings a man into the cutting room only if he can cut his hair, etc.) back to the chair and cuts hair (it lasts a random time). If there are none, he returns to the chair and sleeps in it. Each customer can be either a male client or a female client. When a client arrives, he/she checks what barbers are doing. If there is any barber sleeping (who is able to serve the client), the customer wakes him up, and sits in the cutting room chair. If all barbers (able to serve the client) are cutting hair, the customer stays in the waiting room. If there is a free chair in the waiting room, the customer sits in it and waits their turn. If there is no free chair, the customer leaves.

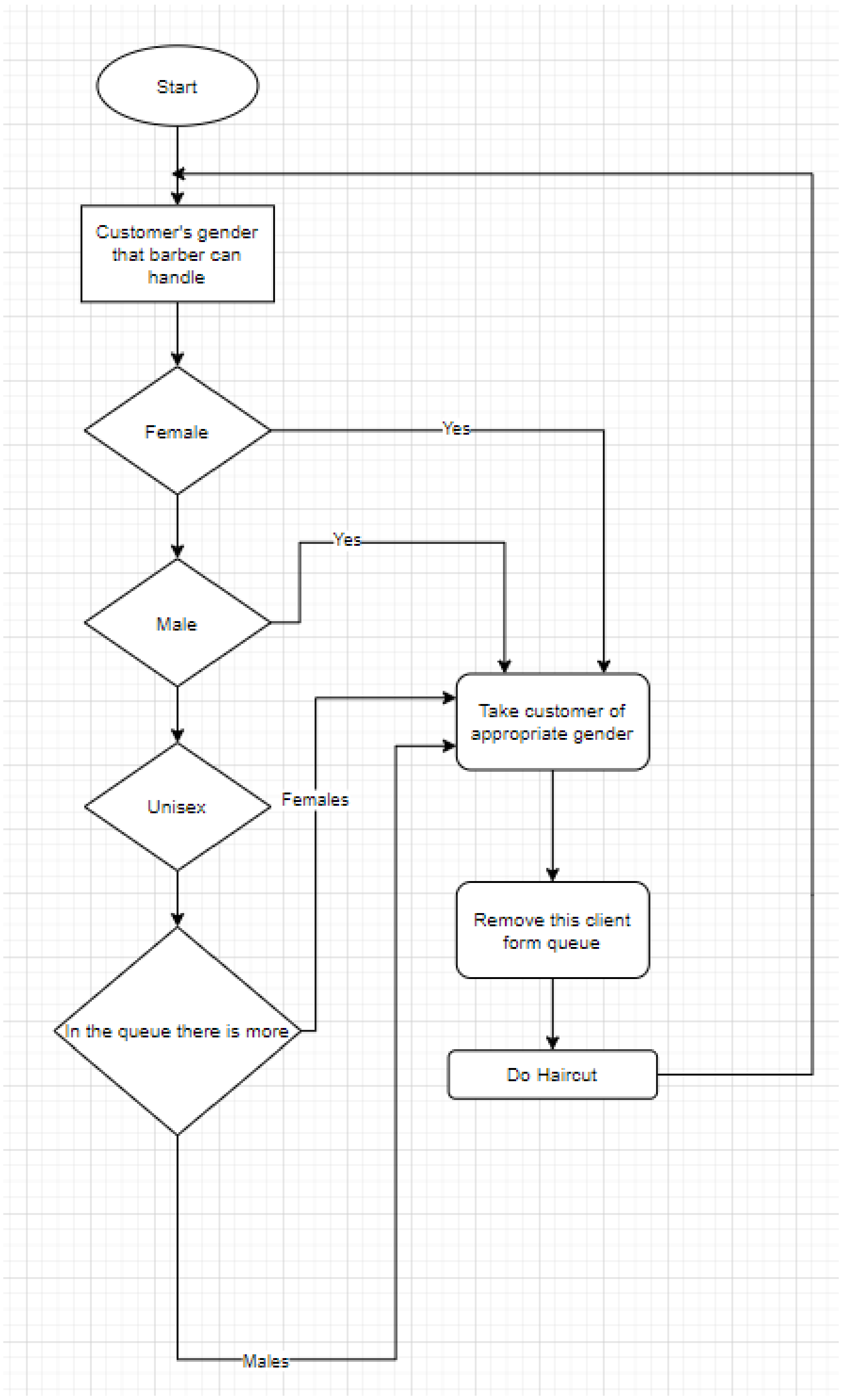
Implement the C language the sleeping barber problem. The program must be parameterized by N1, N2, N3 - number of barbers, M - number of chairs in waiting room.

Print on the standard output verbose messages from customers and the barbers.
Execute the program with different N1, N2, N3 and M values.

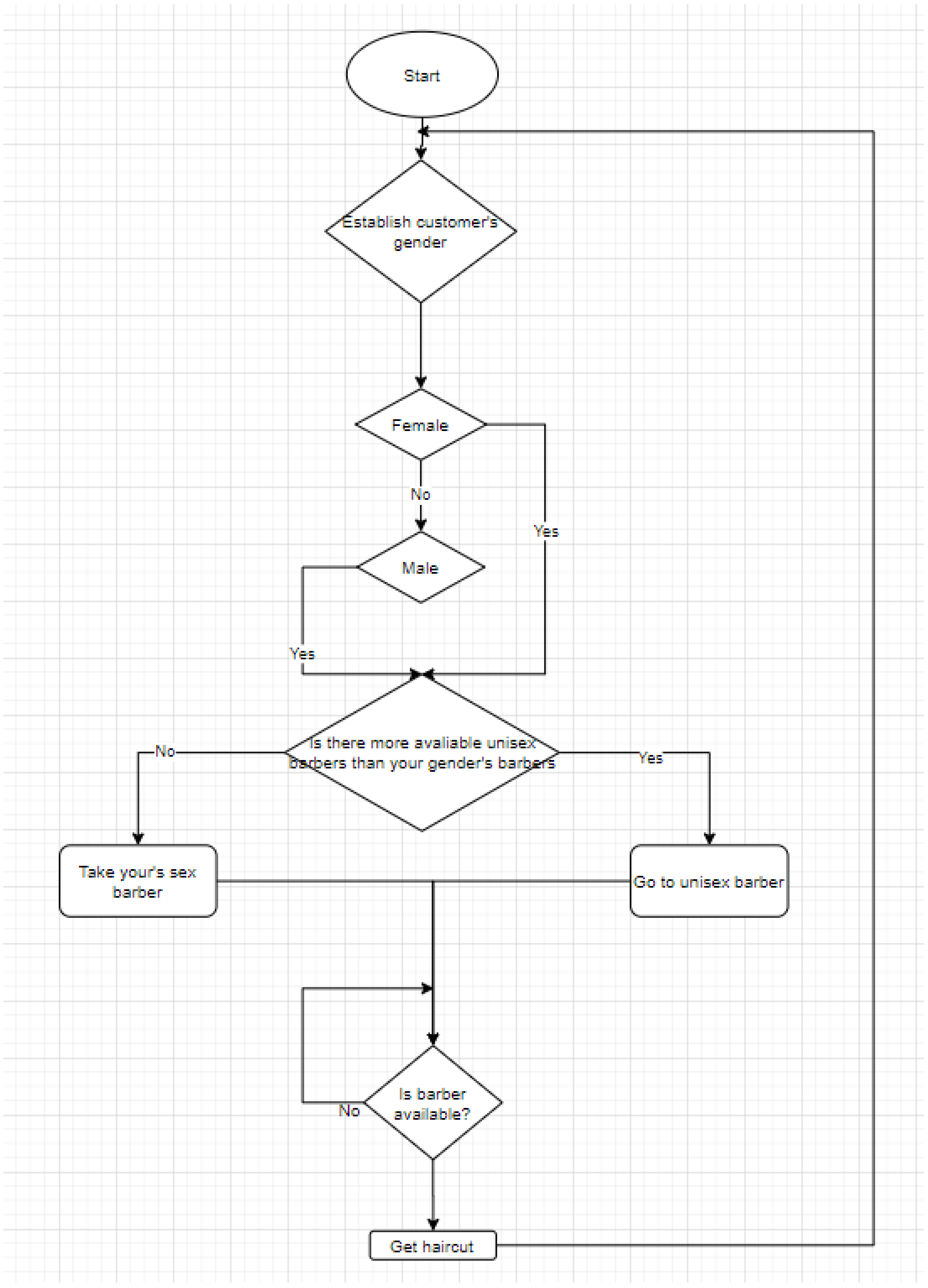
Stated problem is a little bit more complicated than the classical sleeping barber problem, but there are some similarities between those two problems.

2. Algorithm Description

Solution can be found in the **barbers.c** file. It has been implemented using C language which provides build in methods for controlling system V semaphores. They will allow us to access critical fragments of the code and regulate access to those variables by parallel processes. The entire algorithm is focused on two functions: *customer, barber* that which are designed to implement appropriate behaviors of such mentioned in the name. Below you can find flowchart implementation.


Start
Customer's gender that barber can handle
Female
Yes
Male
Yes
Unisex
Females
In the queue there is more
Males
Take customer of appropriate gender
Remove this client form queue
Do Haircut

1. Barber function flowchart



2. Customer function flowchart

3. Running simulation

In order to run the simulation get into EOPSY/Barbers directory. Type in command: ***make*** and ***make run***.