Project Notes

Pizzeria.h
In the .h file we have defined and initialized the constants, resources, statistics, mutexes and mutex states.

MAIN
In main we accept the two parameters from the user, the seed and the number of clients, we initialize the mutexes and their states, and then create the threads that call the order function which takes as argument the id of the order (1 to the number of customers). Once the threads are created, we join them and print t heir results. Finally, we destroy the mutexes/wait conditions.

Order Function (ID)
In this method we implement the preparation and delivery of the order for each client. First, we declare th e local variables and then we start executing the order step-by-step.
-Basically we have taken every global variable in the function as a resource, so we lock it wherever we ne ed to.
-We have added comments to the key points in the code to make it easier to read and understand how it is implemented.

COMMENTS ON THE RESULTS:
We noticed that the actual times for the delivery and the cooling of the pizza are big. This is due to the fac t that the customers' times start before the resources are locked up, with the result that many customers wait a long time for the resources to be released, and the fact that we only have two packaging employee s and that each oven can only hold one pizza leads to additional delay. Finally, we noticed that the later a n order is registered, the longer it takes to prepare and deliver it.