# Assignment1: Inter Process Communication

It's 2023 and the lockdown is finally over!!! You along with your 10 friends have gathered at a restaurant for a get together. Each one of you is going to place your order seperately.

There are three Chefs: ChefA, ChefB, ChefC in the restaurant to prepare your food (process your orders). At a time they can handle at most 5 orders from customers. So, a customer may have to wait while trying to place his order.

Whenever a order is processed by any of the chefs (ChefA or ChefB or ChefC), they send it to the MasterChef for final checking. The MasterChef can check only one order at a time.

Once the MasterChef is done checking the food he orders the waiter to send it to the customer.

Write code to synchronize the customer threads, chef threads, masterhchef thread and waiter

thread.

Hints:
1. You might have to use multiple buffers for solving this problem.
2. Similarly, you might have to use multiple semaphores and locks.
3. Use sprod_scons_with_mutex.cpp and mprod_scons_with_mutex.cpp as the starting block to solve the problem
4. For specification of POSIX threads and semaphores please go through these links: threads and semaphores

## Mark Distribution (Marks 80)

Sempahores, Locks & buffers: 10
Implementing Customer: 10
Chefs (A, B, C): 20
Masterchef: 20
Waiter: 10
Output: 10

## Sample Output

It is obvious that the output of your program is not going to be the same as the one provided below. However, this will helo you in implementing the threads.

I am customer 1
I am customer 3
I am customer 2
I am customer 4
I am customer 5
I am customer 6
I am customer 7
I am customer 8
I am customer 10

I am customer 9
I am chefA
I am chefB
I am chefC
I am Waiter
I am Masterchef
customer 1 has placed his order
customer 3 has placed his order
customer 2 has placed his order
customer 4 has placed his order
customer 5 has placed his order
ChefA has prepared order 1
ChefA has sent order 1 to
masterchef ChefB has prepared
order 3
ChefC has prepared order 2
masterchef has checked order 1
ChefB has sent order 3 to masterchef
masterchef has called waiter to collect order
1 customer 6 has placed his order
masterchef has checked order 3
waiter has delivered order 1
ChefA has prepared order 4
customer 10 has placed his order
masterchef has called waiter to collect order
3 ChefC has sent order 2 to masterchef waiter
has delivered order 3
masterchef has checked order 2
customer 8 has placed his order
ChefB has prepared order 5
ChefA has sent order 4 to masterchef
masterchef has called waiter to collect order
2 waiter has delivered order 2
masterchef has checked order 4
ChefB has sent order 5 to
masterchef customer 7 has placed
his order
ChefC has prepared order 6
masterchef has called waiter to collect order
4 customer 9 has placed his order
…
…
…