# How exactly is synchronization achieved using semaphore in our assignment?

Two Semaphores are created, viz. consumed and produced. The producer must wait for the consumer to consume, before producing the next value and similarly consumer must wait for the producer to produce. This situation can be handled with the use of semaphores.

By using semaphore, we make the producer wait for consumer to consume, with wait(<consumer semaphore>). Once the consumer consumes, producer generates a new value and signal the consumer with signal(<producer semaphore>), to indicate a new value has been generated.

Similarly the consumer waits for producer to produce, with wait(<producer semaphore>). Once the producer produces, consumer consumes the value and signals the producer that value has been consumed with signal(<consumer semaphore>).

# OUTPUT

# C:\Users\Siddharth Jain\Desktop\synced.png

# Can the above synchronization be achieved with just one semaphore? Why or why not?

No. The reason we need two semaphore is, because the producer must wait for the consumer to consume the value, thus one semaphore is required for maintaining the status of consumer, i.e. we need a semaphore for signalling the consumer has consumed and also for waiting on, till consumer has consumed. Whereas we need the second semaphore for maintaining the status of producer, i.e. we need a semaphore for signalling that producer has produce and also for waiting on, till producer has produced.

# Tasks

Prateek:

* Created consumer.c, producer.c, tested on device

Siddharth:

* Created xsh\_prodcons.c, prodcons.h, tested on device