Lab 2 Observations

* Both of these could happen as it is how the concurrency is designed to operate. The queue is emptied, room is available, and more objects are written to it. The queue is added to, there are now objects available and they are they dequeued.
* No, the procedures are mutually exclusive. When a procedure starts accessing the data structure, no other procedure can access it as it is a side-effecting procedure. When the data structure is read as empty, there is no way for objects to be added to it making it also full.
* Producers all have their given entry loops that are the same size = Yes
* No, the consumers are not guaranteed to read the same amount, there is no guarantee that they will have equal access while not being blocked.
* Yes, this makes sense because there is no way for an element to be double read as all procedures are blocked when one is accessing the data structure.
* Not in this implementation, it is designed so that consumers will exit when they read a Some\_Characters’Last which will cut them out before they could potentially read another keeping the other consumers indefinitely locked. If the implementation was flawed there could exist the possibility of an unequal amount of consumers and producers causing issues or there not being enough exit variables available for the consumers.