**Task1:**

**Semaphore is one of the concurrency mechanisms available. Find out about more concurrency mechanisms. How do these mechanisms protect critical sections? Compare their implementations with wait() and signal() operations of semaphores.**

**Answer:**

A semaphore does the allows multiple number of threads to access a particular resource until and unless there is a need of mutual exclusion.

A lock allows only one thread to enter the part that's locked and the lock is not shared with any other processes.

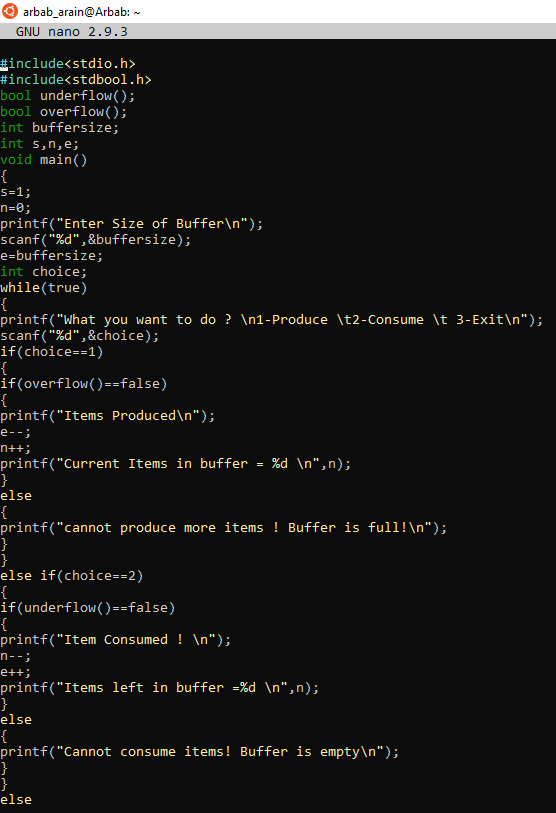
A mutex is the same as a lock but it can be shared by multiple processes. In mutex, acquire will let first caller through, and then block next until release

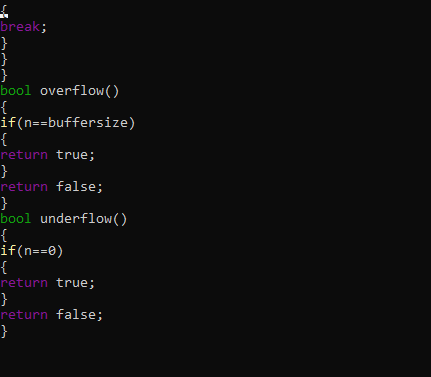
In condition synchronization, acquiring will block first caller until release.

**Task2:**

**Implement Producer Consumer Problem in C language using Semaphore.**

**Code:**





**Output:**

