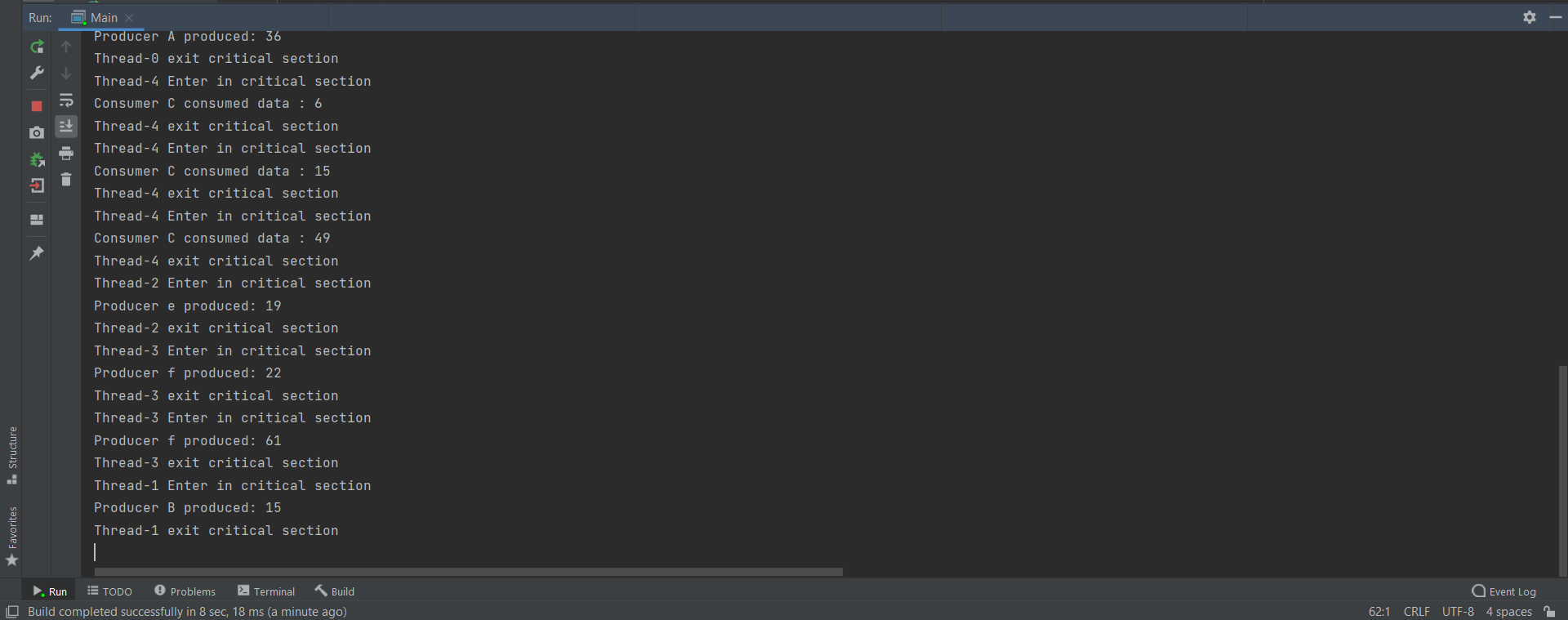
Deadlock Problem:

While using semaphores the concept of ownership of mutex)lock) and the order of increment and decrement of the semaphores should be kept in mind . Any change in order may lead to deadlock so, Order of Wait’s are very important?



Explanation:

If we replace order of wait(empty) with wait(mutex) or

replace order of wait(full) with wait(mutex)

It will prevent other producer or consumer to get in critical section because if we replace it that means we take the lock(mutex) before check the buffer is empty or full, so the consumer will get into critical section with (lock) and may be the buffer is empty so consumer will be wait in the queue with the lock and will not take item, so it’s prevent other consumer to get into critical section because he have the key

From the producer view he will enter the critical section with lock and may be the buffer is full so he will not add something and wait in queue, so other producers cannot get into critical section because he has the key So, it will lead to deadlock

It’s same problem if we replace signal(mutex) with signal (full)

Because producer must return the (lock) before leaving the critical section to let another consumer enter critical section

So, it will lead to deadlock









Solution of Deadlock :

We must verify the order of the wait and signal commands before deployment which is the responsible of programmers

the semaphores should be kept in mind . Any change in order may lead to deadlock



