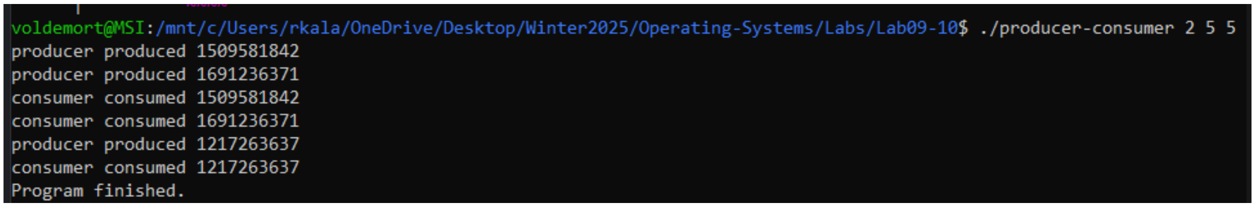
- Two real-world applications of producer/consumer problem are:
  i. Video streaming services like Netflix:
  Producer-> content being broken into data packets
  Consumer-> video player consuming frames for display
  ii. Message queues in distributed systems (DS):
  Producer: Web servers generate requests/messages (API Calls)
  Consumer: Worker processing messages (update inventory)
- How can we engineer a solution to this problem? How does our approach help resolve? One for each example above
  i. For video streaming services we can use circular buffer which is a fixed-size frame buffer. It will absorbs network delays resulting in smooth playback. If the buffer is full, the producer waits so there will not be any frame drops. Consumer always pulls frames at the correct rate which achieves real time sync.
  ii. For message queues in DS we can use a distributed queue. It will prevent overflows because queue has max capacity and producers wait if it is full. And if a consumer crashes, the message is re-queued, so there is fault tolerance.



```
voldemort@MSI:/mnt/c/Users/rkala/OneDrive/Desktop/Winter2025/Operating-Systems/Labs/Lab09-10$ ./producer-consumer 2 5 5
producer produced 1509581842
producer produced 1691236371
consumer consumed 1509581842
consumer consumed 1691236371
producer produced 1217263637
consumer consumed 1217263637
Program finished.
```