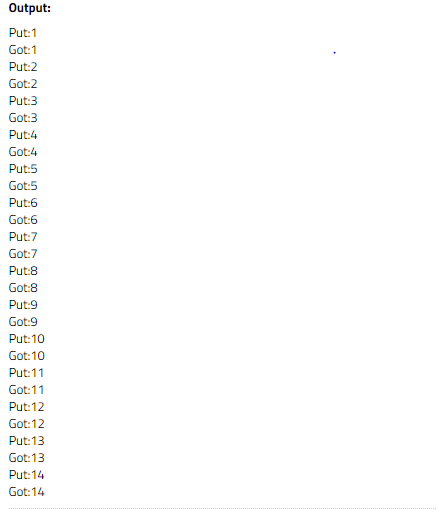
**note :** the execution runs continuosly...... to stop it, click the terminate button (red one) in case of eclipse. if u r running the program on the command line, then press **Control+C**



................

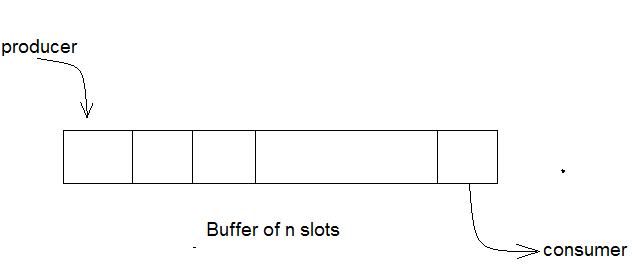
................

**producer - consumer concept**

In [computing](https://en.wikipedia.org/wiki/Computing), the **producer–consumer problem**[[1]](https://en.wikipedia.org/wiki/Producer%E2%80%93consumer_problem#cite_note-ostep1-1)[[2]](https://en.wikipedia.org/wiki/Producer%E2%80%93consumer_problem#cite_note-ostep2-2) (also known as the **bounded-buffer problem**) is a classic example of a multi-[process](https://en.wikipedia.org/wiki/Process_(computing)) [synchronization](https://en.wikipedia.org/wiki/Synchronization_(computer_science)) problem. The problem describes two processes, the producer and the consumer, who share a common, fixed-size [buffer](https://en.wikipedia.org/wiki/Buffer_(computer_science)) used as a [queue](https://en.wikipedia.org/wiki/Queue_(data_structure)). The producer's job is to generate data, put it into the buffer, and start again. At the same time, the consumer is consuming the data (i.e., removing it from the buffer), one piece at a time. The problem is to make sure that the producer won't try to add data into the buffer if it's full and that the consumer won't try to remove data from an empty buffer.

**What is the Problem Statement?**

There is a buffer of n slots and each slot is capable of storing one unit of data. There are two processes running, namely, **producer** and **consumer**, which are operating on the buffer.



**Bounded Buffer Problem**

A producer tries to insert data into an empty slot of the buffer. A consumer tries to remove data from a filled slot in the buffer. As you might have guessed by now, those two processes won't produce the expected output if they are being executed concurrently.

There needs to be a way to make the producer and consumer work in an independent manner.