Semaphore S is an integer variable that can be accessed only through two standard operations: wait() and signal().

wait() → reduce 1

signal() → increase 1

It is used to control access to a common resource by multiple process in a concurrent system.

Producer consumer problem is a classic example of multi process synchronization communication.

The problem describes two processes, The producer and The consumer who share a common, fixed size buffer. The Producer's job is to generate data, Put it into the buffer & start again.

At the same time the consumer is consuming data (removing it from buffer) one piece at a time.

The Problem is to make sure that the producer won't try to add data into buffer if it's full and that the consumer won't try to remove data from an empty buffer.

```
int count = 0;
void producer(void)
{ int itemp;
  while (true)
  {
    produce_item (itemp);
    while (count == n);
      Buffer [in] = itemp.
      in = (in+1) mod n;
      count++;
  }
}
void consumer (void)
{ int item c;
    while(true){
    while (count = = 0)
    itemc = Buffer (out)
    out = (out+1) mod n.
    count --;
    Process item (item c)
  }
}
```