INF2140: Exercises Week 5
(CHAPTER: MONITORS AND CONDITION SYNCHRONIZATION, PART 1)

1. Exercise 5.1 from the book: ONEBUF = (put -> get -> ONEBUF).

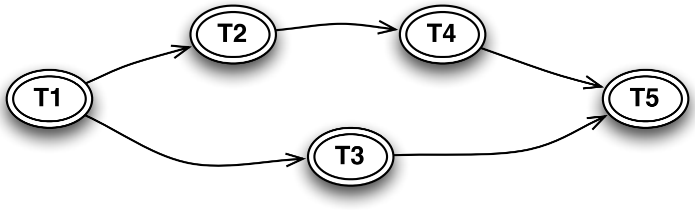
2. Exercise 5.2 from the book:

3. Exercise 5.3 from the book. Implement the monitors in Java

4. A precedence graph is a directed, acyclic graph. Nodes represent tasks and arcs indicate the order in which tasks are to be accomplished. In particular a task can execute as soon as all its predecessors have been completed. Assume that the tasks are processes and that each process has the following outline:

   ```
   T = (
   // wait for predecessors if any
   task ->
   //signal successors if any
   STOP).
   ```

   Using semaphores, show how to synchronize five processes whose permissible execution order is specified by the following precedence graph:



5. The bear and the honeybees: N honeybees and a hungry bear share a pot of honey. The pot is initially empty; its capacity is H portions of honey. The bear sleeps until the pot is full, then eats the honey and goes back to sleep. Each bee repeatedly produces one portion of honey and puts it in the pot. Implement this example in FSP, model the pot using condition synchronization.

6. Modify the exercise 5, so now the pot process is implemented using semaphores.

7. Implement the pot process of exercise 5 as a monitor in Java

8. Implement the pot process of exercise 6 using semaphores in Java