Semaphore: a special data structure protected by the OS that contains an integer value and on which 3 operation can be performed:

- sem_init(): initializes the integer inside the semaphore to a specified value.
- sem_wait(). Other notations: sem_down(); P(from Dutch)
- sem_post(). Other notations: sem_signal(); sem_up(); V (Dutch)

```
sem_wait(sem_t *s){
        decrement value of semaphore;
        if value of semaphore < 0, then
        block
}

sem_post(sem_t *s){
        increment value of semaphore;
        if value <=0, then a process blocked at s, if any, is unblocked
}
```

What you cannot do on semaphores:

- you cannot peek to the integer value stored in the semaphores. You can only run sem_wait, sem_init, sem_post.
- After process P1 posts and P2 wakes up, either can run next
- A post may wake up one process or none, if none was waiting.

1. How to implement with semaphores:

   Serialization: a1 runs before b1

Process A	Process B
…	…
a1;	b1;

2. Rendezvous: a1 runs before b2 and b1 runs before a2

Process A

…
a1;

a2;
…

Process B

…
b1;

b2;
…

3. Mutual exclusion

Process A

…

<critical section>

…

Process B

…

<critical section>

…

4. Multiplexing: at most N threads in the critical section

Process i

…

<critical section>

…