Synchronization Exercises
Exercise 1

Let $S$ be a semaphore that is initialized to 2

Consider the following:

```plaintext
down(S)
down(S)
up(S)
down(S)
```

Does this program block?
Exercise 1: Answer

- No
- The first **down** decrements from 2 to 1
- The second **down** decrements from 1 to 0
- The **up** increments from 0 to 1
- The third **down** decrements from 1 to 0
Exercise 2

- Let $S$ be a semaphore that is initialized to 2

- Consider the following:
  
  - down($S$)
  - down($S$)
  - up($S$)
  - down($S$)
  - down($S$)
  - down($S$)

- Does this program block?
Exercise 2: Answer

- Yes it does
- The first down decrements from 2 to 1
- The second down decrements from 1 to 0
- The up increments from 0 to 1
- The third down decrements from 1 to 0
- The 4th down blocks the process
Exercise 3

- Thread B must do operation \texttt{opB} only after thread A has done \texttt{opA}
- How can you guarantee this using semaphores?
Exercise 3: Incorrect Answer

thread A

down(S)
opA
up(S);

thread B

down(S)
opB
up(S)

Initialize S to 1

Why incorrect? Thread B could get to down(S) before thread A. -- in this case B executes opB before A executes opA
Exercise 3: Correct Answer

Thread A
- \text{opA}
- \text{up}(S);

Thread B
- \text{down}(S)
- \text{opB}

Initialize S to 0

Even if B gets to \text{down}(S) before A gets to \text{opA} it will block until A has executed \text{opA}
Exercise 4

- Assume process A and process B are to take turns executing opA and opB respectively?
- Process A starts first
- How can you guarantee this using semaphores?
Exercise 4: Incorrect Answer

Process A
while(1)
  opA
  up(S_1);
  down(S_1);
}

Process B
while(1) {
  opB
  up(S_2);
  up(S_1);
}

Initialize $S_1$ to 0 and $S_2$ to 1
Process B could beat process A to executing opB before opA
Exercise 4: Correct Answer

Process A
while(1){
  down(S\(_2\))
  opA
  up(S\(_1\));
}

Process B
while(1) {
  down(S\(_1\))
  opB
  up(S\(_2\));
}

Initialize S\(_1\) to 0 and S\(_2\) to 1

Process A will always execute first since the first time it hits down(S\(_2\)) it doesn’t block while process B always blocks the first time it hits down(S\(_1\))
Exercise 4: Correct Answer

Process A

while(1)
{
    opA
    up(S_1);
    down(S_2)
}

Process B

while(1)
{
    down(S_1)
    opB
    up(S_2);
}

Initialize S_1, S_2 to 0

The first time process B enters the loop it blocks but A doesn’t.
Exercise 5

- Why is it important to keep the size of the critical section as small as possible
Exercise 5: Answer

- As a critical section can be executed by one thread/process at a time.
- Other threads/processes must wait until the current thread must finish. Long critical sections lead to lower throughput.