

1. Semaphores and Deadlock

- What solutions in the semaphore deadlock? How?
- Write non-deadlocking semaphore versions of these solutions.

2. 2-phased locking.

```
// variables x and array a are shared variables
insert(int value) {
    atomic {
        a[x] = value;
        x = x + 1;
    }
}
```

- Describe the locking, unlocking, and possible blocking in 2-phased commit when executing the above atomic operation.
- If different process execute `insert`, can the inserts execute concurrently. Explain.
- Assume 5 different processes, p_0, p_1, \dots, p_5 . Process p_i calls (once) `insert(i)`, and x is initially 0. What are the possible outcomes?

3. Deadlock and 2-phased locking. Consider an atomic operation which first reads A, and then later writes A. Does this deadlock? If it does deadlock, can you describe an implementation which does not deadlock. If it doesn't deadlock, explain why.