HW4

16.3 1. T1: R(x) R(y) W(x)
T2: R(x) R(y) W(x) W(y)

2. T1: R(x) R(x) W(x) commit
T2: R(x) W(x) commit

3. T1: W(x) W(y) commit
T2 W(y) W(x) commit

4. a. T2 can't get shared lock on x because T1 would have exclusive lock on x, meaning T1 would have to finish first. (write-read)

b. T1 can't get exclusive lock on x until T2 commits. (read-write)

c. T1 can't have a lock on W(y) before T2 commits on W(y). Same thing with T2 not being able to have a lock on W(x) before T1 commits on W(x).

16.71. Read uncommitted - A new row named student is inserted into "Introduction to Database System" class. Since only requires a new row, no lock is needed on existing rows.

2. Read committed - An existing row needs to be updated in "Enrolled" so exclusive lock is needed.

3. Serializable - Phantom problems such an existing transaction that's already reading the table preventing an insertion or update to the "Enrolled" table.

4. Serializable - Same thing, phantom problem such as the prevention of the transaction to insert/update the "Enrolled" table by an existing transaction that's already reading the table.