1. (10') Since every conflict-serializable schedule is view serializable, why do we emphasize conflict serializability rather than view serializability?
2. (30') The lost update anomaly is said to occur if a transaction T_j reads a data item, then another transaction T_k writes the data item (possibly based on a previous read), after which T_j writes the data item. The update performed by T_k has been lost, since the update done by T_j ignored the value written by T_k.
    a. Give an example of a schedule showing the lost update anomaly.
    b. Give an example schedule to show that the lost update anomaly is possible with the read committed isolation level.
    c. Explain why the lost update anomaly is not possible with the repeatable read isolation level.
3. (20') Consider the following two transactions:

    T_1: read(A);
    read(B);
    if A = 0 then B:= B + 1;
    write(B).

    T_2: read(B);
    read(A);
    if B = 0 then A:= A + 1;
    write(A).

    Add lock and unlock instructions to transactions T_1 and T_2 so that they observe the two-phase locking protocol. Can the execution of these transactions result in a deadlock?
4. (40') Consider the following sequence of actions S,

    S: r1(A), r2(A), r3(B), w1(A), r2(C), r2(B), w2(B), w1(C)

    and answer the following questions:
    a. Is the schedule S view-serializable? If so, provide a view-equivalent serial schedule.
    b. Is the schedule S conflict-serializable? If so, describe all the conflict-equivalent serial schedules.
    c. Is the schedule S a 2PL schedule (with exclusive locks)?
    d. Is the schedule S a 2PL schedule (with shared and exclusive locks)?