HW7

1. Consider the schedule given below in Table below. R(·) and W(·) stand for 'Read' and 'Write', respectively. T stands for transactions and t stands for time stamps.

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
T1			W(B)		R(H)	R(I)	W(A)		R(K)	
T2	R(A)		W(D)	W(E)		W(F)		W(J)		
T3		R(C)			W(G)	R(D)	W(H)	R(B)	R(E)	R(G)

   i. Give the dependency graph of this schedule.
   ii. Is this schedule conflict serializable? If you answer "yes", provide the equivalent serial schedule. If you answer "no", briefly explain why.
   iii. Is this schedule possible under 2PL?

2. Consider the following schedule that involves transactions T1, T2, and T3.
   S: r1(X), r2(Y), r3(Y), w2(Y), w1(X), w3(X), r2(X), w2(X)
   Is this schedule possible under 2PL? If yes, then write the corresponding serial schedule.
   If no, then provide an equivalent non-serial schedule that is possible under 2PL with minimum swapping of operations.

3. Consider the following schedules of transactions T1, T2, T3.
   S1: r2(A), w2(A), r3(C), w2(B), w3(A), w3(C), r1(A), r1(B), w1(A), w1(B)
   S2: r3(C), r2(A), w2(A), w2(B), w3(A), r1(A), r1(B), w1(A), w1(B), w3(C)
   S3: r2(A), r3(C), w3(A), w2(A), w2(B), w3(C), r1(A), r1(B), w1(A), w1(B)
   i. Which of the above schedules are conflict equivalent to each other?
   ii. Which of these schedules are possible in 2PL?
   iii. For the schedules that are not possible in 2PL, show whether it is possible to swap the operations to get equivalent schedule that is possible under 2PL.

4. S: r1(Y), w1(Y), r2(Y), w2(Y), r1(X), w1(X), r2(X), w2(X). Is this schedule possible in 2PL? Explain your answer. Produce a schedule with the same set of operations that will cause a deadlock.

5. T1= read(A), A:=A+5, write(A), read(B), B:=B-10
   T2= read(A), A:=A-10, write(A), read(B), B:=B+5
   T3= read(A), A:=A+3, write(A), read(B), B:=B-2

   A schedule is equivalent (not conflict equivalent, just equivalent) to a serial schedule S if it leaves the database in the same state as S (items have the same values)

   Is there a schedule S for T1, T2 and T3 which is not equivalent to some serial schedule? If so, show such schedule. If not, explain why.

6. Suppose for the new bar New Tavern has a remote order system which allows customers order beers by themselves. The inventory information of New Tavern is shown in the table:

Bar	Beer	Inventory
Tavern	Budweiser	13
Tavern	Heineken	22
Tavern	Pabst Blue Ribbon	3
Tavern	Corona	9

There are three customers A, B and C ordering beers remotely, they start their transactions in the given order:

1) Customer A orders two Budweisers:
   $T_1 = r1(Bud)w1(Bud)$
2) Customer B orders one Budweiser and one Pabst Blue Ribbon:
   $T_2 = r2(Bud)w2(Bud)r2(PBR)w2(PBR)$
3) Customer C orders two Coronas:
   $T_3 = \ r3(Cor)w3(Cor)$
4) Customer D orders three Pabst Blue Ribbons:
   $T_4 = \ r4(PBR)w4(PBR)$

Let

a. S_1 = r1(Bud)r2(PBR)r3(Cor)w1(Bud)r2(Bud)w2(PBR)r4(PBR)w4(PBR)w2(Bud)w3(Cor)
b. S_2 = r3(Cor)r1(Bud)r2(Bud)w1(Bud)w2(Bud)r2(PBR)w3(Cor)r4(PBR)w2(PBR)w4(PBR)

Q1: For schedules S_1 and S_2, which one is conflict-serializable? Which one is not conflict-serializable? Please justify your answer with Precedence Graph.

Q2: For the schedule that is not conflict-serializable, show it will not be granted under 2PL protocol.

Q3: Schedule that is not conflict-serializable, what troubles would it cause for New Tavern?