Question 1:

For each of the following schedules, draw a precedence graph and determine whether the schedule is conflict serializable or no:

1.1)

T1	T2	T3
R(X)		
	R(Y)	
		W(X)
	R(X)	
R(Y)		

1.2)

T1	T2	T3
R(X)		
R(Y)		
W(X)		
	R(Y)	
		W(Y)
W(X)		
	R(Y)	

Question 2:

2.1) Explain the difference between "deadlock prevention" and "deadlock detection and recovery". More details.

2.2) Explain the difference between "Wait/Die" and "Wound/Wait" techniques for deadlock prevention. More details.

2.3) Given the following schedule, draw the wait-for graph. **Is there a deadlock?**

T1	T2	T3	T4
Lock-S(A)			
R(A)			
	Lock-X(B)		
	W(B)		
Lock-S(B)			
		Lock-S(C)	
		R(C)	
	Lock-X(C)		
			Lock-x(B)
		Lock-x(A)	

2.4) Consider the following two transactions T1 and T2:

T1: R(X) W(X) R(Y) W(Y) Commit
T2: R(Y) W(Y) R(X) W(X) Commit

Show a schedule for T1 and T2 operations that leads to a deadlock when you use exclusive/shared locking (i.e., 3-state locking). **Draw the wait-for graph for your schedule.**

Question 3:

3.1) Explain how does the recovery manager ensure atomicity of transactions? More details.

3.2) Explain how does the recovery manager ensure durability of transactions? More details.

Question 4:

Given the following log, **show the steps** that are taken by the recovery manger to recover from the crash.

```
……..
<T1 start>
<T3 start>
<checkpoint {T1,T2,T3}>
<T1,p5,200,300>
<T2,p3,400,500>
<T2 commit>
<T3,p3,500,600>
<T1,p5,200>
<T1 abort>
XXCRASHXX
```

Question 5:

Given the following log, **show the steps** that are taken by the recovery manager to abort transaction T2.

```
<T2 start>
<T1,p2,100,200>
<T1,p1,50,60>
<T2,p5,100,200>
<T3,p4,100,200>
<T3 commit>
<T2,p5,200,300>
<T2,p3,200,300>
XXT2 CRASH XX
```