Practice 11 Concurrency Control:

Question 1

Consider the following arrival order of operations to the scheduler. If the scheduler uses strict two phase locking to schedule the operations, modify the table below to show the new schedule.

Ta	Tb	Tc
	Write(x)	
Read(y)		
	Read(z)	
		Read(x)
	Write(y)	
Write(x)		
	Read(x)	
	Commit	
		Write(z)
Commit		
		Commit

Answer:

Ta	Tb	Tc
	WriteLock(x);	
	Write(x);	
ReadLock(y);		
Read(y);		
	ReadLock(z);	
	Read(z);	
		ReadLock (x) → blocked
	WriteLock (y) → blocked	
WriteLock(x) → blocked		

It is a deadlock.

Question 2

The following table shows the schedule for transactions T_1 and T_2 with T_1 having an "older" time-stamp than T_2.

T_1	T_2
Read(a)	
	Read(b)
Write(b)	
	Write(a)
Commit	
	Commit

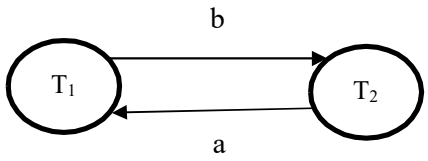
(1) Assume that Strict Two Phase Locking is used for concurrency control. Define the wait-for-graph.

(2) Show the new schedule if the wait-die method is used.
(3) Show the new schedule if the wound-wait method is used.

Answers:
(1) With S2PL, the schedule is

T_1	T_2
ReadLock(a); Read(a)	
	ReadLock(b); Read(b)
WriteLock(b) → blocked	
	WriteLock(a) → blocked

The wait-for-graph is:



(2) Wait-die: If TS(Ti) < TS(Tj), Ti waits else Ti dies

T_1	T_2
ReadLock(a); Read(a)	
	ReadLock(b); Read(b)
WriteLock(b) → blocked	
	WriteLock(a) → **aborted and restarted**
WriteLock(b)	
Write(b)	
Commit	
Unlock(a,b)	
	ReadLock(b); Read(b)
	WriteLock(a)
	Write(a)
	Commit
	Unlock(a,b)

(3)
Wound-wait: If TS(Ti) < TS(Tj), Tj wounds else Ti waits

T_1	T_2
ReadLock(a); Read(a)	
	ReadLock(b); Read(b)
WriteLock(b) → **Abort T_2**	
Write(b)	
Commit	
Unlock(a,b)	
	ReadLock(b); Read(b)
	WriteLock(a)
	Write(a)
	Commit
	Unlock(a,b)