True/False:

1. Berkeley's first CS Ph.D. (1969) Jim Gray won the Turing Award.
True.

2. A 2PL Schedule may lead to cascading abort, Strict 2PL can fix this.
True.

3. There was no downtime for GMail in 2010.
False. GMail had 99.984% availablity in 2010.

Short Answer:

1. What are anomalies with Interleaved Execution?
Answer:
Read-Write conflict (Unrepeatable reads)
Write-read conflict (reading uncommitted data)
Write-write conflict (overwriting uncommitted data)

2. What are the requirements for two transaction operations to conflict?
Answer:
– Belong to different transactions
– Are on the same data
– At least one of them is a write

3. Two schedules are conflict equivalent iff:
Answer:
– Involve same operations of same transactions
– Every pair of conflicting operations is ordered the same way

4. What are meanings of MTBF, MTTF, MTTR?
Answer: Mean time between (to) failure. Mean time to repair.
Total Outage duration (MTTR) = Time to Detect (need good monitoring) + Time to Diagnose (need good docs/ops, best practices) + Time to Decide (need good org/leader, best practices) + Time to Act (need good execution!)

Long Answer:

1. Consider the two transactions below. Assume each instruction (i.e., read, write, addition, subtraction) takes
one time unit, and acquiring/releasing a lock takes zero time units. Once a transaction acquires a shared
lock it cannot upgrade it to an exclusive lock, and once a transaction acquires an exclusive lock it

cannot
downgrade it to a shared lock..
Transaction1 Transaction2
R(A); R(A);
A = A + 100; A = A – 50;
W(A); W(A)
R(B);
B = B – 100;
W(B);

a) What is the minimum possible execution time taken by both transactions when using 2PL (2 phase locking)? Show a schedule that achieves the minimum time. The diagram below shows the first several instructions executed by each transaction for such a schedule. Note that Transaction 2 is not getting the lock when requesting it, instead, Transaction2 needs to wait for the lock to be released by Transaction 1.

6 time units.

Transaction 1 Lock_X(A) <granted>	Transaction 2
R(A)	Lock_X(A)
A = A + 100	
W(A)	
Lock_X(B)<granted>	
Unlock(A)	<Lock_X(A)granted>
R(B)	R(A)
B = B-100	A = A – 50;
W(B)	W(A)
Unlock(B)	Unlock(A)

b) What is the minimum possible execution time taken by both transactions when using strict 2PL? Show a schedule that achieves the minimum time.

Answer: 9 time units.

Transaction1	Transaction2
Lock_X(A) <granted>	
R(A)	Lock_X(A)
A=A+100	
W(A)	
Lock_X(B)<granted>	
R(B)	
B=B-100	
W(B)	
Unlock(A)	<Lock_X(A)granted>
Unlock(B)	R(A)
	A=A-50
	W(A)
	Unlock(A)

c) Repeat question (a), assuming Transaction1 is replaced with transaction:
Transaction1'
R(A);
R(B);
A = A + 100;
B = B – 100;
W(A);
W(B);

Answer: 8 time units

Transaction1'	Transaction2
Lock_X(A) <granted>	
R(A)	Lock_X(A)
Lock_X(B) <granted>	
R(B)	

A=A+100	
B=B-100	
W(A)	
Unlock(A)	<Lock_X(A)granted>
W(B)	R(A)
Unlock(B)	A=A-50
	W(A)
	Unlock(A)

d) Now assume Transaction2 is replaced with transaction:
Transaction2'
R(B);
B = B + 50;
W(B);
R(A);
A = A – 50;
W(A);
Is it possible to get into deadlock by executing Transaction1 and Transaction2' ? If not, use no more than two sentences to explain why not? If yes, give an example of schedule leading to deadlock.

Answer: Yes.

Transaction1	Transaction2'
Lock_X(A) <granted>	Lock_X(B) <granted>
R(A)	R(B)
A=A+100	B=B+50
W(A)	W(B)
Lock_X(B)	Lock_X(A)
deadlock	deadlock