Q1)

(a) Yes. Because in no case does conflict occurs. Since transactions are concurrent and unlocks are after commits, in any case, one transaction will have to wait for the other transaction.

(b) Yes. T2 aborts. When I started with T1, exclusive lock for A (X(A)) will be used. Then this transaction wants to use exclusive lock for B (X(B)). If T2 used it before, T1 need to wait for it. Also, same situation occurs for T2. To solve this deadlock, T2 would be aborted under the wait-die deadlock prevention scheme.

(c) No. Because, when a transaction used an exclusive lock for A or B and write something on it, the other transaction cannot read this value until the transaction which has written commits. In this situation all unlocks are after commit. So, cascading rollback is not possible.

(d) Yes. Because, locks and unlocks are used in a such way that one transaction cannot make any progress (read or write) until the transaction, which used the exclusive lock for A(X(A)) first, has finished its progress and unlocked.

(e) No. Because one transaction can finish its progress before the other one in any situation.

(f) Yes.

T1	T2
X1(A)	
R(A)	
W(A)	
X1(B)	
U1(A)	
R(B)	
W(B)	
U1(B)	
	X2(A)
	X2(B)
	R(B)
	R(A)
	W(A)
	U2(A)
	W(B)
	U2(B)
ABORT	

Time ↓

In this situation, cascading rollback is possible.

Q2)

(a)

i.

Operation	A			B			C		
	RTS	WTS	C	RTS	WTS	C	RTS	WTS	C
Initially	0	0	True	0	0	True	0	0	True
R1(A)	1	0	True						
R2(B)				2	0	True			
R3(A)	3	0	True						
w1(A)									
T1 aborts									
R2(C)							2	0	True
w3(B)				2	3	False			
w2(C)							2	2	False
R2(A)	3	0	True						
w3(C)							2	3	False
T3 commits				2	0	True	2	2	True
w2(B)				2	2	False			
T2 commits				0	0	True	0	0	True

[3]

ii.

Operation	A			B			C		
	RTS	WTS	C	RTS	WTS	C	RTS	WTS	C
Initially	0	0	True	0	0	True	0	0	True
R1(A)	2	0	True						
R2(B)				3	0	True			
R3(A)	2	0	True						
w1(A)	2	2	False						
R2(C)							3	0	True
w3(B)									
T3 aborts									
w2(C)							3	3	False
T1 commits	0	0	True						
R2(A)	3	0	True						
w2(B)				3	3	False			
T2 commits	0	0	True	0	0	True	0	0	True

(b)

The commit bit is very important. Because without using commit bit, we cannot avoid cascading aborts Commit bit makes it recoverable, and it can avoid cascading aborts.

[4]

Q3) (a) Log:

LSN	transID	prevLSN	type	pageID	log entry	undoNextLSN
1	T1	-	Update	P1	write A (A→A1)	-
2	T2	-	Update	P1	write B (B→B2)	-
3	T2	2	Update	P2	write C (C→C3)	-
4	T2	3	Abort	-	-	-
5	T2	-	CLR	-	Undo T2 LSN3	2
6	T2	-	CLR	-	Undo T2 LSN2	-
7	-	-	Checkpoint	-	-	-
8	T2	6	End	-	-	-
9	T3	-	Update	P2	write D (D→D4)	-
10	T1	1	Commit	-	-	-
11	T1	10	End	-	-	-
12	T4	-	Update	P1	write A (A1→A5)	-
13	T3	9	Update	P1	write B (B→B6)	-
14	T4	12	Commit	-	-	-

Transaction Table:

transID	lastLSN	status
T4	14	Committed
T3	13	Running

Dirty Page Table:

PageID	recLSN
P1	12
P2	9

Pages in Memory:

- P1: A has value A5, B has value B6. The PageLSN is 13.
- P2: C has its initial value. D has value D4. The PageLSN is 9.
- Page P1 is flushed to the disk with PageLSN = 1.

(b)

Transaction Table

transID	lastLSN	status
T4	14	Committed
T3	13	Running

Dirty Page Table

PageID	recLSN
P1	1
P2	9

(c)

Redo phase starts with LSN 1 because the minimum recLSN in dirty page table of Analysis Phase is 1. However, it skips over LSN 7 without loading P1 from disk to check that the change need not be applied.

LSN 1 - 7 : Skipped
LSN 8 : Skipped
LSN 9 : Redone
LSN 10 : Skipped
LSN 11 : Skipped
LSN 12 : Redone
LSN 13 : Redone
LSN 14 : Skipped

Pages in Memory:

- P1 : A has value A5, B has value B6. The PageLSN is 13
- P2 : C has its initial value. D has value D4. The PageLSN is 9.

(d) Log :

LSN	transID	prevLSN	type	pageID	log entry	undoNextLSN
1	T1	-	Update	P1	write A (A→A1)	-
2	T2	-	Update	P1	write B (B→B2)	-
3	T2	2	Update	P2	write C (C→C3)	-
4	T2	3	Abort	-	-	-
5	T2	-	CLR	-	undo T2 LSN3	2
6	T2	-	CLR	-	undo T2 LSN2	-
7	-	-	Checkpoint	-	-	-
8	T2	6	End	-	-	-
9	T3	-	Update	P2	write D (D→D4)	-
10	T1	1	Commit	-	-	-
11	T1	10	End	-	-	-
12	T4	-	Update	P1	write A (A1→A5)	-
13	T3	9	Update	P1	write B (B→B6)	-
14	T4	12	Commit	-	-	-
15	T4	14	End	-	-	-
16	T3	-	CLR	-	undo T3 LSN 13	9
17	T3	-	CLR	-	undo T3 LSN 9	-
18	T3	17	End	-	-	-

Pages in Memory:

- P1 : A has value A5. B has its initial value. The PageLSN is 16.
- P2 : C has its initial value. D has its initial value. Th PageLSN is 17.