I. Find an **interleaving** of the steps for the two transactions below that will produce an incorrect result.

T1:		***T2:***	
1.1	READ(A,s)	2.1	READ(A,t)
1.2	s = s+1	2.2	t = t*1.01
1.3	WRITE(A,s)	2.3	WRITE(A,t)
1.4	READ(B,s)	2.4	READ(B,t)
1.5	s = s-1	2.5	t = t*1.01
1.6	WRITE(B,s)	2.6	WRITE(B,t)

II. Draw precedence graphs for the schedules below. Then decide which are conflict-serializable.
(Note: operations are read, write, and commite; subscripts identify the transaction.)
From Michael Böhlen, https://files.ifi.uzh.ch/dbtg/dbs/FS17/ex12/sol12.pdf

A. $S_1 = r_3(A), r_2(B), w_2(B), r_1(B), w_3(C), c_3, r_2(D), c_2, r_1(D), c_1$

B. $S_2 = w_3(D), r_1(A), w_2(D), w_2(A), w_3(A), c_3, w_1(C), r_2(D), c_2, c_1$

C. $S_3 = r_1(X), w_2(X), r_2(x), W_3(x)$

III. Locking. Is there an interleaving of these transactions that respects locking but isn't serializable?

T1:		T2:	
1.1	LOCK(A)	2.1	LOCK(A)
1.2	READ(A,s)	2.2	READ(A,t)
1.3	s = s+1	2.3	t = t*1.01
1.4	WRITE(A,s)	2.4	WRITE(A,t)
1.5	UNLOCK(A)	2.5	UNLOCK(A)
1.6	LOCK(B)	2.6	LOCK(B)
1.7	READ(B,s)	2.7	READ(B,t)
1.8	s = s-1	2.8	t = t*1.01
1.9	WRITE(B,s)	2.9	WRITE(B,t)
1.10	UNLOCK(B)	2.10	UNLOCK(B)

IV. Timestamps. What is the outcome of the following schedule? Keep track of the metadata.

$TS(T_1)$	$TS(T_2)$	$TS(T_3)$	RT(A)	WT(A)	C(A)	RT(B)	WT(B)	C(B)	RT(C)	WT(C)	C(C)

$S = r_1(A), r_2(B), r_3(C), w_2(B), r_1(B), r_3(B), r_2(C), w_2(C), w_1(A), w_2(B), w_3(C), c_1, c_2, c_3$