1. Consider the following sequence of log records representing the actions of one transaction T:

   < START T >;< T,A, 10 >;< T,B, 20 >;< T,C, 30 >;< COMMIT T >;

   Tell all the sequences of events that are legal according to the rules of UNDO logging, where the events of interests are the writing to disk of the blocks containing database elements, and the blocks of the log containing the update and commit records. You may assume that log records are written to disk in the order shown, i.e., it is not possible to write one log record to disk while a previous record is not written to disk.

2. The following is a sequence of undo-log records written by two transactions T and U:

   < START T >; < T,A,10 >; < START U >; < U,B,20 >; < T,C,30 >; < U,D,40 >;
   < COMMIT U >;< T,E,50 >; < COMMIT T >

   Describe the action of the recovery manager, including changes to both disk and the log, if there is a crash and the last log record on disk is:
   a. < START U >
   b. < COMMIT U >
   c. < T,E, 50 >
   d. < COMMIT T >
   For each of the above situations, what values written by T and U must appear on disk (you need only to specify the variable)? Which values might appear on disk? Consider both before and after the recovery.

3. Repeat Question 2 for Redo logging.

4. (2021 exam question) Consider a DBMS that uses **undo/redo logging** with **non-quiescent checkpoints** for recovery and uses the **basic two-phase locking protocol** (i.e., 2PL with the locking rules 1, 2 and 3 as described in our lecture) for concurrency control. Suppose the actions of three transactions T_1, T_2 and T_3 are interleaved resulting in the following sequence of log records just right before a system crash.

   < T_1, Start >
   < T_2, Start >
   < T_1, X, 0, 10 >
   < T_2, ?, ?, 20 > ⇐ **incomplete log record IC1**
   < Checkpoint Start>
   < ?, Y, ?, 30 >   ⇐ **incomplete log record IC2**
   < T_3, Start>
   < T_1, Y, 30, 40 >
   < Checkpoint End>
   < T_1, commit >
   < T_3, X, ?, 100 >   ⇐ **incomplete log record IC3**
   SYSTEM CRASH AT THIS POINT !!!!!

In the sequence, the log entries follow the format <Transaction ID, Object, Old Value, New Value>. Suppose the database has only two objects X and Y, and the initial values of these objects are both 0 prior to the start of the log file shown above. Moreover, three of the log records are incomplete in the sense that some information are missing. For example, for incomplete log IC1, the object and its old value are missing.

A. Suppose it is possible for a transaction to write an object multiple times (each generating a log record). Identify the missing values of the incomplete log records. You should identify all possible combinations of IC1, IC2 and IC3 that result in a correct log file. Complete your answer in the table below. The first column shows an example (not necessarily a correct answer): suppose a correct sequence of logs should include IC1 being $< T_2$, **X, 0**, 20 $>$, IC2 being $< T_1$, Y, **0**, 30 $>$, and IC3 being $< T_3$, X, **0**, 100 $>$, then fill in "**X, 0**" in the first row of the column, "**T_1, 0**" in the second row, and "**0**" in the third row. Note that while the table allows **up to** 4 possible answers, this **does not mean** that there are 4 answers.

	SAMPLE	Answer I (Log I)	Answer II (Log II)	Answer III (Log III)	Answer IV (Log IV)
IC1	X, 0				
IC2	T_1, 0				
IC3	0				

B. Based on your possible log files in the previous part, list all possible values of A and B on disk before recovery. Complete your answer in the table below. Again, the number of possible answers depend on your answer in (A) (though 4 are given below).

	X	Y
Log I		
Log II		
Log III		
Log IV		