**1.** Consider the following transaction log from the start of the execution of a DBMS instance, which is performing undo/redo logging with checkpointing:

```
1.  <START T1>
2.  <T1, A, 17, 22>
3.  <START T2>
4.  <START T3>
5.  <T2, C, 31, 42>
6.  <T3, E, 14, 9>
7.  <T1, B, 12, 13>
8.  <T1, A, 8, 17>
9.  <T2, D, 3, 6>
10. <COMMIT T1>
11. <START T4>
12. <T4, F, 5, 7>
13. <T2, D, 1, 3>
14. <START CKPT (T2, T3, T4)>
15. <T2, C, 2, 31>
16. <T4, F, 16, 5>
17. <START T5>
18. <T3, E, 15, 14>
19. <T5, G, 30, 29>
20. <COMMIT T2>
21. <END CKPT>
22. <T5, G, 32, 30>
23. <ABORT T3>
24. <T4, H, 11, 22>
25. <COMMIT T5>
26. <COMMIT T4>
```

Assume that the update log entries are of the format `<transaction, data item, new value, old value>`.
Also consider the following possible failure scenarios:
(1) The system crashes just before line 9 is written to the disk.
(2) The system crashes just before line 16 is written to the disk.
(3) The system crashes just before line 20 is written to the disk.
(4) The system crashes just before line 21 is written to the disk.

What are the values of data items A, B, C, D, E, F, G, and H on the disk after successful recovery in each of the above scenario.

**2. A.** Undo logging requires that before an item X is modified on disk the log records pertaining to X are in the disk. (This is the WAL rule.) Show using an example that an inconsistent database may result if log records for X are not output to the disk before X.
**B.** Redo logging: Show using an example that an inconsistent database may result if some items are written on the disk before the commit is written on the log, even if WAL holds.
**C.** Undo logging: Show using an example that an inconsistent database may result if some items are not written in the disk by the time the commit is written. Assume that WAL holds.

Your examples should involve a crash and should clearly show (I) the write actions of the transaction, (II) the state of the log and the database at the time of the crash, and (III) why successful recovery can not be accomplished.

3. Consider a transaction that performs the following actions in the given order

1. Write object A
2. Write object B
3. Commit

Decide whether each one of the following snapshots of the database and the log are possible or impossible at any point during or after the end of the transaction. The snapshots refer to data that are actually in the disk. Assume that if we do not mention explicitly that something is stored on the disk then it is not stored on the disk. For each item give an answer for both Undo and Redo logging.

Snapshots:
1. Log has entry for write(A) and the database has the new value of A
2. The database has the new value for A
3. The log has entries for write(A) and write(B) and also has the commit entry
4. The log has entries for write(A) and write(B). The database has the new values for A and B
5. The log has entries for write(A), write(B), and the commit entry. The database has the new value for A
6. Log has entry for write(A) and commit and the database has the new value of A

4. Consider the following sequence of log records for undo logging:

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

Suppose the last log record that appears on disk at the time of a crash is COMMIT U . What will the recovery manager do to recover from this crash, in terms of updates to the disk and to the log?

5. Consider the following sequence of log records for undo/redo logging:

<START S>
<S,A,60,61>
<COMMMIT S>
<START T>
<T,A,61,62>
<start checkpoint here>
<START U>
<U,B,20,21>
<T,C,30,31>
<START V>
<U,D,40,41>
<V,F,70,71>
<COMMIT U>
<T,E,50,51>
<COMMIT T>
<V,B,21,22>
<COMMIT V>

What is the initial state of the database? Which transactions must be indicated in the checkpoint start? Recall that there is flexibility in when the corresponding end checkpoint record occurs. Circle each log record that can be immediately followed by the corresponding end checkpoint record.