Warmup #1

Explain why log records for transactions on the undo-list must be processed in reverse order, whereas redo is performed in a forward direction.

Problem 2

Consider a database with the following initial values, and the attached command log:

\[ A = 41, \quad B = 66, \quad C = -2, \quad D = 104, \quad E = 23 \]

LOG:

\[
\begin{align*}
&T_0, \text{start} > \\
&T_1, \text{start} > \\
&T_1, B, 66, 102 > \\
&T_2, \text{start} > \\
&T_2, C, -2, 99 > \\
&T_1, B, 102, 142 > \\
&T, \text{checkpoint} : T_0, T_1, T_2 > \\
&T_0, A, 41, 100 > \\
&T_3, \text{start} > \\
&T_2, \text{commit} > \\
&T_3, D, 104, -40 > \\
&T_3, \text{commit} > \\
&T_4, \text{start} > \\
&T_4, E, 23, 24 > \\
\end{align*}
\]

Assume immediate database modification and the system crashes before the remaining transactions can commit. Use the recovery protocol for concurrent transactions (which persists all in-memory dirty pages and transaction log entries at each checkpoint) to answer the following questions.

1. List any transactions that will need to be undone or redone in the recovery process.

2. List, in order, the set of logged operations to be performed to undo or redo the transactions, using the recovery protocol for concurrent transactions. (i.e. “Set A to 7”, “Set B to 39”, etc.)

3. Give the final values for \( A, B, C, D, \) and \( E \) after the recovery.