Describe your implementation including any design decisions you made. Make sure to emphasize anything that was difficult or unexpected.

Setup:
Prior to getting started with lab 4, we first had to make some changes to or code. First, we had to make sure that any time an update was made that this update was being kept track of by the log file, tracking the before image of the page prior to the update. Then, we had to slightly change our transactionComplete() function so that all updates done by a committed transaction should also update the before image of all the updated pages.

EX1 LogFile.rollback():
Initially, the hardest part about implementing the exercises for the lab was to first understand the formatting of the logs in the log file. Once we understood the formatting, implementation wasn't too bad. For rollback, we decided to loop through the current log file and for each type of record in the log file we perform a different action. If we come across a begin or commit record, then we don't do anything, since we are only looking to rollback any of the updates made by rolled back transactions. If we come across an abort record, then we stop going through the log file because that is where the transaction stopped. When we come across an update record, we check to see that the id of the transaction updating this page matched the id of the transaction being rolled back and that we don't already have the before image of the page in our list, if these two conditions pass, then we add this page to our list of pages to be rolled back. Once we have our list of pages to be rolled back, we iterate through the list and discard the current images of the pages and insert back in the before image, prior to the updates, of the page. This allows us to perform a rollback on any of the updates made by an aborted/rolled back transaction.
Ex2: logFile.recover():
First, we decide to loop through the log file from the last check point and for each transaction id, we put it into the tidToFirstLogRecord. We simply only read from last checkout because we knew any update of committed/aborted transactions before that checkout already made it to disk. While reading the log from the last checkout to the end, if we come across checkpoint_record, then we should throw exception because we only read from the last check point, there is no other checkpoint after that. If we come across begin record, we add it into our tidToFirstLogRecord if it is not present there, otherwise, throw inconsisten log because there are two begin record for the same transaction id. If we come across abort record, simply just roll back the transaction and remove it out of tidToFirstLogRecord. If we come across update record, first we need to check that its transaction id is in tidToFirstLogRecord, then we produce it update. After produce all the log record in the log file from the last checkpoint, we will left with the transactions that were executing when system crash, we need to abort all of its updates, so we just need to call roll back with its transaction id. We added a constructor for TransactionId class with long integer transaction id, so we pass it in roll back function.

Discuss and justify any changes you made outside of LogFile.java.
We did not need to make any changes outside of LogFile.java other than for setup as discussed above.

Describe a unit test that could be added to improve the set of unit tests that we provided for this lab.
A unit test that I would suggest adding would be a test that had 4 transactions, with two updating one page, T1 and T2, and the other two updating another, T3 and T4. Have the test abort and rollback transactions T2 and T4 and commit transactions T1 and T3. Finally, check to see that the updates to pages made by the committed transactions T1 and T3 were recorded and the updates to pages made by T2 and T4 were rolled back. I think this would be a good unit test to add since the tests only consider at most 3 concurrent transactions doing updates separately, rather than having 4 or more concurrent transactions.

If you have any feedback for us on the assignment, you can add it to the writeup, send us an email, or make a note and paste your note in the course evaluation form at the end of the quarter.