Homework_3 Lab solutions:

A. COMMIT or ROLLBACK:
   7. Updated value is lost because of ROLLBACK (Abort) caused due to system failure.

B. FAILURE of TRANSACTION with SAVEPOINT:
   5. The updated value is lost. The savepoint did not save the update.

C. COMMIT, ROLLBACK and SAVEPOINT:
   10. The restoration is successful. The votes value doesn’t change because the savepoint was created after the votes were updated.
   11. Oracle doesn’t support roll forward.
   14. Oracle doesn’t support rollback after commit.

D. CONCURRENCY CONTROL: ISOLATION LEVELS & IMPLICIT DATA LOCKING
   6. Score is (10).
   7. The query doesn’t hang. Partner sees the original value.
   8. The query hangs. Over-writing uncommitted transaction generates error.
   11. Yes. The updates can be seen in the partner’s window.
   12. Oracle follows 2PL.
   17. Partner2 doesn’t see the updated value even after Partner1 has committed.
   18. Default Isolation Level: Read Committed. (A transaction may read only data that has been committed in the database.)

E. CONCURRENCY CONTROL: EXPLICIT DATA LOCKING
   5. Yes.
   8. Oracle doesn’t allow the update because SELECT FOR UPDATE allows to lock the records which will not be released until partner1 commits.
   11. The query doesn’t hang. It gets aborted. SELECT FOR UPDATE cant be executed as the lock is already held by partner.

F. NON-REPEATABLE READ & PHANTOM READ
   7. Partner gets different result. Partner reads data that is already committed.
   Non-repeatable reads and phantom reads can be prevented by changing the isolation level to serializable.

G. BIG PICTURE
   1. No. Even though another user is executing a transaction, updates can be made by the current user as long as his transaction does not access the data items locked by the other user.
   2. Yes. read/write locks are acquired for the data being updated by a transaction. Another transaction cannot see the updates made by the other uncommitted transaction.
3. Yes. The locks are acquired in one phase and are release when the transaction commits. When a lock is released no more locks are acquired. Hence 2PL is adhered.
4. Yes. As the records being modified are only locked, another transaction can insert a new row and a SELECT statement can return two different views when executed twice.
5. Yes. A transaction can wait for the lock on particular data items before starting execution. This is can result in deadlock and preventive steps should be taken.
6. Yes. Livelocks should be avoided by setting priorities based on age.
7. No. As the transaction changes are not seen by the other transactions until committed, no two transactions are dependent on each other avoiding cascaded rollbacks.
8. Yes. Failure of a transaction never effects another transaction in Oracle. Hence Oracle has a recoverable database CCT.