Question A:

```
void threadT31Func(void){
    char Ba[20];
    const char* threadName = "T31";
    PE-0 (pthread_setname_np (pthread_self(), threadName));
    REPORT2 (myReadcond(daSKPtr[0], Ba, 6, 6, 0, 0));
    int RetVal = myReadcond(daSKPtr[0], Ba, 6, 6, 0, 0);
    cout << "RetVal in T31: " << RetVal << " " << Ba << " " << sterror << endl;
}
void threadT41Func(void){
    ...
    thread T31. join();
    ...
}
```

1> E. c.d are correct

c) thread T31 (45, thread T31)
   threadT31. detach()

d) void threadT41Func(void){
       thread T31. join()
   }

2) E. None of above

3) C. 6
   myReadcond(daSKPtr[0], Ba, 6, 6, 0, 0));

4) D. "ijKl"

5) B. myReadcond() didn't need to change errno

6) D. during myClose(datSKpt[0]) call in T32
   When does threadT42Func() unblocks after being blocked in the first myTcdrain() call, if T42 does unblock from that call

7) B. resumes b/w the 2nd setSchedPrio(40) and "T32" starting & ending.
   When does T42 resume after being blocked in the first myTcdrain() call, if T42 does unblock from that call?

8) -1

myWrite (daSktPr[1])

9) broken pipe.

10) B. -1

myClose (daSktPr[1])

11) EBADF

12) B. No, wait() is used

Can socketLK in member func drainning be of class lock_guard <mutex> instead of unique_lock < mutex> w/o changing ?

13) E. Yes, unique_lock can do everything lock_guard does.

unique_lock → lockguard.

14) D. Yes, it is unlocked during the call to the wait() if called.

If socket LK in draining ever unblock?

15) A. No,

If socket LK in writing ever unblock?

16) B.

17) A