State if the following statements are TRUE (T) or FALSE (F)

1. Two concurrent global transactions, T1 and T2, consist of the following:
   \[ T1 = [r1(x1), w1(x1), r1(y1), w1(y1)] \]
   \[ T2 = [r2(x2), w2(x2), r2(y2), w2(y2)] \]
where \( ri(x) \) and \( wi(x) \) denote a read and a write operation by transaction \( i \) on data item \( x \); data items \( x1 \) and \( x2 \) are stored at site A, while \( y1 \) and \( y2 \) are stored at site B; and \( y1 \) is a replica of \( x1 \), and \( y2 \) is a replica of \( x2 \). In addition, there are two local transactions, L3 and L4:
   \[ L3 = [r3(x1), r3(x2)] \]
   \[ L4 = [r4(y1), r4(y2)] \]
executing concurrently with \( T1 \) and \( T2 \).
Consider the following two schedules produced by the local schedulers at Site A:
   \[ SA1 = [r1(x1), r3(x1), r2(x2), w1(x1), w2(x2), r3(x2)] \]
   \[ SA2 = [r1(x1), w1(x1), r3(x1), r2(x2), r3(x2), w2(x2)] \]
Consider the following two schedules produced by the local schedulers at Site B:
   \[ SB1 = [r4(y1), r2(y2), r1(y1), w1(y1), w2(y2), r4(y2)] \]
   \[ SB2 = [r1(y1), w1(y1), r2(y2), r4(y1), r4(y2), w2(y2)] \]

a) SA1 is locally serializable. [T]
b) SB1 is locally serializable. [T]
c) SA2 and SB1 will produce a globally serializable schedule. [F]

2. Consider hierarchical 2PC with the hierarchy shown in the figure below. Here, node 1 acts as global coordinator; node 2 acts as local coordinator for nodes 4-6, and as participant for node 1; node 3 acts as local coordinator for nodes 7 and 8, and as participant for node 1.

   ![Diagram](image)

Count the number of messages and rounds needed to commit a transaction (ignore DONE messages). Number of messages = 14; number of rounds = 2. [F]
3. In a hierarchical deadlock detector, each deadlock will only be detected at one detector.  

4. It is possible to detect a false deadlock in distributed deadlock detection scheme (i.e., a deadlock is detected at a site, but it turns out to be otherwise.)  

5. Given the following wait-for information:

   Local wait-for graphs:
   
   Site A:  \( T_1 \rightarrow T_2 \)  
   Site B:  \( T_4 \rightarrow T_2 \)  
   Site C:  \( T_3 \rightarrow T_4 \)  
   \( T_2 \rightarrow T_3 \)  
   \( T_6 \rightarrow T_4 \)  
   \( T_5 \rightarrow T_3 \)  
   \( T_7 \rightarrow T_8 \)  
   \( T_8 \rightarrow T_7 \)  

   Intersite wait-for graphs (:\( T_3^A \) means sub-transactions of \( T_3 \) at site A, etc) :
   
   \( T_3^A \rightarrow T_3^C \)  
   \( T_2^B \rightarrow T_2^A \)  
   \( T_7^C \rightarrow T_7^B \)  
   \( T_4^C \rightarrow T_4^B \)  
   \( T_8^B \rightarrow T_8^C \)  

   a) There are 3 cycles in the global wait-for graphs.  
   b) To break any of the deadlocks, you should not roll-back \( T_5 \).  

6. To generate tasks using the symmetric fragment-and-replicate strategy for parallel join processing, we must always range-partition the two participating relations in the same manner, i.e., if \( a_1, a_2, \ldots, a_n \) is the partitioning vector for relation \( R \), then relation \( S \) must also use the same partitioning vector.  

7. Attribute value skew may give rise to load imbalance in parallel join processing. Absolutely nothing can be done to handle such skew.