1. (Solution and Grading by: Bo Teng)
   Name 10 Processes as P1, P2, ......, P10
   Round 1: P1 sends its value, v1 to P2 and then fail,
   Round 2: P2 sends v1(among other received values) to P3, and then fail
   Now P3 has v1, and P4-P10 does not have v1. P3 and P4-10 end up having a different
   set of values, and safety will be violated.

2. (Solution and Grading by: Si Liu)

3. (Solution and Grading by: Hongwei Wang)
4. (Solution and Grading by: Bo Teng)
   a. No. The protocol is incorrect (not clear what happens when two leaders are elected). The system might get stuck. An under-specified protocol cannot be called live.
   b. No. Two or more leaders may be elected as each gets 21 votes, and this may violate safety.
   c. The algorithm would be faster because fewer processes will need to respond and time is bounded by the slowest process. (But it’s still not correct.) (Full credit is also granted if you argue that the algorithm would be slower because the algorithm need to run more rounds to reach consensus.)

5. (Solution and Grading by: Sili Hui)
   Process snapshots occur at the first point they send/receive a Marker message. All 6 channel states are empty EXCEPT: d, and f, which appear on their respective channel states (C(EM), C(NM)).

6. (Solution and Grading by: Sili Hui)
   a. Route Election message to id 0 (via Chord routing), then have it forward message to immediate predecessor. That node is the new leader, which can then multicast an Elected message to everyone informing them that it is the new leader. Multicast is also done via the Chord finger tables, in a manner similar to Gnutella Query spread (forwarded only once, but no TTL)
   b.  
      i. Safety: Since finger tables are correct, the highest id process is the only one who receives the Election message finally and sends an Elected message  
      ii. Liveness: Liveness works because there are no failures and Chord routing eventually delivers messages
   c.  
      i. Completion time: O(logN) round trip time  
      ii. # of msgs: O(N*logN) since the finger tables are used as multicast links, and each process forwards at most once along each of its O(log(N)) Chord links.
   d. When failure happens, election msg may not route to leader. When finger table entries are inconsistent, may elected the wrong leader. Other related cases are accepted as well.

7. (Solution and Grading by: Xiaocheng Yuan)
   The algorithm uses 3 types of messages: ELECTION, OK, and COORDINATOR.
   Algorithm fails if N < k.
   It operates as follows:
   1. When a process detects a failure among any of the existing leaders, it sends ELECTION messages to all processes.
2. When a process receives an ELECTION message, it sends OK messages to all processes with higher ids than it, and then wait for OK messages for a period of time.

3. After the OK reception period is over, all processes who received greater than or equals to k OK messages, will standby with a timeout to wait for COORDINATOR messages.

4. After the OK reception period is over, all processes who received less than k OK messages claim self as one of the leaders, sends out COORDINATOR messages to all processes(including itself).

5. Any process that does not receive exactly k COORDINATOR messages re-initiates an election.

Since OK messages are only sent to processes with lower priority, after a period of OK waiting time, OK messages would have been disseminated properly with high priority processes each receiving less than k OK messages, therefore guaranteeing safety.

If potential leaders failed during the execution of the algorithm, the number of COORDINATOR messages will be incorrect, and election will be re-initiated.

8. (Solution and Grading by: Xiaocheng Yuan)

**Safety**: Safety is violated with this change in the timestamp used. Suppose there are three processes a, b, and c, the scenario below would cause both a and b to enter the CS at the same time.

1. a sends request with (5, a)
2. b replies to a because it has no outstanding requests
3. b sends request with (0, b)
4. a replies to b because (0, b) < (5, a)
5. c, who has no outstanding requests, replies to both a and b, and a and b enter the CS at the same time

**Liveness**: Liveness will be maintained with this change. Despite the timestamp being changed from Lamport to FIFO, once all other processes have finished making their CS requests, the waiting process's request will be fulfilled. If all processes start with the same counter, no one will need to wait for longer than N-1 other critical sections given they all make requests. Even if there's a difference between the process with the highest counter and the lowest counter, say k, the high counter process would not need to wait longer than (N-1)*(K+1) rounds.

**Ordering**: Causal ordering may be disobeyed. If two processes P1 and P2 concurrently send requests where P1’s request causally occurred before P2’s request but P2 happens to have a lower FIFO counter than P1 (because P2 has executed fewer events at itself than P1) then P2 will get access first.

9. (Solution and Grading by: Shiv Verma)

a. Incorrect. M32 was sent in V11 but received in V12
b. Correct. Every correct process received the same set of multicasts and everything in the view stayed in the view.
c. Incorrect. M32 was delivered by p1 in V11 but by p2 in V12.

d. Incorrect. P1 and p2 never delivered their own multicasts. Thus their sets of received multicasts in V11 differ.

e. Correct. All non-faulty processes received the same set of multicasts.

f. Incorrect. M32 sent in V11, but delivered in the next view.

10. (Solution and Grading by: Shiv Verma)

a)

No, the algorithm is not correct. The algorithm waits too long to capture channel state, and so it might capture more messages than necessary in the snapshot. In the above example, M01 will also get recorded when P1 receives the last marker message from P2. Ideally only M21 should be recorded in the snapshot.

b) Two changes are required:

i) Remove “if this is the (N-1)th marker being received at Pi”, the following for loop, and the “else do nothing clause”.

ii) Change the “Starts recording the incoming messages on each of the incoming channels at Pi: C_j (for j=1 to N except i)” to “Starts recording the incoming messages on each of the incoming channels at Pi: C_j (for j=1 to N except i and except the channel that the first marker was just received on)”. This is a minor fix (relevant to the fix in lecture slides).

c) Indy suggests your first words be: “One small step for a man, a giant leap for Lamport timestamps.” (All other answers accepted).