1. Totally-ordered multicast (5pt)

2. Distributed system (5pt)

3. Availability, Reliability (5pt)

4. Omission failures (5pt)

5. Bully Algorithm (5pt)
6. close-to-open cache consistency (5pt)

7. Primary-backup protocols (5pt)

8. Secure channals (5pt)

9. Cloud computing (5pt)

10. MapReduce (5pt)
1.  1) Explain the values for the finger table of node 9 in the following Chord DHT-based system. 
2) Assume node 4 is requested to look up key 29. How is this key resolved? You must explain your answer! 

2.  Explain the principle of an epidemic protocol. (5pt)
3. Why is the following data store not sequentially consistent? Is it causally consistent? Be sure to explain your answer. (5pt)

4. What is a $k$ fault-tolerant group, and how does $k$ depend on failure semantics? (5pt)

5. Explain the basic Paxos protocol (10pt)
6. Explain which functions are implemented in the runtime library of an RPC system. See also the following figure. (5pt)
7. When using vector clocks for enforcing causally ordered multicasting, $V C_i[i]$ is incremented only when process $P_i$ sends a message, and sends $V C_i$ as a timestamp $t s(m)$ with message $m$. How should we interpret the following two conditions for delivering $m$ when received by process $P_j$? 

1) $t s(m)[i] = V C_j[i]+1.$

2) $t s(m)[k] \leq V C_j[k]$ for $k \neq i.$

8. For each of the following applications, do you think at-least-once semantics or at most once semantics is best? Discuss. 

(a) Reading and writing files from a file server.

(b) Compiling a program.

(c) Remote banking.