1. What is a deadlock?

Deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.

2. When a deadlock occurs, will the system remain in the deadlocked state permanently or just temporarily for a limited amount of time?

Deadlock is a state that occurs in a multitasking (muti-threading) environment when two or more processes go into a loop waiting for resources forever.

3. Can deadlock happen if there is only one process? why?

One process cannot hold a resource, yet be waiting for another resource that it is holding. So it is not possible to have a deadlock involving only one process.

4. What is the difference btw the coding in Fig.3-2a and Fig.3-2b?

5. Consider Fig. 3-4. Suppose that in step (0) *C* requested *S* instead of requesting *R and S*. Would this lead to deadlock?

Neither change leads to deadlock. There is no circular wait in either case.

6. Students working at individual PCs in a computer laboratory send their files to be printed by a server which spools the files on its hard disk. Under what conditions may a deadlock occur if the disk space for the print spool is limited? How may the deadlock be avoided?

Students working at individual PCs in a computer laboratory send their files to be printed by a server that spools the files on its hard disk. Under what conditions may a deadlock occur if **the disk space for the print spool is limited**? How may the deadlock be avoided? deadlock.

7. In the preceding question (Question 6) which resources are preemptable and which are nonpreemptable?

The printer is nonpreemptable; the system cannot start printing another job until the previous one is complete. The spool disk is preemptable; you can delete an incomplete file that is growing too large and have the user send it later, assuming the protocol allows that.

8. Understand the graph algorithm for detecting deadlocks in case there is only resource of each type

In this case for Deadlock detection, we can run an algorithm to check for the cycle in the Resource Allocation Graph. The presence of a cycle in the graph is a sufficient condition for deadlock.

9. Understand the matrix algorithm for detecting deadlocks in case there are multiple resources of each type

Detection of the cycle is necessary but not sufficient condition for deadlock detection, in this case, the system may or may not be in deadlock varies according to different situations.

10. What is an unsafe state?

Unsafe State - **If Operating System is not able to prevent Processes from requesting resources which can also lead to Deadlock**, then the System is said to be in an Unsafe

11. Understand Dijkstra's Banker algorithm for avoiding deadlocks

The Banker’s algorithm sometimes referred to as avoidance algorithm or Deadlock algorithm was developed by Edsger Dijkstra (another of Dijkstra’s algorithms!).

It tests the safety of allocation of predetermined maximum possible resources and then makes states to check the deadlock condition.

12. What is the weakness of the Banker algorithm?

**It requires the number of processes to be fixed; no additional processes can start while it is executing**. It requires that the number of resources remain fixed; no resource may go down for any reason without the possibility of deadlock occurring.

13. What is the idea of spooling to prevent deadlocks?

In spooling, **when multiple processes request the printer, their jobs ( instructions of the processes that require printer access) are added to the queue in the spooler directory**. The printer is allocated to jobs on a first come first serve (FCFS) basis.