1. The property of dynamic scheduling that makes it different from static scheduling is
   a. It is faster than the static scheduling
   b. It reschedules every time when a new process enters the queue
   c. For it the task with shorter deadline starve
   d. For it the tasks with longer deadline starve

   Ans. b (It reschedules every time when a new process enters the queue)
   Justification: Dynamic scheduling reschedules the task every time a new task enters the queue and, in this process, it may pre-empt a task.

2. Suppose 3 repetitive tasks be (4,1), (6,3), (9,3) (Consider period and deadline is same for these tasks) following earliest deadline first. All of them start at timestamp 0. When the timestamp 4 ends the following will take place
   a. Task 3 will start executing.
   b. Task 2 will continue being executed
   c. Task 1 will start executing
   d. Task 1 will be stopped and task 3 will execute.

   Ans. c (Task 1 will start executing)
   Justification: At time stamp 0 task 1 will start executing and will carry on for 1 unit. At time stamp 2 task 2 will start executing and will continue for 3 unit. Now after end of timestamp 4, task 1 reappears and its deadline is 4. The new deadline of task 3 is 5. So, task 1 will be assigned to be executed as its deadline is the earliest.

3. Schedulability check condition in earliest deadline first rule is
   a. Both sufficient and necessary condition to check whether tasks can be scheduled.
   b. Sufficient but not necessary condition to check schedulability.
   c. Necessary but not sufficient condition to check schedulability.
   d. Fails to check schedulability of the tasks in some cases. So, neither necessary nor sufficient.

   Ans. a (Both sufficient and necessary condition to check whether tasks can be scheduled)
   Justification: Schedulability check in earliest deadline first scheduling is both necessary and sufficient condition.

4. Suppose 3 task be (3,1), (5,3), (7,3). Can they be scheduled using earliest deadline first rule? (Period and deadline are same for these tasks)
   a. Yes
   b. No
   c. Can not be decided

   Ans. b (no)
   Justification: According to schedulability check ((1/3) + (3/5) + (3/7)) > 1. The condition is sufficient and necessary for tasks to be schedulable. As the condition fails, these tasks can not be scheduled.

5. Suppose two task T1 and T2 where the priority of T1 is greater than T2, have critical section for same resource starts executing at different time. T2 starts executing first and enters critical section and then the T1 enters the queue. The following will take place:
   a. T1 will start executing and when critical section of T2 comes, T2 will stop and T1 will start executing till its critical section ends.
   b. T1 will start executing and when the critical section comes, it will also keep executing as the critical section of T2 is halted.
   c. T2 will keep on being executed as it is performing critical section.
   d. The outcome can not be predicted.

   Ans. a (T1 will start executing and when critical section of T2 comes, T2 will stop and T1 will start executing till its critical section ends.)

| | |
|---|---|
| | Justification: When T1 enters the queue the critical section of T2 will stop and T1 will execute till its critical section arrives. Then T1 will be halted and T2 will be executed till its critical section is completed executing and next T1 will be executed. |
| 6. | Statement 1: For three or more than three processes having critical section for same resources, the blocking time of a process may even exceed the time required to compute critical section.<br>Statement 2: If the scheduling test fails for RMS scheduling, no scheduling will be possible.<br>a. Statement 1 and 2 both are true.<br>b. Statement 1 is true but statement 2 is false.<br>c. Statement 2 is true but statement 1 is false.<br>d. Statement and 2 both are false.<br>Ans. b (Statement 1 is true but statement 2 is false.)<br>Justification: For three or more than processes having critical section for same resources, the blocking time of a process may even exceed the time required to compute critical section.<br>In case of RMS the condition for schedulability was sufficient but not necessary, i.e. if the test fails, it does not make sure that no scheduling is possible. |
| 7. | The disadvantage of priority inversion protocol is:<br>a. The process that does not need the critical section, starves.<br>b. The process that appears later starves irrespective of its priority<br>c. The process with longest critical section may starve<br>d. When there are more than two process, the higher priority process which needs to access the critical section may starve.<br>Ans. d (When there are more than two process, the higher priority process which needs to access the critical section may starve.)<br>Justification: In priority inversion protocol the priority changes when some higher priority process asks for shared resource that is being accessed by lower priority process. So, one process may take long time due to access of critical section by other lower priority process which will hamper the real time system. |
| 8. | The disadvantages of priority inversion protocol can efficiently be overcome by<br>a. Non pre-emption of process.<br>b. RMS protocol<br>c. Priority inheritance protocol<br>d. Earliest deadline first protocol<br>Ans. c (Priority inheritance protocol)<br>Justification: Priority inheritance protocol is used to overcome the disadvantages of priority inversion protocol. |
| 9. | Statement 1: Disallowing pre-emption may cause starvation of higher priority process without critical section.<br>Statement 2: In priority inversion protocol the priority changes when some higher priority process asks for shared resource that is being accessed by lower priority process.<br>a. Statement 1 and 2 both are true.<br>b. Statement 1 is true but 2 is false<br>c. Statement 1 is false but 2 is true<br>d. Statement 1 and 2 both are false.<br>Ans. a (Statement 1 and 2 both are true.)<br>Justification: Disallowing pre-emption may cause starvation of higher priority process without critical section. As in that case when a lower priority process enters into critical section, higher priority process without even need of the critical section waits for lower priority process to finish its critical section. |

In priority inversion protocol the priority changes when some higher priority process asks for shared resource that is being accessed by lower priority process. Only after critical section finished being executed, the higher priority process starts executing.

10.

Suppose 3 tasks are following priority inheritance protocol and Pr(T1)> Pr(T2) > Pr(T3). Now T3 starts executing and enters critical section. Then T2 enters and pre-empts T1 and next T1 enters and pre-empts T2. Now when the T1 will require critical section following will take place:

a. T1 will start executing and after completion of task T1, T2 will start executing.
b. T3 will inherit the priority of T1 and it will execute its critical section and then T2 will start execution
c. T3 will inherit the priority of T1 and it will execute its critical section and then T1 will start execution
d. The tasks will enter into a deadlock situation as the critical section is hold by T3 and T1 is highest priority process.

Ans. c (T3 will inherit the priority of T1 and it will execute its critical section and then T1 will start execution)
Justification: In this case priority inheritance protocol will be followed and T3 will inherit the priority of T1 as priority inheritance is transitive property. After T3 completes execution of critical section, T1 will start executing the critical section.