Question 1. (8 points)
Explain the difference between preemptive and non-preemptive, priority-driven and time-driven, static and dynamic, on-line and off-line scheduling algorithms.

Question 2. (12 points)
Construct the schedule according to the following four algorithms, the rate monotonic (RM), the deadline monotonic (DM), the earliest deadline first (EDF) and the least laxity first (LLF) for the following set of periodic tasks \( t_i(C_i, D_i, T_i) \) where \( C_i \) is computation time, \( D_i \) is the relative deadline, \( T_i \) is the length of the period: \( t_1(1,3,3), t_2(3,5,9), t_3(1,6,6) \). If the task set is not schedulable, give an explanation. All tasks arrive at time 0.

Question 3. (10 points)
(a) Give two different explanations of why the periodic tasks (Computation time, Period), (1, 6), (3, 9) and (2, 18) are schedulable by the rate-monotonic algorithm.
(b) Suppose that we add one more task into the task set in (a) where the new tasks utilization is 0.15. Is the new task set schedulable by the rate-monotonic algorithm? Be sure to explain your answer with EXAMPLES.

Question 4. (10 points)
A job may be blocked for many reasons. The term blocking time, denoted by \( b_i \), refers to the maximum total duration for which each job in task \( T_i \) may be delayed by both lower-priority tasks and deferred execution of higher-priority tasks. Extend the schedulability conditions 2 & 3 to consider tasks that can be blocked.

Question 5. (10 points)
Given a task set which is RM-schedulable, will the RM and EDF schedules be the same or different? Justify your answer with examples.
Note: if the answer is “can’t tell”, you’ll need two examples for each answer of “same” and “different”.

1
Question 6. (10 points)
Consider a set of $n$ independent tasks. All tasks arrive at $t = 0$. Each task $T_i$ is characterized by its computation time $C_i$ and deadline $D_i$.
Prove that EDF is optimal for both preemptive AND non-preemptive cases.

Question 7. (10 points)
Prove the RM algorithm is an optimal static-priority algorithm for uni-processor with the same conditions in Question 7.

Question 8. (10 points)
Consider a two-task system where each preemption has an overload of $x$. Given $C_1$, $C_2$, $P_1$, $P_2$ ($P_1 > P_2$), obtain the minimum value of $P_2$ for which the task set is RM-schedulable.

Question 9. (10 points)
Determine whether there is a feasible schedule for the following set of periodic tasks.
Construct a schedule.
- $T_1$: $c_{11} = 1$, $c_{12} = 3$, $c_{13} = 2$, $d_1 = 14$, $p_1 = 16$
- $T_2$: $c_{21} = 1$, $c_{22} = 2$, $d_2 = p_2 = 8$
- $T_3$: $c_3 = 3$, $d_3 = p_3 = 16$
$T_1$ must rendezvous with $T_2$ after the first, second, and third scheduling blocks.
$T_2$ must rendezvous with $T_1$ after the first scheduling block.

Question 10. (10 points)
Is it possible to find a set of $n^4$ tasks which can be scheduled on an $n$-processor system?
Justify your answer.