State whether the following are TRUE or FALSE. Give a brief explanation for your answer. One mark for correct answer, and one mark for the explanation.

1. The presence of a cycle in the resource-allocation graph in a system with multiple instances of resource types guarantees that the system is deadlocked.

2. The medium-term scheduler is used as a standby. If either the short-term scheduler or long-term scheduler fails for some reason then the medium-term scheduler handles the work of the failed scheduler.

3. A system is currently in an unsafe state. It moves to a safe state in the next time step. If it now continues to stay in a safe state, then it will never enter a deadlock.

4. At time step \( T = 0 \), there are three processes \( P_0, P_1, P_2 \) in the ready queue of a CPU. The scheduling scheme in use is SRTF (Shortest Remaining-Time First) algorithm. Process \( P_2 \) is selected to run at \( T = 0 \). It is still running when at time instant \( T = 1 \), a process \( P_3 \) enters the ready-queue. The process \( P_1 \) may now be scheduled by the scheduler preempting the CPU from process \( P_2 \), at \( T = 1 \).

5. In a multi-programming environment, a process which manipulates two variables might accidently exchange their values on account of context-switches which move the process in and out of the CPU.
6. The purpose of implementing a small block of code, where a process manipulates shared data, as an *atomic instruction*, in a multi-programming environment is to reduce the slow-down caused by context-switches.

7. In a priority-scheduling scheme, the *aging* policy can be used to ensure processes never starve.

8. In a single processor system, the presence of multiple register-sets allows more than one process to use the CPU *simultaneously*.

9. A semaphore can be used to ensure that instruction $I_1$ in one process $P_1$ executes before another instruction $I_2$ in another process $P_2$.

10. The Dispatch latency (delay caused by context-switches) can be reduced if a system has multiple register-sets rather than just a single set of registers, if the number of processes in memory is at most the number of register-sets.

11. In a round-robin scheduling scheme, if a process terminates *exactly* at the same time as the expiry of its time-slice while it is using the CPU, then all processes in the ready queue are moved out of the main memory temporarily.

12. A priority inversion protocol protects low priority processes from pre-emption in favour of higher priority processes, provided they are modifying data structures which are shared with processes at that higher priority.

13. In a multiprocessor system, a process cannot be moved across processors if its current processor fails if the instruction sets of the two processors are different.

14. A process currently holding no resources may be involved in a deadlock currently, if it has a future requirement of at least two different resource-types.

15. The next burst-length of a process in the ready queue is determined on the basis of the burst-lengths in its history (previous runs on the CPU).
Section B

16. Describe the life cycle of a process in a multiprogramming environment, including a description of its various states.

17. Describe briefly the roles of the short-term (CPU) scheduler, the long-term (job) scheduler and the medium-term scheduler. Specify the circumstances under which the medium-term scheduler is invoked.

18. Describe the general structure of any program with a critical section. In particular, list the significant units in its code and briefly describe the role of each unit.

19. List the FOUR approaches to deadlocks and describe them briefly in a few lines. State the advantage and disadvantage of each of them.

Section C

20. Consider a system consisting of \( m \) resources of the same type, being shared by \( n \) processes. Resources can be requested and released by processes only one at a time. Show that the system is deadlock free if the following conditions hold:

   (a) The maximum needs of each process is between 1 and \( m \).

   (b) The sum of all maximum needs is less than \( m + n \).

21. What are the three requirements of a critical section protocol? Describe them. Prove that the following implementation meets all the requirements in a two process critical section problem. The processes are \( P_i \) and \( P_j \).
Code for process $P_i$

repeat
  $flag[i] \leftarrow true$;
  while $flag[j]$ do
    if $turn = j$ then
      $flag[i] \leftarrow false$;
      while $turn = j$ do
        no-op
      end while
      $flag[i] \leftarrow true$;
    end if
  end while
CRITICAL SECTION
$turn \leftarrow j$
$flag[i] \leftarrow false$
$flag[i] \leftarrow false$
REMAINDER SECTION
until $false$