1.Which of the following statements is/are TRUE with respect to deadlocks?
A
Circular wait is a necessary condition for the formation of deadlock.
B
In a system where each resource has more than one instance, a cycle in its wait for graph indicates the presence of a deadlock.
C
If the current allocation of resources to processes leads the system to unsafe state, then deadlock will necessarily occur.
D
In the resource-allocation graph of a system, if every edge is an assignment edge, then the system is not in deadlock state.

2.

Consider a system with 3 processes that share 4 instances of the same resource type. Each process can request a maximum of K instances. Resource instances can be requested and released only one at a time. The largest value of K that will always avoid deadlock is ____.

3.A system has 6 identical resources and N processes competing for them. Each process can request atmost 2 resources. Which one of the following values of N could lead to a deadlock?
A 1
B 2
C 3
D 6

4.A computer has six tape drives, with n processes competing for them. Each process may need two drives. What is the maximum value of n for the system to be deadlock free?
A 6
B 5
C 4
D 3

5.Consider a multi-threaded program with two threads T1 and T2. The threads share two semaphores: s1 (initialized to 1) and s2 (initialized to 0). The threads also share a global variable x (initialized to 0). The threads execute the code shown below.

// code of T1

```
wait(s1);

x = x + 1;
```

```
print(x);

wait(s2);

signal(s1);
```

// code of T2

```
wait(s1);

x = x + 1;

print(x);

signal(s2);

signal(s1);
```

5.Which of the following outcomes is/are possible when threads T1 and T2 execute concurrently?
A
T1 runs first and prints 1, T2 runs next and prints 2
B
T2 runs first and prints 1, T1 runs next and prints 2
C
T1 runs first and prints 1, T2 does not print anything (deadlock)
D
T2 runs first and prints 1, T1 does not print anything (deadlock)

6.Consider the following pseudocode, where S is a semaphore intialized to 5 in line#2 an counter is a shared variable intialized to 0 in line#1. Assume that the increment operation in line#7 is not atomic.

1. int counter = 0;

2. Semaphore S = init(5);

3. void parop(void)

4. {

5. wait (S);

6. wait (S);

7. counter++;

8. signal (S);

9. signal (S);

10. }

If five threads execute the function parop concurrently, which of the following program behavior (s) is/are possible?

A There is a deadlock involving all the threads.

B The value of counter is 5 after all the threads successfully complete the execution of parop.

C The value of counter is 1 after all the threads successfully complete the execution of parop.

D The value of counter is 0 after all the threads successfully complete the execution of parop.

In a system, there are three types of resources and four processes that execute concurrently. At the outset, the processes have declared their maximum resource requirements using a matrix named Max as given below. For example, Max[ ] is the maximum number of instances of that would require. The number of instances of the resources allocated to the various processes at any given state is given by a matrix named Allocation.
Consider a state of the system with the Allocation matrix as shown below, and in which 3 instances of E and 3 instances of F are the only resources available.

Allocation			
	E	F	G
P_0	1	0	1
P_1	1	1	2
P_2	1	0	3
P_3	2	0	0

Max			
	E	F	G
P_0	4	3	1
P_1	2	1	4
P_2	1	3	3
P_3	5	4	1

From the perspective of deadlock avoidance, which one of the following is true?

1. The system is in safe state.
2. The system is not in state, but would be if one more instance of E were available
3. The system is not in safe state, but would be safe if one more instance of F were available
4. The system is not in state, but would be if one more instance of G were available

Allocation				**Max**				**Need=Max-Allocation**			
Process	E	F	G	Process	E	F	G	Process	E	F	G
P_0	1	0	1	P_0	4	3	1	P_0	3	3	0
P_1	1	1	2	P_1	2	1	4	P_1	1	0	2
P_2	1	0	3	P_2	1	3	3	P_2	0	3	0
P_3	2	0	0	P_3	5	4	1	P_3	3	4	1

Available Resource $[E, F, G] = (3, 3, 0)$
With $(3, 3, 0)$ we can satisfy the request of either P_0 or P_2.

Let's assume request of P_0 satisfied.
After execution, it will release resources.
Available Resource $= (3, 3, 0) + (1, 0, 1) = (4, 3, 1)$

Give $(0, 3, 0)$ out of $(4, 3, 1)$ unit of resources to P_2 and P_2 will completes its execution.
After execution, it will release resources.
Available Resource $= (4, 3, 1) + (1, 0, 3) = (5, 3, 4)$

Allocate $(1, 0, 2)$ out of $(5, 3, 4)$ unit of resources to P_1 and P_1 will completes its execution.
After execution, it will release resources.
Available Resource $= (5, 3, 4) + (1, 1, 2) = (6, 4, 6)$
And finally, allocate resources to P_3.

So, we have one of the possible safe sequence: $P_0 \longrightarrow P_2 \longrightarrow P_1 \longrightarrow P_3$

Correct Answer: A