**Question 1:**

Consider a system with 5 processes (P1 to P5) and 3 types of resources (R1, R2, R3). The maximum demand, allocation, and available resources are given in the tables below:

**Maximum Demand Matrix:**

|  | **R1** | **R2** | **R3** |
| --- | --- | --- | --- |
| P1 | 7 | 5 | 3 |
| P2 | 3 | 2 | 2 |
| P3 | 9 | 0 | 2 |
| P4 | 2 | 2 | 2 |
| P5 | 4 | 3 | 3 |

**Allocation Matrix:**

|  | **R1** | **R2** | **R3** |
| --- | --- | --- | --- |
| P1 | 0 | 1 | 0 |
| P2 | 2 | 0 | 0 |
| P3 | 3 | 0 | 2 |
| P4 | 2 | 1 | 1 |
| P5 | 0 | 0 | 2 |

**Available Resources:**

| **R1** | **R2** | **R3** |
| --- | --- | --- |
| 3 | 3 | 2 |

**Question:** Is the system in a safe state? If so, provide a safe sequence.

a) Yes, Safe Sequence: P1, P2, P3, P4, P5  
**b) Yes, Safe Sequence: P2, P4, P5, P1, P3**  
c) Yes, Safe Sequence: P4, P5, P2, P1, P3  
d) No, the system is not in a safe state

**Question 2:**

Given the following resource allocation graph with 4 processes (P1 to P4) and 3 resources (R1 to R3):

**Maximum Demand Matrix:**

|  | **R1** | **R2** | **R3** |
| --- | --- | --- | --- |
| P1 | 6 | 0 | 3 |
| P2 | 3 | 2 | 2 |
| P3 | 3 | 1 | 1 |
| P4 | 4 | 2 | 2 |

**Allocation Matrix:**

|  | **R1** | **R2** | **R3** |
| --- | --- | --- | --- |
| P1 | 4 | 0 | 1 |
| P2 | 1 | 2 | 1 |
| P3 | 2 | 1 | 1 |
| P4 | 1 | 0 | 2 |

**Available Resources:**

| **R1** | **R2** | **R3** |
| --- | --- | --- |
| 1 | 0 | 0 |

**Question:** Determine if there is a deadlock in the system.

**a) No, there is no deadlock**b) Yes, deadlock involving P1, P2, and P3  
c) Yes, deadlock involving P2 and P3  
d) Yes, deadlock involving P1 and P4

**Question 3:**

In a system with 4 processes (P1 to P4) and 3 resources (R1, R2, R3), the resource allocation is as follows:

**Maximum Demand Matrix:**

|  | **R1** | **R2** | **R3** |
| --- | --- | --- | --- |
| P1 | 3 | 2 | 2 |
| P2 | 6 | 1 | 3 |
| P3 | 3 | 1 | 4 |
| P4 | 4 | 2 | 2 |

**Allocation Matrix:**

|  | **R1** | **R2** | **R3** |
| --- | --- | --- | --- |
| P1 | 1 | 0 | 0 |
| P2 | 6 | 1 | 2 |
| P3 | 2 | 1 | 1 |
| P4 | 0 | 0 | 2 |

**Available Resources:**

| **R1** | **R2** | **R3** |
| --- | --- | --- |
| 1 | 1 | 0 |

**Question:** What is the state of the system? Is it safe, and if so, provide a safe sequence.

a) Yes, Safe Sequence: P1, P3, P4, P2  
b) Yes, Safe Sequence: P2, P1, P3, P4  
**c) No, the system is not in a safe state**  
d) Yes, Safe Sequence: P1, P4, P3, P2

**Question 1:**

A system uses two semaphores, S and Q, to control access to a critical section. Initially, S = 1 and Q = 0. The processes P1 and P2 use the following code segments:

**Process P1:**

P(S)

<critical section>

V(Q)

**Process P2:**

P(Q)

<critical section>

V(S)

Assuming both processes are started at the same time, which of the following statements is true?

a) P1 will execute its critical section first, followed by P2  
b) P2 will execute its critical section first, followed by P1  
**c) Both P1 and P2 will execute their critical sections concurrently**  
d) A deadlock will occur

**Question 2:**

Consider three processes P1, P2, and P3 that share a resource protected by a semaphore S. The initial value of S is 1. The processes execute the following code segments:

**Process P1:**

P(S)

<critical section>

V(S)

**Process P2:**

P(S)

<critical section>

V(S)

**Process P3:**

P(S)

<critical section>

V(S)

Assuming no other synchronization mechanisms are used, what is the maximum number of processes that can be in the critical section at the same time?

a) 0  
**b) 1**  
c) 2  
d) 3

**Question 3:**

A system has two types of resources managed by semaphores S1 and S2. The initial values of S1 and S2 are both 1. Two processes P1 and P2 need to access these resources as follows:

**Process P1:**

P(S1)

P(S2)

<critical section>

V(S2)

V(S1)

**Process P2:**

P(S2)

P(S1)

<critical section>

V(S1)

V(S2)

What potential problem does this setup create?

a) Starvation  
**b) Deadlock**  
c) Priority inversion  
**d) Mutual exclusion violation**

**Question 4:**

Consider a system with 4 processes (P1 to P4) and a binary semaphore S initialized to 1. The processes access a shared resource as follows:

**Process P1:**

P(S)

<critical section>

V(S)

**Process P2:**

P(S)

<critical section>

V(S)

**Process P3:**

P(S)

<critical section>

V(S)

**Process P4:**

P(S)

<critical section>

V(S)

If processes P1, P2, P3, and P4 attempt to enter the critical section simultaneously, in which order will they be able to execute their critical sections?

a) P1, P2, P3, P4  
b) P4, P3, P2, P1  
c) P1, P3, P2, P4  
**d) Any arbitrary order depending on the scheduler**

**Question 5:**

Consider a producer-consumer problem with a buffer of size 5. The producer and consumer processes use two semaphores, empty (initialized to 5) and full (initialized to 0), to manage access to the buffer. The code segments are:

**Producer:**

P(empty)

<add item to buffer>

V(full)

**Consumer:**

P(full)

<remove item from buffer>

V(empty)

If the producer has added 3 items to the buffer and the consumer has removed 1 item, what are the current values of empty and full?

a) empty = 4, full = 1  
**b) empty = 3, full = 2**  
c) empty = 2, full = 3  
d) empty = 3, full = 1

**Question**

Each Process Pi, i= 1.......9 is coded as follows

repeat

P(mutex)

{Critical section}

V(mutex)

forever

The code for P10 is identical except it uses V(mutex) in place of P(mutex). What is the largest number of processes that can be inside the critical section at any moment?

1. **1**
2. 2
3. 3
4. None of above

---Nahi hua tha

**Question**

Suppose n processes, P1, …. Pn share m identical resource units, which can be reserved and released one at a time. The maximum resource requirement of process Pi is Si, where Si > 0. Which one of the following is a sufficient condition for ensuring that deadlock does not occur? 

C