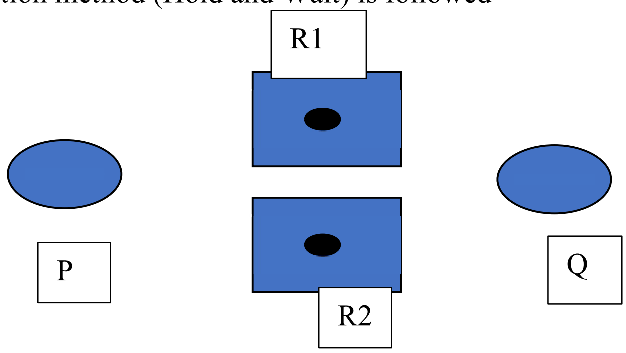
TUTORIAL-6

Q 1. Explain whether the deadlock will occur for this situation? Semaphores *A* and *B*, initialized to 1

P_0	P_1
wait (A)	*wait(B)*
wait (B)	*wait(A)*
signal(B)	*signal(A)*
signal(A)	*signal(B)*

Q 2. Prove that the process/resource configuration below cannot result in a deadlock if the Deadlock prevention method (Hold and Wait) is followed



Q 3. Consider following system(One resource class only)

Process	Holding	Max claims
A	4	6
B	4	11
C	2	7

unallocated: 2
Determine whether it is in safe state or not? If C should have a claim of 9 instead of 7, then is there any safe state?

Q 4. Deadlock is defined as two or more processes are waiting indefinitely for an event that can be caused by only one of the waiting processes. Give example of deadlock for 2 processes sharing some common semaphore.

Q 5. Consider the following snapshot of a system:

a. What is the content of the matrix *Need*?

b. Is the system in a safe state?

c. If a request from process *P*1 arrives for (0,4,2,0), can the request be granted immediately?

	Allocation	*Max*	*Available*
	A B C D	*A B C D*	*A B C D*
P_0	0 0 1 2	0 0 1 2	1 5 2 0
P_1	1 0 0 0	1 7 5 0	
P_2	1 3 5 4	2 3 5 6	
P_3	0 6 3 2	0 6 5 2	
P_4	0 0 1 4	0 6 5 6	

Q 6. There are four processes and two resources in the system. Each resource has two instances.

Furthermore:

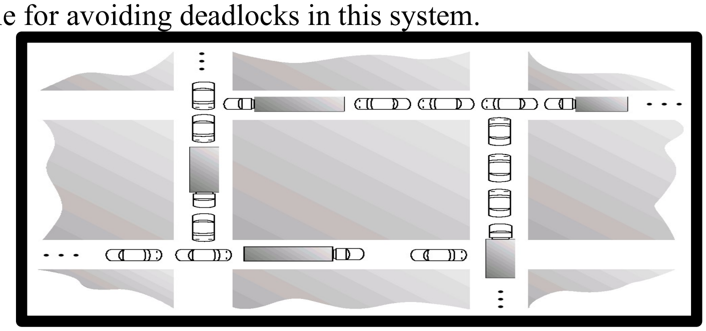
- P1 acquires an instance of R2, and requests an instance of R1
- P2 acquires an instance of R1, and doesn't need any other resource
- P3 acquires an instance of R1 and requires an instance of R2
- P4 acquires an instance of R2, and doesn't need any other resource.

a) Draw the resource allocation graph.

b) Is there a cycle in the graph? If yes name it.

c) Whether the system is in deadlock? If yes, explain why. If not, provide sequence of execution of processes.

Q 7. Show that the four necessary conditions for deadlock indeed hold in this example. State a simple rule for avoiding deadlocks in this system.



Q 8. 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?

Q 9. 'm' processes share 'n' resources of the same type. The maximum need of each process doesn't exceed 'n' and the sum of all their maximum needs is always less than m+n. Whether deadlock will occur or not?

Q 10. 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. What will be the largest value of K that will always avoid deadlock?