BTLT 4 – SYNCHRONIZATION AND DEADLOCK

1. Consider an online shopping process consisting of the following steps:
   -> makeOrder
   -> receiveOrder
   -> requireAdditionalInformation
   -> provideRequiredInformation
   -> verifyReceivedInformation
   -> deliverProduct
   -> receiveProduct
   The following processes Client and Shopping allow performing the above steps:

Client {
  makeOrder();
  provideRequiredInformation();
  receiveProduct();
}
	Shopping {
	  receiveOrder();
	  requireAdditionalInformation();
	  verifyReceivedInformation();
	  deliverProduct();
	}

   Synchronize processes Client and Shopping Stored by using semaphores to ensure the order of the above shopping process.

2. To form a H20 molecule, one oxygen atom and two Hydrogen atoms are needed. Let us suppose three processes P1, P2 and P3 allowing to create an Oxygen atom, a Hydrogen, and a H20 molecule, respectively, as below:

P1 {
  create1Oxygen();
}
	P2 {
	  create1Hydrogen();
	}
		P3 {
		  create1H20();
		}

   Write pseudo-code for these three processes and use semaphores to synchronize them so that:
   - Different processes P1 and P2 can coexist at the same time in the system.
   - Only one process P3 performs its function at a time in the system.
   - If one Oxygen atom and two Hydrogen atoms have been produced, all instances of P1 and P2 will be blocked until all existing atoms assembled into H20 molecule (P3). P3 will be blocked if there are not enough oxygen and hydrogen atoms.

3. Consider a system consisting of 4 resources of the same type that are shared by 2 processes, each of which needs at most 3 resources. Can deadlock occur? Explain your answer.

4. Consider a system consisting of m resources of the same type that are shared by 3 processes, each of which needs at most 2 resources. What is the minimum value of m that ensures no deadlock?

5. Consider a system having 6 six tape drives sharing by n processes, each of which needs at most 2 tape disks.
   a. What is the minimum value of n that can lead the system to deadlock situation?
   b. What is the maximum value of n that can ensure no deadlock?

6. Consider a system has 4 processes P1, P2, P3, and P4 sharing a resource R. P1, P2, P3, and P4 needs at most 3, 1, 2, and 2 units of the resource R, respectively. What is the minimum number of units of R that can ensure no-deadlock?

7. Consider the following snapshot of a system.

	Allocation					Max					Need				
	A	**B**	**C**	**D**	**E**	**A**	**B**	**C**	**D**	**E**	**A**	**B**	**C**	**D**	**E**
P1	1	3	1	3	2	4	5	2	3	2	3	2	1	0	0
P2	2	0	0	3	1	3	2	3	4	2	1	2	3	1	1
P3	4	1	0	0	0	4	1	1	1	2	0	0	1	1	2
P4	0	0	0	1	1	1	2	1	2	2	1	2	1	1	1
P5	1	2	3	0	0	2	2	3	1	1	1	0	0	1	1

Available					
A	B	C	D	E	
1	2	2	2	1	
2	4	5	3	2	P4, P5: ok
8	5	5	6	3	P2, P3: ok
					P4, P5, P3, P2, P1
0	0	1	2	1	d.==> no-safe

   a. Fill in the matrix Need.
   b. Is the system in a safe state Safe? Explain your answer.
   c. If P1 requests for (1, 2, 3, 4, 5), can the request be granted? Explain your answer.
   d. If P1 requests for (1, 2, 1, 0, 0), can the request be granted? Explain your answer.