1. i) Using the Banker's Safety algorithm, determine whether or not the state is unsafe. If the state is safe, illustrate the order in which the processes may complete. Otherwise, illustrate why the state is unsafe. You must calculate the Need matrix.

	Allocation			Max			Available		
	R1	**R2**	**R3**	**R1**	**R2**	**R3**	**R1**	**R2**	**R3**
P0	2	5	0	8	5	9	10	7	7
P1	1	1	1	5	6	8			
P2	2	5	2	6	10	8			
P3	4	0	3	5	6	10			
P4	0	3	0	9	5	9			

ii) If a request from process P3 arrives for [0, 2, 7], can the request be granted immediately? If yes then after accepting the request does the system remain in the safe state?

Answer: i) P1 → P2 → P3 → P4 → P0
ii) Yes. If granted, P3 → P4 → P0 → P1 → P2

2. i) Using the Banker's Safety algorithm, determine whether or not the state is unsafe. If the state is safe, illustrate the order in which the processes may complete. Otherwise, illustrate why the state is unsafe. You must calculate the Need matrix.

	Allocation			Max			Available		
	R1	**R2**	**R3**	**R1**	**R2**	**R3**	**R1**	**R2**	**R3**
P0	0	2	3	10	7	6	2	5	7
P1	2	3	2	6	6	5			
P2	5	4	5	7	7	7			
P3	3	2	5	5	5	7			
P4	3	2	0	6	10	6			

ii) If a request from process P3 arrives for [0, 1, 2], can the request be granted immediately? If yes then after accepting the request does the system remain in the safe state?

Answer: i) P2 → P3 → P4 → P0 → P1
ii) Yes. If granted, P2 → P3 → P4 → P0 → P1

3. i) Using the Banker's Safety algorithm, determine whether or not the state is unsafe. If the state is safe, illustrate the order in which the processes may complete. Otherwise, illustrate why the state is unsafe. You must calculate the Need matrix.

	Allocation			Max			Available		
	R1	**R2**	**R3**	**R1**	**R2**	**R3**	**R1**	**R2**	**R3**
P0	5	1	5	10	7	7	2	9	1
P1	3	0	5	8	9	5			
P2	0	1	4	5	7	9			
P3	4	1	2	7	10	5			
P4	4	1	5	5	8	5			

ii) If a request from process P0 arrives for [2, 3, 0], can the request be granted immediately? If yes then after accepting the request does the system remain in the safe state?

Answer: i) P4 → P0 → P1 → P2 → P3
ii) No. If approved, no safe sequence was found.

4. Consider the following resource allocation graph has:
   Set of processes, P = {P1, P2, P3, P4}
   Set of resources, R = {R1, R2, R3}
   Set of edges, E = {P1 → R3, P2 → R1, P3 → R2, R1 → P1, R1 → P3, R2 → P1, R2 → P2, R2 → P4, R3 → P2, R3 → P4}
   - Resource type R1 has 2 instances.
   - Resource type R2 has 3 instances.
   - Resource type R3 has 2 instances.

   Draw the resource allocation graph and explain the possibility of deadlock.

   Answer: No deadlock.

5. Consider the following resource allocation graph has:
   Set of processes, P = {P1, P2, P3, P4}
   Set of resources, R = {R1, R2, R3}
   Set of edges, E = {P1 → R2, P2 → R3, P3 → R3, P4 → R2, R1 → P1, R1 → P3, R1 → P4, R2 → P2, R2 → P3, R3 → P1}
   - Resource type R1 has 3 instances.
   - Resource type R2 has 2 instances.
   - Resource type R3 has 1 instance.

   Draw the resource allocation graph and explain the possibility of deadlock.

   Answer: Deadlock.