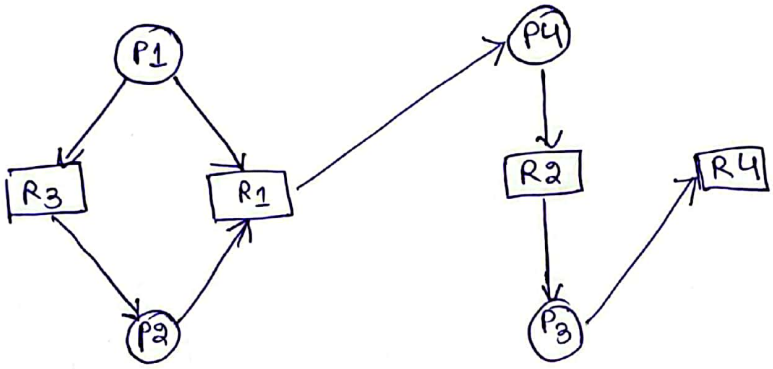
P.S. Assignment -4

Q1) consider a system with four processes P1, P2, P3 and P4, resources types R, R2, R3, R4 each one with a single instance. Draw the resource allocation graph corresponding to following resource allocation state:

- P1 requests for R3 and R1
- P2 holds R3 and requests for R1
- P3 holds R2 and requests for R1
- P4 holds R1 and requests for R2

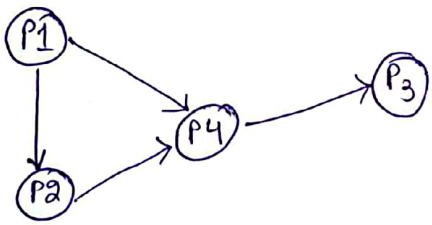
Using wait graph, check whether the system is in deadlock or not? If so how many processes are involved in deadlock?

(Ans)



the system is not in deadlock. [No cycle]

wait for graph



Q2) consider a system with 12 tapedrives & 3 processes: P0, P1, P2. P0 requires 4 tapedrives, P1 requires 10 tapedrives, P2 requires 9 tapedrives for completing their task. suppose at time t0, P0 is holding 2 tapedrives, P1 is holding 5 tapedrives and P2 is holding 2 tapedrives then check whether the current resource allocation state is safe or not. If yes specify the safe sequence.

(Ans) given :- 12 tapedrive → Resource

Process → P0, P1, P2

	Request
P0	4
P1	10
P2	9

	Allocated	Req.	Available	Req < available	New available
P0	2	2	3	true	5
P1	5	5		true	10
P2	2	7		true	12

Yes, current allocation is safe

Safe sequence :- P0, P1, P2

03) consider a system has 10 units of a resource type and 5 resources the current resource allocation state is given as follows:

Process	Used	Max
P0	2	5
P1	1	6
P2	2	6
P3	1	2
P4	1	4

If P2 will request for 2 more instances of the same resource type, check the request can be granted immediately or not?

(Ans)

Process	Used	Max	Req	Available	Req < available	New available
P0	2	5	3	3	True	5
P1	1	6	5		True	6
P2	2	6	4		True	8
P3	1	2	1		True	9
P4	1	4	3		True	10

Safe sequence :- P0, P1, P2, P3, P4

P2 will be granted with 2 more additional instances if it request as there are 3 more instances available.

Q4) consider a system with two resources A and B. A has 6 instances and B has 3 instances. Can the system execute the following processes without any deadlock ? If yes specify the safe sequence.

Process	Allocation		Max	
	A	B	A	B
P0	1	1	2	2
P1	1	0	4	2
P2	1	0	3	2
P3	0	1	1	1
P4	2	1	6	3

(Ans)

Process	Allocation		Max		Need		available		need < available		new available	
	A	B	A	B	A	B	A	B	A	B	A	B
P0	1	1	2	2	1	1	1	0	False	True	2	2
P1	1	0	4	2	3	2			F	T	4	2
P2	1	0	3	2	2	2			F	T	3	2
P3	0	1	1	1	1	0			T	T	1	1
P4	2	1	6	3	4	2			F	F	6	3

Safe sequence :- $< P_3, P0, P2, P1, P4 >$

Q5) Consider the following resource allocation state with 3 processes & 3 resources :

	Allocation			Max		
	X	Y	Z	X	Y	Z
P0	0	0	1	7	4	3
P1	3	2	0	6	2	0
P2	2	1	1	3	3	3

there are 3 instances of type X, 2 instances of type Y and 2 instances of type Z, still available.

a) find the content of the need matrix.

b) Is the system in a safe state?

c) If P_0 will request for 2 more instances of type Z, can the request be granted immediately or not?

(Ans) a)

	Allocation			Max			Need			available			new available		
	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
P0	0	0	1	7	4	3	7	4	2	3	2	2	8	5	4
P1	3	2	0	6	2	0	3	0	0				6	4	2
P2	2	1	1	3	3	3	1	2	2				8	5	3

b) Safe sequence :- $\{ P_1, P2, P0 \}$

c) No, P0 can't be assigned with 2 more instance of 2 as P0 already needs 2 instance of Z and are available with only 3 instance of 2

Q6) Consider the following resource allocation state with 4 processes & 4 resources, and an available vector of <0100>

	Allocation				Request			
	A	B	C	D	A	B	C	D
P0	0	1	1	0	0	0	1	0
P1	0	1	0	1	1	0	0	1
P2	1	2	0	0	0	0	0	1
P3	0	0	1	2	0	0	0	0
P4	1	0	1	0	0	1	0	0

a) Find the initial number of instances available for each resource type

b) Check whether the system is deadlock free or not

c) If P_0 is assigned with 1 more instance of type B, then check whether the system is deadlock free or not.

(Ans)

	Allocation				Request				New available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	1	1	0	0	0	1	0	1	2	3	2
P1	0	1	0	1	1	0	0	1	1	2	3	3
P2	1	2	0	0	0	0	0	1	2	5	3	3
P3	0	0	1	2	0	0	0	0	0	0	1	2
P4	1	0	1	0	0	1	0	0	1	1	2	2

a) available

A → 0

B → 1

C → 0

D → 0

b) Safe sequence :- $\langle P_3, P_4, P_0, P_1, P_2 \rangle$

Deadlock free.

a) $P_0 \rightarrow$ 1 more type B

	Allocation				Request				available				new available			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
													0	2	2	2
P_0	0	2	1	0	0	0	1	0	0	0	0	0	2	5	3	3
P_1	0	1	0	1	1	0	0	1					1	4	2	2
P_2	1	2	0	0	0	0	0	1					0	0	1	2
P_3	0	0	1	2	0	0	0	0					2	4	3	2
P_4	1	0	1	0	0	1	0	0								

safe sequence :- $< P_3, P_0, P_2, P_4, P_1 >$

deadlock free