2)

Process	Allocation			MAX		
	R_1	R_2	R_3	R_1	R_2	R_3
P_0	0	1	0	7	5	3
P_1	2	0	0	3	2	2
P_2	3	0	2	9	0	2
P_3	2	1	1	2	2	2
P_4	0	0	2	4	3	3

Available

R_1	R_2	R_3
3	3	2

Suppose at time T_1, process P_2 request 1 additional instance of resource type R_1, 2 instance of resource type R_3. Can the request be granted immediately

⇒ Request of P_2 (1, 0, 2)

$need_{P_2}$ = 902 − 302 = 600 (MAX − allocation)

Request ≤ need

1 0 2 ≤ 600

1 ≤ 6

0 ≤ 0

2 ≤ 0 ×

∴ The process P_2 request can not be granted immediately

3) Consider a system with set of 5 process & 4 Resources

Process	Allocation				MAX			
	A	B	C	D	A	B	C	D
P_0	0	0	1	2	0	0	1	2
P_1	1	0	0	0	1	7	5	0
P_2	1	3	5	6	2	3	5	6
P_3	0	6	3	2	0	6	5	2
P_4	0	0	1	4	0	6	5	6

Available

A	B	C	D
1	5	2	0

(i) If the request from process P_1 arrives for 4 instance of a resource type 'B' & 2 instances for resource type 'C'. Can the request be granted immediately

=> Request of P_1 (0, 4, 20)

need = 1750 − 1000 = 0750

Request ≤ need

0420 ≤ 0750

0 ≤ 0 ✓

4 ≤ 7 ✓

2 ≤ 5 ✓

0 ≤ 0 ✓

Request ≤ Available

0420 ≤ 1520

0 ≤ 1 ✓

4 ≤ 5 ✓

2 ≤ 2 ✓

0 ≤ 0 ✓

Available = Available − Request

= 1520 − 0420

= 1100

Need = Need − Request

= 0750 − 0420

= 0330

Allocation = Allocation + Request

= 1000 + 0420

= 1420

∴ Process P_1 request can be granted immediately

ii) Suppose on granting the above P_1 request, can the system be in safe state or not

=> Available = 1100

Allocation of P_1 = 1420

Need of P_1 = ~~0750~~ 0330

work = available = 1100

need ≤ work

0330 ≤ 1100

0 ≤ 1

3 ≤ 1 ×

3 ≤ 0 ×

0 ≤ 0

Process	Allocation A B C D	MAX A B C D	Need A B C D
P_0	0 0 1 2	0 0 1 2	0 0 0 0
P_1	1 4 2 0	1 7 5 0	0 3 3 0
P_2	1 3 5 4	2 3 5 6	1 0 0 2
P_3	0 6 3 2	0 6 5 2	0 0 2 0
P_4	0 0 1 4	0 6 5 6	0 6 4 2

i = 0, P_0 work = available = 1100

need ≤ work

0000 ≤ 1100

0 ≤ 1

0 ≤ 1

0 ≤ 0

0 ≤ 0

work = work + allocation = 1100 + 0012 = 1112

i = 1, P_1 work = 1112

need ≤ work

0330 ≤ 1112

0 ≤ 1

3 ≤ 1 ×

3 ≤ 1 ×

0 ≤ 2

i = 2, P_2 work = 1112

need ≤ work

1002 ≤ 1112

1 ≤ 1

0 ≤ 1

0 ≤ 1

2 ≤ 1

work = work + allocation

= 1112 + 1354

= 2466

i = 3, P_2 work = 2466

need ≤ work

0020 ≤ 2466

0 ≤ 2

0 ≤ 4

2 ≤ 6

0 ≤ 6

work = work + allocation

= 2466 + 0632

= 2 10 9 8

P_4 , work = 2 10 9 8

need ≤ work

0642 ≤ 2 10 9 8

0 ≤ 2

6 ≤ 10

4 ≤ 9

2 ≤ 8

work = work + allocation

= 2 10 9 8 + 0014

= 2 10 10 12

Roll back

P_1 , work = 2 10 10 12

need ≤ work

0330 ≤ 2 10 10 12

0 ≤ 2

3 ≤ 10

3 ≤ 10

0 ≤ 12

work = work + allocation

= 2 10 10 12 + 1420

= 3 14 12 12

∴ System is in safe state & safe sequence is P_0, P_2, P_3, P_4, P_1

4) Consider a system with a set of 5 process P_0, P_1, P_2, P_3, P_4. There are 3 types of resources R_1, R_2, R_3 There are 10 instance of R_1, 5 instance R_2 & 7 instance R_3. At time to, situation is as follows

Process	Allocation			MAX		
	R_1	R_2	R_3	R_1	R_2	R_3
P_0	0	1	0	7	5	3
P_1	2	0	0	3	2	2
P_2	3	0	2	9	0	2
P_3	2	1	1	2	2	2
P_4	0	0	2	4	3	3

Apply Banker's algorithm for the following question.

(i) Is the system in safe state at to

(ii) Suppose at time t_1, process P_2 request 1 addition instance of resource type R_1 & 2 instances of resource type R_3 will the process request be granted immediately

(iii) After granting process P_1 Request, will the system be in safe state

(i) =>

	Need		
	R_1	R_2	R_3
P_0	7	4	3
P_1	1	2	2
P_2	6	0	0
P_3	0	1	1
P_4	4	3	1

Available		
R_1	R_2	R_3
10-7	5-2	7-5
3	3	2

i=0, P_0, work = available = 332

need $\leq$ work

$743 \leq 332$

$7 \leq 3$ ×

$4 \leq 3$ ×

$3 \leq 2$ ×

P_1, need $\leq$ work

$122 \leq 332$

$1 \leq 3$ ✓

$2 \leq 3$ ✓

$2 \leq 2$

work = work + allocation

= 332 + 200

= 532

P_2 work = 532

need $\leq$ work

$600 \leq 532$

$6 \leq 5$ ×

$0 \leq 3$

$0 \leq 2$

P_3 work = 532

need ≤ work

011 ≤ 532

0 ≤ 5

1 ≤ 3

1 ≤ 2

work = work + allocation

= 532 + 211

= 743

P_4 work = 743

need ≤ work

431 ≤ 743

4 ≤ 7

3 ≤ 4

1 ≤ 3

work = work + allocation

= 743 + 002

= 745

Roll back

P_0 work = 745

need ≤ work

743 ≤ 745

7 ≤ 7

4 ≤ 4

3 ≤ 5

work = work + allocation

= 745 + 010

= 755

P_2 work = 755

need ≤ work

600 ≤ 755

6 ≤ 7

0 ≤ 5

0 ≤ 5

work = work + allocation

= 755 + 302

= 1057

∴ System is in safe state & safe sequence is

P_1, P_3, P_4, P_0, P_2

∴

(ii) ⇒ P_2 request = (1,0,2)

need ≤ 600

request ≤ need

$102 \leq 600$

$1 \leq 6$ ✓

$0 \leq 0$

$2 \leq 0$ ✗

∴ The process P_2 request can not be granted immediately

(iii) Since the process request can not granted and hence forth no need to apply safety algorithm as value remains same

5) Consider a system with set of 3 Process

Process	Allocation			MAX			Available		
	A	B	C	A	B	C	A	B	C
P_0	2	2	3	3	6	8	2	3	0
P_1	2	0	3	4	3	3			
P_3	1	2	4	3	4	4			

(i) Calculate need

(ii) Apply Banker's algorithm for following sub questions

(a) Check whether system is in safe state or not

(b) If the P_3 does not want any addition instance of type A, B, C. Is there any necessity to apply resource request algorithm

(c) Suppose if P_2 request for 4 addition instance of type A, 6 addition instance of type B, & addition instance of type C. Check if a process request can be granted immediately or not

(d) Suppose on granting process request, check if the system is in safe state or not

Deadlock detection problems

Consider a set of system with set of 5 process

Process	Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	0	0	0	0	0	0
P_1	2	0	0	2	0	2			
P_2	3	0	3	0	0	0			
P_3	2	1	1	1	0	0			
P_4	0	0	2	0	0	2			

⇒ work = available = 000

P_0, request ≤ work

000 ≤ 000 ✓

work = work + allocation

= 000 + 010 = 010

P_1, work = 010

request ≤ work

202 ≤ 010 ✗

P_2 work = 010

request ≤ work

000 ≤ 010 ✓

work = work + allocation

= 010 + 303

= 313

P_3 work = 313

request ≤ work

100 ≤ 313 ✓

work = work + allocation

= 313 + 211 = 523

P_4 work = 523

request ≤ work

002 ≤ 523 ✓

work = work + allocation

= 523 + 002

= 525

Roll back

P_1 : work = 525

request ≤ work

~~000~~ 202 ≤ 525 ✗✓

P_4 : work = 521

request ≤ work

002 ≤ 521

∴ System is not in deadlock

24

Process	Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	0	0	0	0	0	0
P_1	2	0	0	2	0	2			
P_2	3	0	3	0	0	1			
P_3	2	1	1	1	0	0			
P_4	0	0	2	0	0	2			

P_0 = work = 000

request ≤ work

000 ≤ 000

work = work + allocation = 000 + 010 = 010

P_1, work = 010

request ≤ work

202 ≤ 010 ✗

P_2 , request ≤ work

001 ≤ 010 ✗

P_3 , request ≤ work

100 ≤ 010 ✗

P_4 request $\leq$ work

$002 \leq 010$ ✗

Roll back

P_2 request $\leq$ work

$001 \leq 010$ ✗

∴ ~~System is in~~

∴ Deadlock is detected in the system