Banker's Algorithm:

Q: Given the following state of banker's algorithm

6 process → P0 to P5

4 resource types → A (15 instances); B (6 instances)
C (9 instances); D (10 instances)

Snapshot at time T0: Available

A

A	B	C	D
6	3	5	4

Allocation

Max

Process	Current allocation A	B	C	D	Maximum Demand A	B	C	D
P0	2	0	2	1	9	5	5	5
P1	0	1	1	1	2	2	3	3
P2	4	1	0	2	7	5	4	4
P3	1	0	0	1	3	3	3	2
P4	1	1	0	0	5	2	2	1
P5	1	0	1	1	4	4	4	4

$\boxed{\begin{matrix}7\\2\\3\\4\\4\\3\end{matrix}}$

1 ans)

Verify the Available array has been calculated correctly

Allocated resources

A = (2 + 0 + 4 + 1 + 1 + 1) = 9

B = (0 + 1 + 1 + 0 + 1 + 0) = 3

C = (2 + 1 + 0 + 0 + 0 + 1) = 4

D = (1 + 1 + 2 + 1 + 0 + 1) = 6.

Remaining available = Total - Allocated.

A = (15 - 9) = 6

B = (6 - 3) = 3

C = (9 - 4) = 5

D = (10 - 6) = 4

∴ Available (A, B, C, D) = (6, 3, 5, 4)

So, given available array is verified.

2) Calculate the Need matrix

The content of need matrix can be calculated using formula.: $\boxed{Need = Max - Allocation}$

Process	Need			
	A	B	C	D
P0	7	5	3	4
P1	2	1	2	2
P2	3	4	4	2
P3	4	3	3	1
P4	4	1	2	1
P5	3	4	3	3

Show that the current state is safe, i.e, show a

3) ~~for~~ safe sequence of processes. In addition, to the sequence show how the available (working array) changes as each process terminates

Sol:

For, process P_0, Need = (7, 5, 3, 4)

Available = (6, 3, 5, 4)

Need <= Available → False,

so, system will move to next process

For process P_1, Need = (2, 1, 2, 2)

Available = (6, 3, 5, 4)

Need <= ~~Allocation~~ Available → true.

Request for P_1 is granted

∴ Available = Available + Allocation

= (6, 3, 5, 4) + (0, 1, 1, 1)

Available = (6, 4, 6, 5), available array after termination of P_1

For process P_2, Need = (3, 4, 4, 2)

Available = (6, 4, 6, 5)

Need <= Available → true.

∴ Request for P_2 is granted

Available = Available + Allocation

$= (6, 4, 6, 5) + (4, 1, 0, 2)$

Available $= (10, 5, 6, 7)$, after termination of P_2

For process P_3, Need $= (4, 3, 3, 1)$

Available $= (10, 5, 6, 7)$

Need <= Available → True.

∴ Request for P_3 is granted.

Available = Available + Allocation

$= (10, 5, 6, 7) + (1, 0, 0, 1)$

$= (11, 5, 6, 8)$ (after termination of P_3

For process P_4, Need $= (4, 1, 2, 1)$ Available $= (11, 5, 6, 8)$

~~Allocation~~ Need <= Available → True.

∴ Request for P_4 is granted

Available = Available + Allocation

$= (11, 5, 6, 8) + (1, 1, 0, 0)$

$= (12, 6, 6, 8)$, after termination of P_4

For process P5, Need = (3, 4, 3, 3)

Available = (12, 6, 6, 8)

Need <= Available → True.

∴ P5 request is granted.

Available = Available + Allocation

= (12, 6, 6, 8) + (1, 0, 1, 1)

= (13, 6, 7, 9), after termination of P5

For process P0, Need = (7, 5, 3, 4)

Available = (13, 6, 7, 9)

Need <= Available → true

∴ P0 request is granted

Available = Available + Allocation

= (13, 6, 7, 9) + (2, 0, 2, 1)

Available = (15, 6, 9, 10), after termination of P0

So, All processes are completed & system is in safe state

Safe sequence: P1, P2, P3, P4, P5, P0

d) P5 = (3, 2, 3, 3)

Ans given request of P5 can be granted because,

we have Available = (15, 6, 9, 10)

Need = (3, 2, 3, 3)

As, Need <= Available.

Request of P5 can be granted.