M.SATHISH KUMER
E0119052

BANKERS ALGORITHM

1. Available Resources => 1 5 2 0

	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P_0	0	1	1	0	0	2	1	0	1	5	2	0
P_1	1	2	3	1	1	6	5	2				
P_2	1	3	6	5	2	3	6	6				
P_3	0	6	3	2	0	6	5	2				
P_4	0	0	1	4	0	6	5	6				

* Need Matrix:

Process	A	B	C	D
P_0	0	1	0	0
P_1	0	4	2	1
P_2	1	0	0	1
P_3	0	0	2	0
P_4	0	6	4	2

★ For process P_0,

Resource Available => A B C D
1 5 2 0

Resource To be provided => A B C D
0 1 0 0

Here resource to be provided is comparatively a subset of Resource available

So the process can be executed without interruption

A	B	C	D	
1	5	2	0	(available)
0	1	1	0	(assigned for P_0)
1	6	3	0	

★ For process P_1,

Resource Available => A B C D
1 6 3 0

Resource to be Provided => 0 4 2 1

∴ Process P_1 can't be performed

★ For process P_2,

Resource Available => A B C D
1 6 3 0

Resource to be provided => 1 0 0 1

∴ Process P_2 can't be performed

★ For process P_3;

Resource Available ⇒ A B C D
1 6 3 0

Resource to be provided ⇒ 0 0 2 0

Here resource to be provided < Resource available

A	B	C	D	
1	6	3	0	(available)
0	6	3	2	(allocated)
1	12	6	2	

★ For Process P_1 again,

Resource Available ⇒ A B C D
1 12 6 2

Resource Provided ⇒ 0 4 2 1

Here resource to be provided < Resource available

A	B	C	D	
1	12	6	2	(available)
1	2	3	1	(allocated)
2	14	9	3	

★ For process P_2,

Resource Available ⇒ A B C D
2 14 9 3 (available)

Resource Provided ⇒ 1 3 6 5 (allocated)

3 17 15 8

* For process P_4 :

A	B	C	D	
3	17	15	8	(available)
0	0	1	4	(allocated)
3	17	16	12	

Max instances of Type A = 3

Max instances of Type B = 17

Max instances of Type C = 16

Max instances of Type D = 12