(1) $E = \begin{vmatrix} 2 & 4 & 1 & 4 & 4 \end{vmatrix}$ $A = \begin{vmatrix} 0 & 1 & 0 & 2 & 1 \end{vmatrix}$

$$C = \begin{vmatrix} 0 & 1 & 1 & 1 & 2 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 2 & 1 & 0 & 0 & 0 \end{vmatrix} \quad R = \begin{vmatrix} 1 & 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 2 & 1 \\ 0 & 2 & 0 & 3 & 1 \\ 0 & 2 & 1 & 1 & 0 \end{vmatrix}$$

Satisfaction,
P1: NO
P2: YES

P2 is scheduled run. After that,

$$A = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

$$C = \begin{vmatrix} 0 & 1 & 1 & 1 & 2 \\ 0 & 2 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 2 & 1 & 0 & 0 & 0 \end{vmatrix}$$

P2 runs to completion.

$$A = \begin{vmatrix} 0 & 2 & 0 & 3 & 1 \end{vmatrix}$$

$$C = \begin{vmatrix} 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 2 & 1 & 0 & 0 & 0 \end{vmatrix} \quad R = \begin{vmatrix} 1 & 1 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 3 & 1 \\ 0 & 2 & 1 & 1 & 0 \end{vmatrix}$$

Satisfaction,
P1: NO.
P3: YES.

P3 is let to run. After that, A and C becomes,

$$A = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

$$C = \begin{vmatrix} 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 3 & 2 \\ 2 & 1 & 0 & 0 & 0 \end{vmatrix}$$

P3 runs to completion. After that, A, C, R becomes,

$$A = \begin{vmatrix} 0 & 2 & 0 & 3 & 2 \end{vmatrix}$$

$$C = \begin{vmatrix} 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \end{vmatrix} \qquad R = \begin{vmatrix} 1 & 1 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 1 & 1 \end{vmatrix}$$

At this stage, none of P1 or P4 have available required resources. They are deadlocked.

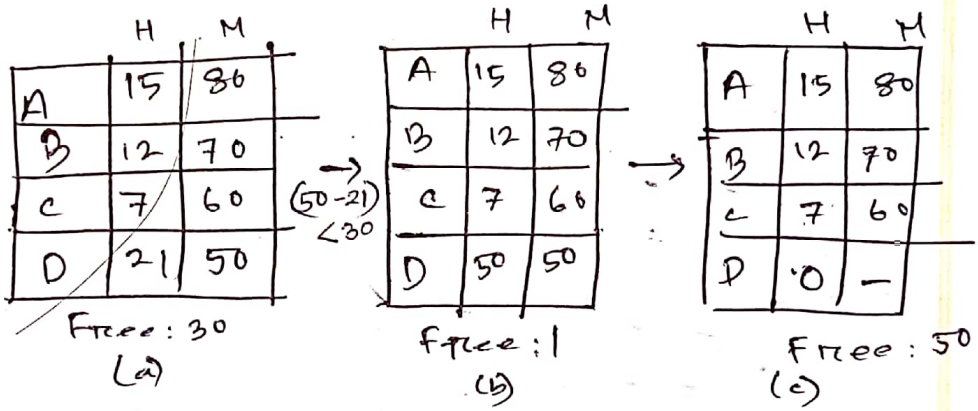
So, there is deadlock is this system. P1 and P4 are deadlocked.

(2)

a) For safe state, there is at least one scheduling order of processes ~~exist~~ exists so that all of the processes can run to completion, even if their maximum requests for resources come immediately.

For unsafe state, such scheduling order does not exist. System never tries to move to an unsafe state.

b)



Here we donot have enough free resource to further forward. So, we could not make a scheduling order. That's why, given state is unsafe state.

(c) Example, Scenario can be ~~from~~ starting from given state,

assume that all of processes released their currently allocated resource right after starting. So,

	H	M
A	15	80
B	12	70
C	07	60
D	21	50

all released →

	H	M
A	0	80
B	0	70
C	0	60
D	0	50

Free: 85

From here, all processes can complete execution.

Scheduling order can be,

A → B → C → D.