

# Practice

	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P <sub>0</sub>	1	0	1	2	1	1	2	1	1
P <sub>1</sub>	2	1	2	5	4	4			
P <sub>2</sub>	3	0	0	3	1	1			
P <sub>3</sub>	1	0	1	1	1	1			

Available :

A	B	C
2	1	1

Finish :

F	F	F	F
---	---	---	---

	Allocation			Max			Need		
	A	B	C	A	B	C	A	B	C
P0	1	0	1	2	1	1	1	1	0
P1	2	1	2	5	4	4	3	3	2
P2	3	0	0	3	1	1	0	1	1
P3	1	0	1	1	1	1	0	1	0

Work:

A	B	C
2	1	1

Step 1: Work= {2, 1, 1}

As Need[P0] = {1, 1, 0} < Work => Process P0 is selected. So,

Work + Allocation [P0] = {2, 1, 1} + {1, 0, 1} = {3, 1, 2}

Work:

A	B	C
2	1	1

$$\begin{array}{r}
 \text{Po} + \quad \quad 1 \quad 0 \quad 1 \\
 \hline
 \text{Work} = \quad 3 \quad 1 \quad 1
 \end{array}$$

Finish Array :

T	F	F	F
---	---	---	---

Step 2: Work= {3, 1, 2}

As Need[P1] = ( 3, 3, 2) > Work => Process P1 is not selected.

Work Array remains same.

Finish Array :

T	F	F	F
---	---	---	---

Step 3: Work= {3, 1, 2}

As Need[P2] = ( 0, 1, 1) < Work => Process P2 is selected.

New Work Array is -

Work + Allocation [P2]= { 3, 1, 2 } + { 3, 0, 0 } = { 6, 1, 2 }

Work:

	A	B	C
	3	1	1
P2 +	3	0	0
<hr/>			
Work=	6	1	2

Finish Array :

T	F	T	F
---	---	---	---

Step 4: Work= {6, 1, 2}

As Need[P3] = ( 0, 1, 0) < Work => Process P3 is selected.

New Work Array is -

Work + Allocation [P3]= { 6, 1, 2 } + { 1, 0, 1 } = { 7, 1, 3 }

Work:

	A	B	C
	6	1	2
P3 +	1	0	1
<hr/>			
Work=	7	1	3

Finish Array :

T	F	T	T
---	---	---	---

Step 5: Work= {7, 1, 3}

As Need[P1] = ( 3, 3, 2) > Work => Process P1 is not yet selected.

Work Array remains same.

Finish Array :

T	F	T	T
---	---	---	---

Since we couldn't find a safe sequence, therefore the system is not in a safe state.