Name : ABID ALI

Student_no : 2019380141

Homework -7

Operating System

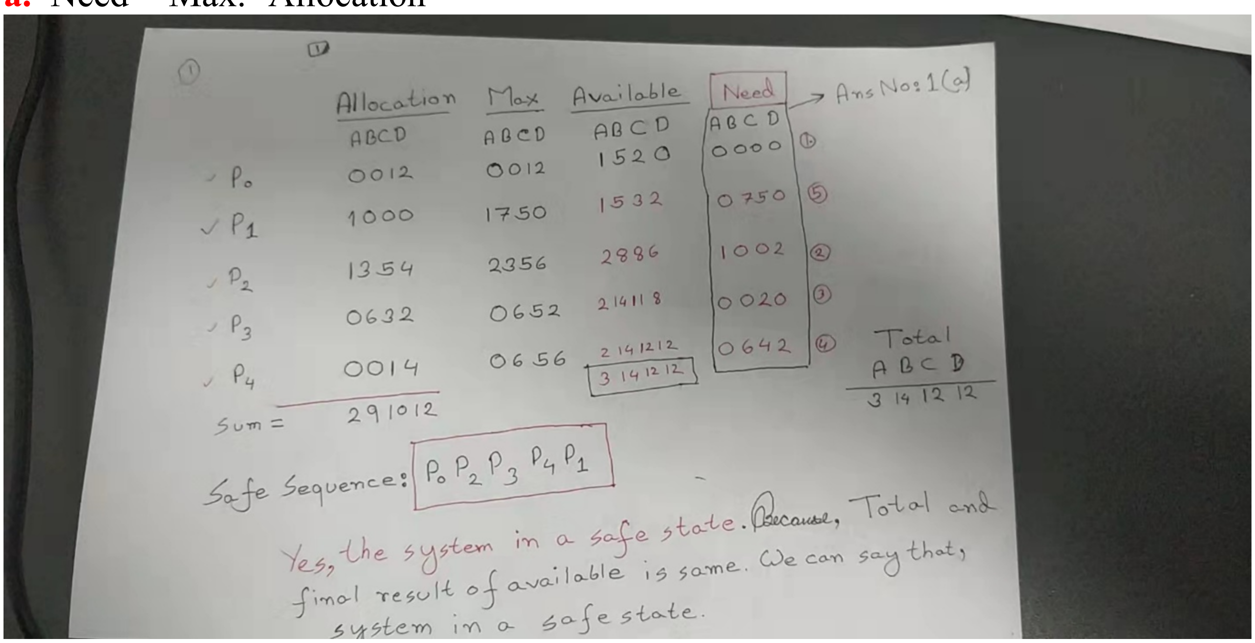
1 Consider the following snapshot of a system:

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

Answer the following questions using the banker's algorithm:
a. What is the content of the matrix Need?
b. Is the system in a safe state?
c. If a request from process P1 arrives for (0,4,2,0), can the request be granted immediately?

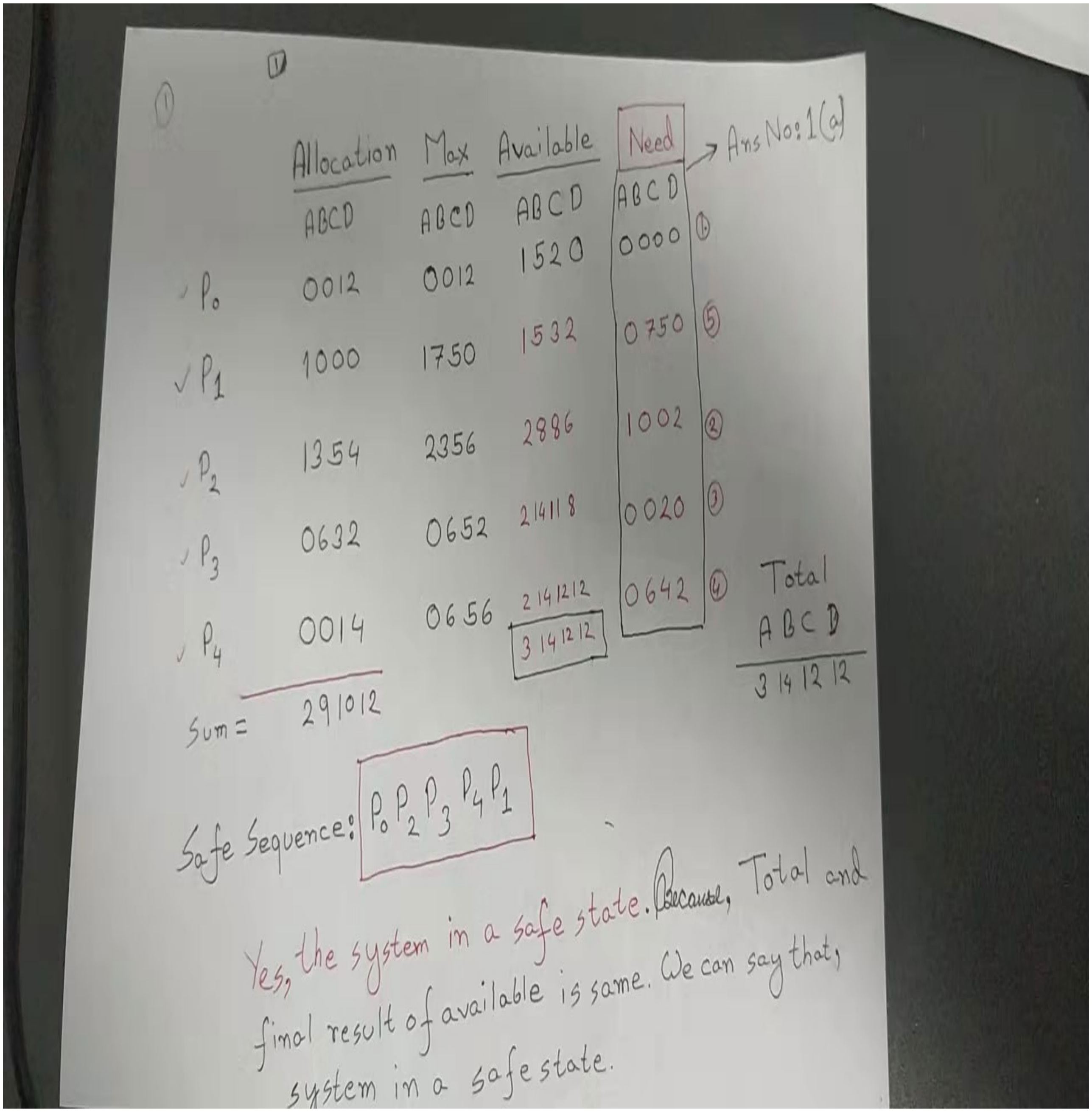
Answer No:1

a. Need = Max.- Allocation



The values of **Need** for processes *P*0 through *P*4 respectively are (0, 0, 0, 0), (0, 7, 5, 0), (1, 0, 0, 2), (0, 0, 2, 0), and (0, 6, 4, 2).

b. The system is in a safe state, Yes. With **Available** being equal to (1, 5, 2, 0), either process *P*0 or *P*3 could run. Once process *P*3 runs, it releases its resources, which allow all other existing processes to run.



(1)

	Allocation	Max	Available	Need
	ABCD	ABCD	ABCD	ABCD
			1520	0000 (1)
P_0	0012	0012	1532	0750 (5)
P_1	1000	1750	2886	1002 (2)
P_2	1354	2356	2 14 11 8	0020 (3)
P_3	0632	0652	2 14 12 12	0642 (4)
P_4	0014	0656	$\boxed{3\ 14\ 12\ 12}$	
Sum =	2 9 10 12			

→ Ans No: 1(a)

Total
ABCD
3 14 12 12

Safe Sequence: $\boxed{P_0\ P_2\ P_3\ P_4\ P_1}$

Yes, the system in a safe state. Because, Total and final result of available is same. We can say that, system in a safe state.

c.

(2)

$P_1 \rightarrow (0, 4, 2, 0)$

$Need(P_1) \rightarrow Max - Allocation$

$= 1750 - 1000$

$= 0750$

(1) Request $\leq$ need ✓

$0420 \leq 0750$

(2) Request $\leq$ available ✓

$0420 \leq 1520$

(3)

* available = available − request

$= 1520 - 0420$

$= 1100$

* allocation = allocation + request

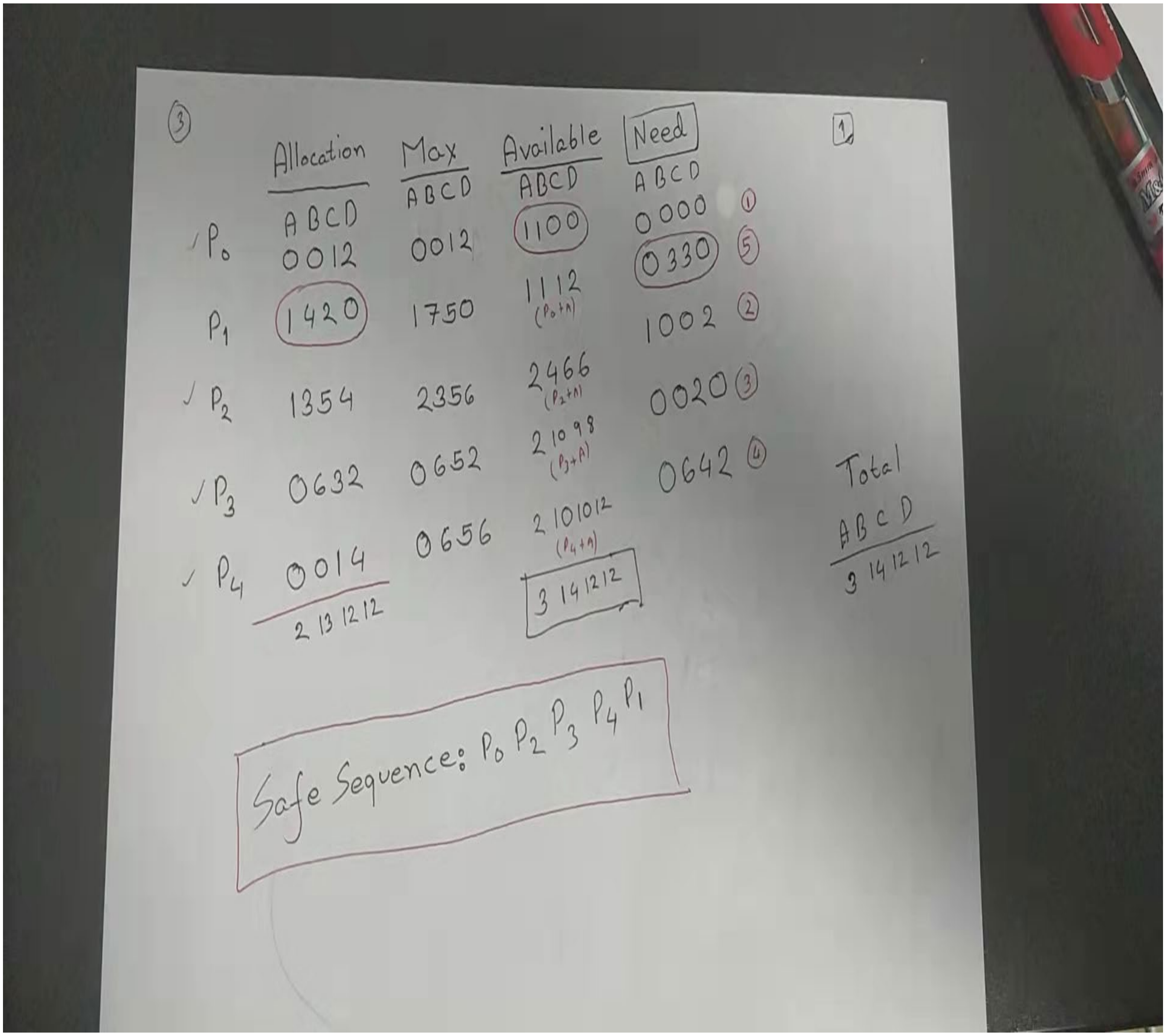
$= 1000 + 0420$

$= 1420$

* Need = Need − Request

$= 0750 - 0420$

$= 0330$



The request can be granted immediately. This results in the value of **Available** being (1, 1, 0, 0). One ordering of processes that can finish is *P*0, *P*2, *P*3, *P*1, and *P*4.

2 Consider the following snapshot of a system:

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

Answer the following questions using the banker's algorithm:
a. Illustrate that the system is in a safe state by demonstrating an order in which the processes may complete.

b. If a request from process P1 arrives for (1, 1, 0, 0), can the request be granted immediately?

c. If a request from process P4 arrives for (0, 0, 2, 0), can the request be granted immediately

Answer No:2

(a)

Max = Allocation + Need

	Allocation (A B C D)	Max (A B C D)	Available (A B C D)	Need (A B C D)	
P_0	2 0 0 1	4 2 1 2	3 3 2 1	2 2 1 1	①
P_1	3 1 2 1	5 2 5 2	5 3 2 2	2 1 3 1	④
P_2	2 1 0 3	2 3 1 6	6 6 3 4	0 2 1 3	⑤
P_3	1 3 1 2	1 4 2 4	7 10 6 6	0 1 1 2	②
P_4	1 4 3 2	3 6 6 5	10 11 8 7	2 2 3 3	③
Sum =	9 9 6 9		$\boxed{12\ 12\ 8\ 10}$		

Need = Max − Allocation

Total = Sum of Allocation + Available

Total			
A	B	C	D
12	12	8	10

Safe Sequence: $\boxed{P_0\ P_3\ P_4\ P_1\ P_2}$

⇒ Allocation + Available

Yes, System is in Safe State

(b)

$P_1 \rightarrow (1, 1, 0, 0)$

Need (P_1) → Max − Allocation

= 5252 − 3121

= 2131

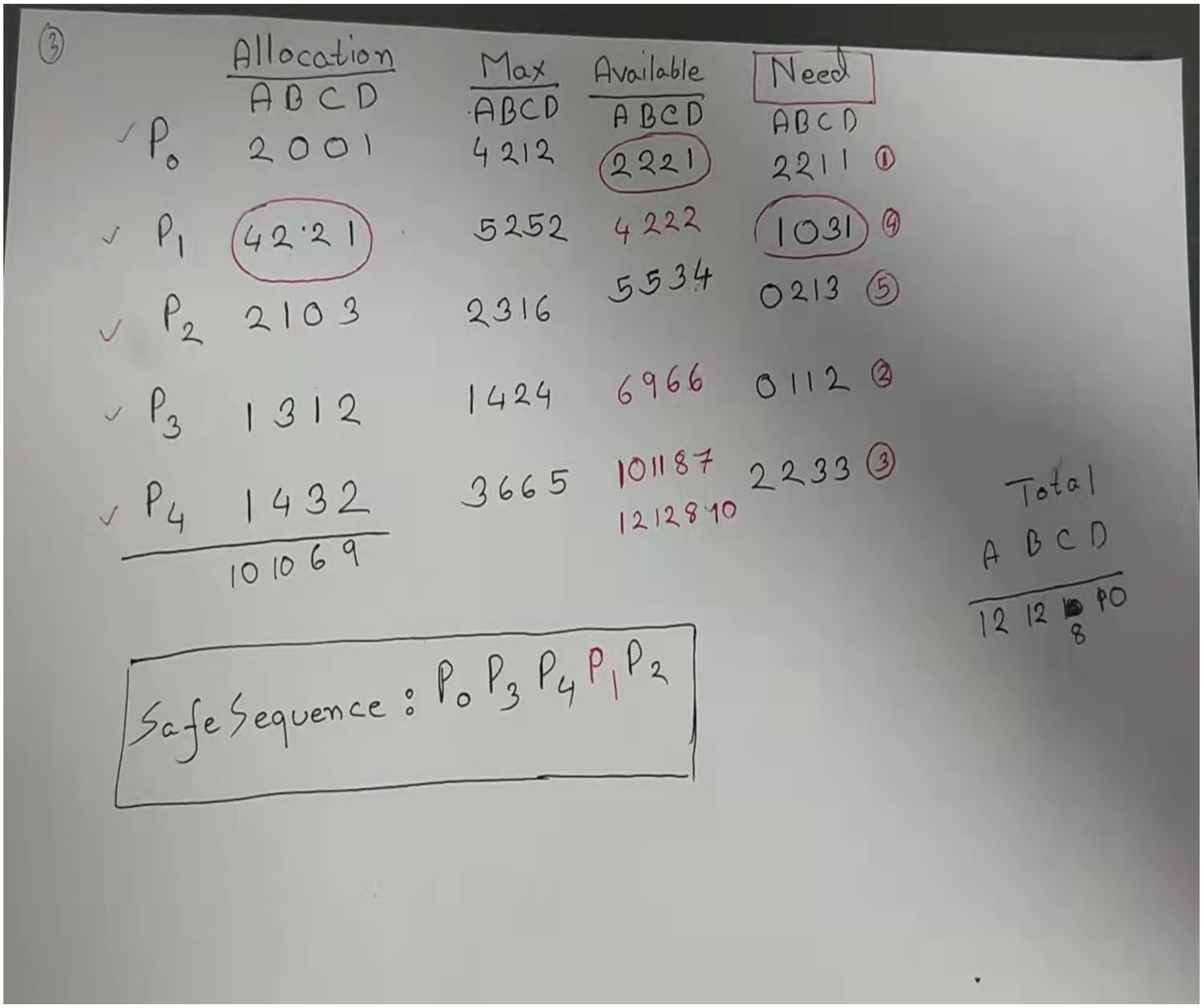
① Request ≤ need
1,100 ≤ 2131 ✓

② Request ≤ available
1100 ≤ 3321 ✓

③ * available = available − request
= 3321 − 1100
= 2221

* allocation = allocation + request
= 3121 + 1100
= 4221

* Need = Need − Request
= 2131 − 1100
= 1031



The request can be granted immediately. This results in the value of **Available** being (2, 2, 2, 1). One ordering of processes that can finish is *P*0, *P*3, *P*4, *P*1, and *P*2.

(c)

④ $P_4 \to (0,0,2,0)$ $\boxed{2}$

Need(P_4) → Max. − Allocation

= 3665 − 1432

= 2233

① Request ≤ need ✓

0020 ≤ 2233

② Request ≤ available ✓

0020 ≤ 3321

③ * available = available − request

= 3321 − 0020

= 3301

* allocation = allocation + request

= 1432 + 0020

= 1452

* Need = Need − Request

= 2233 − 0020

= 2213

	Allocation	Max	Available	Need
	A B C D	A B C D	A B C D	
P_0	2 0 0 1	4 2 1 2	2 2 1 3	2 2 1 1 (1)
P_1	3 1 2 1	5 2 5 2	4 2 1 4	2 1 3 1 (5)
P_2	2 1 0 3	2 3 1 6	6 3 1 7	0 2 1 3 (2)
P_3	1 3 1 2	1 4 2 4	7 6 2 9	0 1 1 2 (3)
P_4	1 4 5 2	3 6 6 5	8 10 7 11	2 2 1 3 (4)
	9 9 8 9		$\boxed{11\ 11\ 9\ 12}$	

Total

A B C D

$\boxed{11\ 11\ 9\ 12}$

$P_0\ P_2\ P_3\ P_4\ P_1$

The request can be granted immediately. This results in the value of **Available** being (2, 2, 1, 3). One ordering of processes that can finish is *P*0, *P*2, *P*3, *P*4, and *P*1.