DEPARTMENT OF SOFTWARE ENGINEERING

The University of Lahore

Faculty of Information Technology

Assignment Cover Letter

(Individual Work)

Student Name	Atta Elahi	Program	BS(SE)
SAP ID	70082385	Title of Assignment	03
Course Code	CS 11303	Due Date	26/12/2022
Course Name	Operating Systems	Submission Date	26/12/2022
Section	T		

The assignment should meet the below requirements:

1- Assignment (hard copy) is required to be submitted on clean paper and soft copy as per lecturer's instructions.
2- Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3- The above information is complete and legible.
4- Compiles pages are firmly attached.
5- Assignment has been copied (softcopy & hardcopy) for each student ahead of the submission.

Plagiarism/Cheating

The university seriously regards all forms of plagiarism, cheating and collision as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the University.

Declaration of Originality

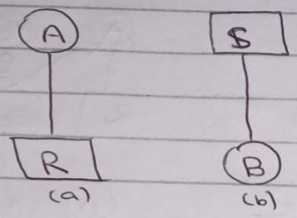
By signing this assignment, I understand, accept and consent to The University of Lahore terms and policies on plagiarism.

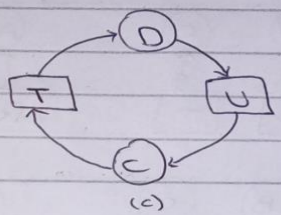
I hereby declare that this work represents my own effort, and that all text and code have been written by me and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature	

Question-1

Question 1.

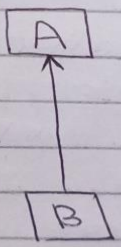




So, this process can complete until it gets the required resource and it can't released the already present resources before complete. So the system is in a deadlock.

This graph also helps us to make plan in order to prevent deadlock, we just have to prevent occuring of a loop in the graph.

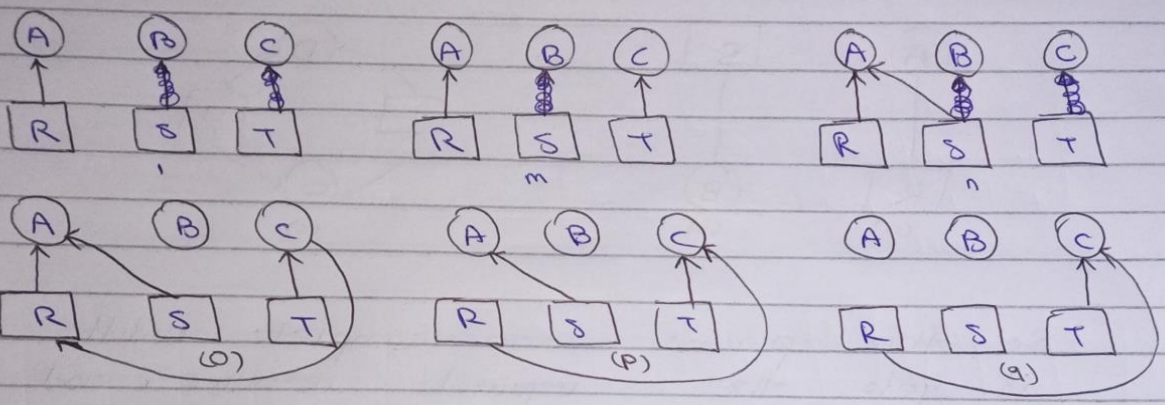
The resource graph can be illegal if one process is requesting for another process instead of a resource.



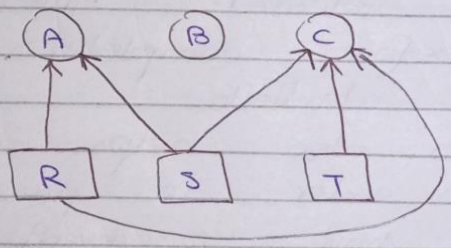
Here B process is requesting for A

Question-2

Question 2:



Step (o):

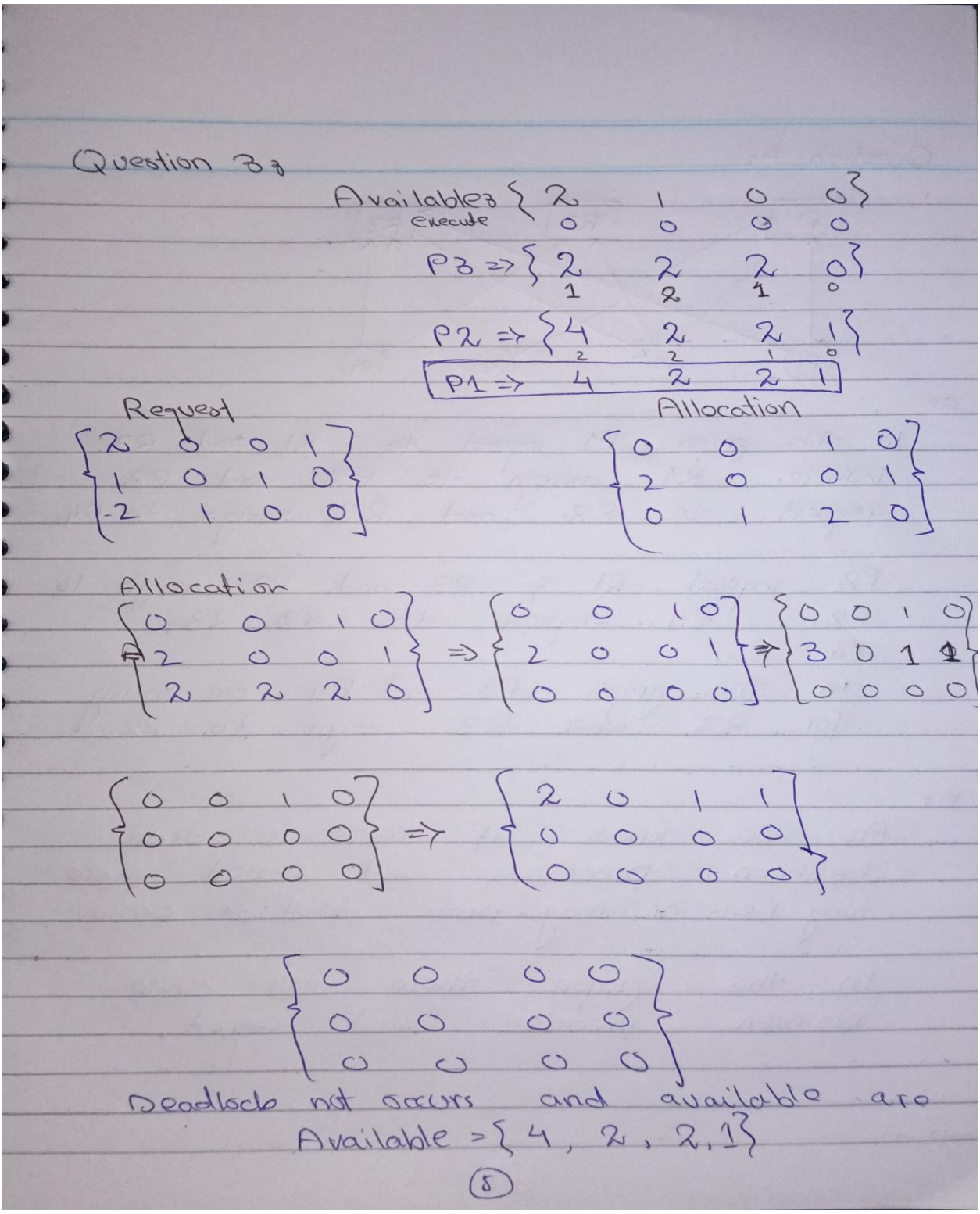


As we look C. request S instead of R but deadlock won't occurs because no cycle will be formed when an arrow is added from S to C.

As we added another arrow from R to C. In here deadlock lock won't occurs because no cycle will be formed. ~~when an arrow is added~~

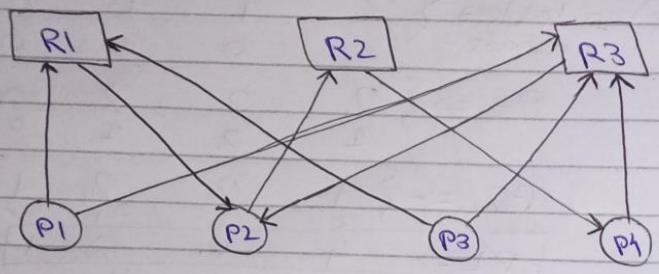
C can be access S after A released it.

Question-3


Question 3:
Available: { 2 1 0 0 }
execute { 0 0 0 0 }
P3 => { 2 2 2 0 } → { 1 2 1 0 }
P2 => { 4 2 2 1 } → { 2 2 1 0 }
P1 => 4 2 2 1
Request
Allocation
Allocation
Deadlock not occurs and available are
Available = { 4, 2, 2, 1 }
(5)

Question-4

Question 4:



a:

In this graph P1 request to R1 and R3. Where R1 assign to P2 and P2 request to R2 and R2 assign to P4.

P2 request R1 & R3 and R3 assign to P2. P4 request to R3.

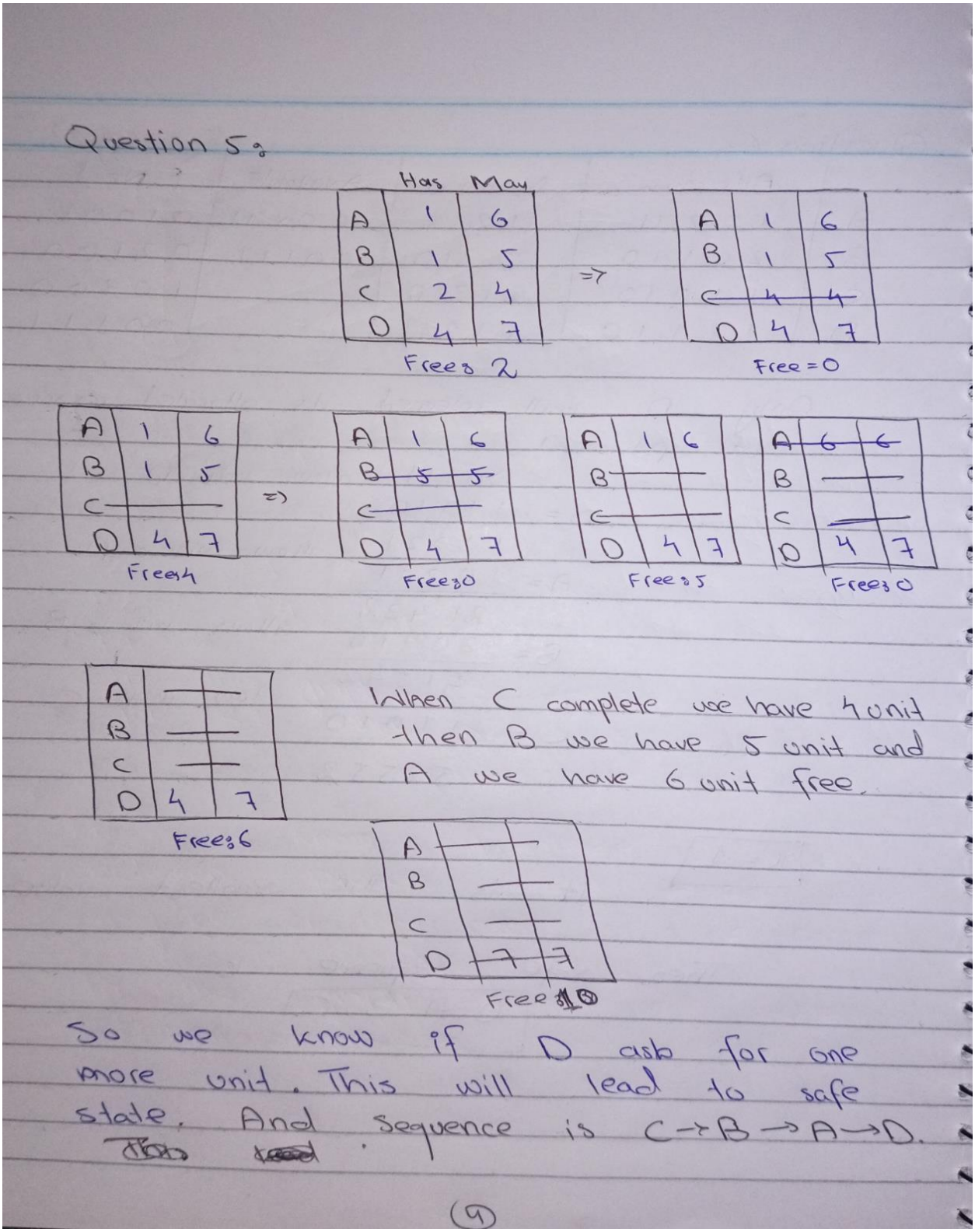
In this graph P3 and P4 are waiting for R3 wher R3 assign to P2.

b:

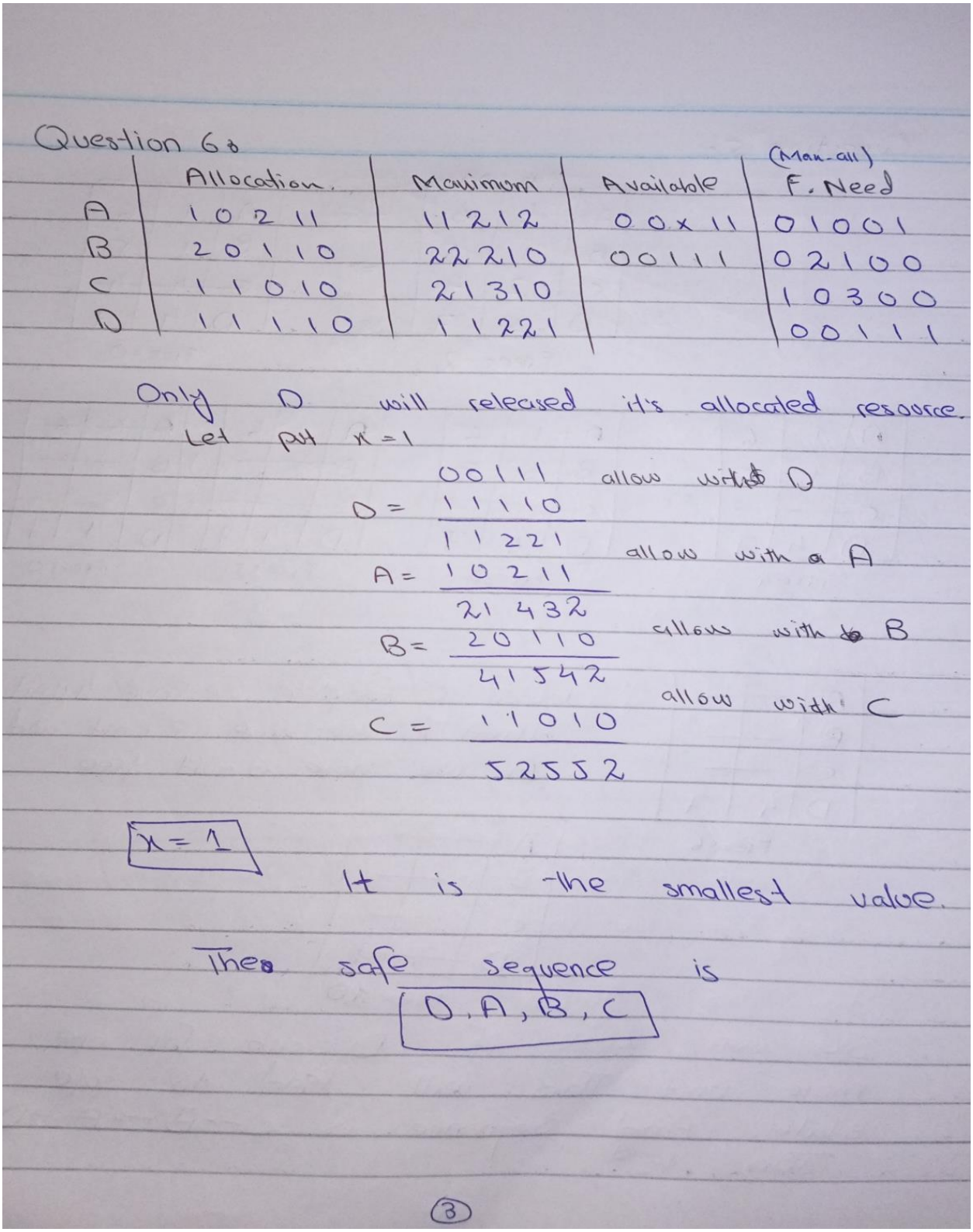
As we know if there is circle between processes and request there may be or may not deadlock occurs.

In this graph there will circle between processes and request.

Question-5


Question 5:
Has Max
A 1 6
B 1 5
C 2 4
D 4 7
Free: 2
=>
A 1 6
B 1 5
C 4 4
D 4 7
Free = 0
A 1 6
B 1 5
C
D 4 7
Free: 4
=>
A 1 6
B 5 5
C
D 4 7
Free: 0
A 1 6
B
C
D 4 7
Free: 5
A 6 6
B
C
D 4 7
Free: 0
A
B
C
D 4 7
Free: 6
When C complete we have 4 unit then B we have 5 unit and A we have 6 unit free.
A
B
C
D 7 7
Free: 10
So we know if D ask for one more unit. This will lead to safe state. And sequence is C → B → A → D.
(4)

Question-6



Question 6:

	Allocation	Maximum	Available	(Max-all) F. Need
A	1 0 2 1 1	1 1 2 1 2	0 0 x 1 1	0 1 0 0 1
B	2 0 1 1 0	2 2 2 1 0	0 0 1 1 1	0 2 1 0 0
C	1 1 0 1 0	2 1 3 1 0		1 0 3 0 0
D	1 1 1 1 0	1 1 2 2 1		0 0 1 1 1

Only D will released it's allocated resource.

Let put x = 1

D = 00111 + 11110 = 11221 allow with D

A = 10211 + 11221 = 21432 allow with A

B = 20110 + 21432 = 41542 allow with B

C = 11010 + 41542 = 52552 allow with C

x = 1

It is the smallest value.

The safe sequence is

D, A, B, C