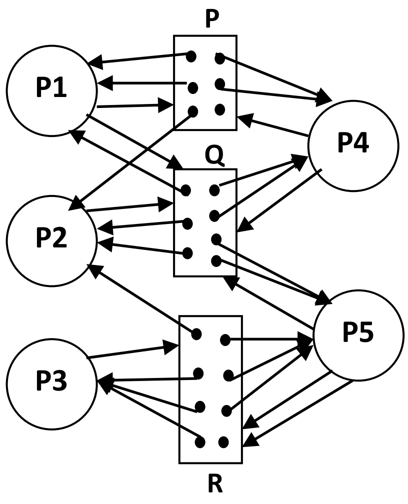
VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering
Fall Semester 2023-24 UG Seniors (B.Tech 2021 Batch)
Continuous Assessment Test – 2

SLOT: B1 + TB1, B2 + TB2

Programme Name &Branch: B.Tech& BCB, BCE, BCI, BCT, BDS, BKT
Course Name & code: Operating Systems & BCSE303L
Exam Duration: 90 Min. **Maximum Marks: 50**

General instruction(s): Answer all the questions

Q.No.	Question	Max Marks
1.	Consider the system with 5 processes (P1, P2, P3, P4 and P5) and three resource types (P, Q and R). The following resource allocation graph represents the current allocation and the requests of each process.  i) Check whether the current system is in safe state or not? If safe, print the safe sequence. If unsafe, what is the minimum number of resource instance is needed to be available for this system to be safe. (7 Marks) ii) With reference to the above safe system, suppose process P1 requests one additional instance from R, check whether it can be granted immediately or not? If yes, print the order in which the process are completing their execution.(3 Marks)	10
2.	Write the pseudo code of Peterson's method and analyze with justification each of the output for the below given scenarios. <table><tr><th>S.No</th><th>Interested[P0]</th><th>Interested [P1]</th><th>Turn</th></tr><tr><td>1</td><td>True</td><td>False</td><td>P0</td></tr><tr><td>2</td><td>True</td><td>True</td><td>P0</td></tr><tr><td>3</td><td>True</td><td>True</td><td>P1</td></tr><tr><td>4</td><td>False</td><td>True</td><td>P1</td></tr></table> The above table shows the interest of process P0 and P1 to enter the critical section ("True" denotes interested, "False" denotes not interested) and "Turn" denotes which process is currently under execution. Assume initially both processes are not interested to get into the critical section. (7 marks) Also justify whether mutual exclusion and progress is guaranteed for Peterson's method with an example for each. (3 Marks)	10
3.	a) Consider a speaker presenting a technical session and a maximum of 50 participants are allowed to attend the presentation. If the participant wants to interact with the speaker during the presentation, he/she must get the permission from the speaker. If the speaker agrees to interact, speaker stops his presentation and opens the session for discussion. Once the speaker closes the discussion session and resumes his presentation, the participants are not allowed to interact. The participants are	10

allowed to enter or exit the hall only during the discussion time. Discuss the need of process synchronization and develop the solution (pseudo code) to this problem. (6 Marks)

b) Apply binary semaphores to the following scenario given below. Consider two processes P_0 and P_1 are executed in sequence. Trace the algorithm and write the possible sequence of output. (4 Marks)

```
  mutex a,b;
  a=1, b=0;
        P0                   P1
 while (true) {        while (true) {
      down (a);             down (b);
      print ("3");          print ("2");
      up (b); }             up (a); }
```

4. a) Consider there are 4 partitions of size 4MB, 8 MB, 20 MB and 2 MB respectively. Assume that the partitions are released for the next job to occupy once the previous jobs have run to completion. The list of jobs and their appropriate time for execution is given below, — 10

Job ID	J1	J2	J3	J4	J5	J6	J7
Job size (in MB)	2	14	3	6	2	10	20
Time for execution	4	10	2	1	6	1	8
Completion time	?	?	?	?	?	?	?

Identify at what time each of the jobs will get over if "Best-fit" algorithm is applied to the above scenario with the help of a chart. Also identify whether there arises a situation of external fragmentation at any time of execution. (4 Marks)

b) Consider a system which is Byte addressable which contains total main memory as 64 Bytes. Assume that the first 4 frames of the main memory are already occupied by a system level process. Now a new process has arrived with size 32 Bytes which has to be accommodated into the main memory using paging technique. Assume the frame size = page size = 4 Bytes. Identify the location (with frame number and frame offset) where the 12th Byte of the process is stored in the main memory. Depict the above scenario with a detailed view of how the bytes are stored in the main memory, process table and page table. (6 Marks)

5. a)Suppose we have a page trace {4, 3, 2,1,4,3,5,4,3,2,1,5}. Use FIFO and LRU to run this page trace with 3 page frames and then with 4 page frames. For each replacement algorithm, clearly show the page replacement activities, number of page faults, hit ratio, and miss ratio. Also, indicate Belady anomaly if there is any for each of the replacement algorithms given. (7 Marks) — 10

b) A paging scheme uses Translation Lookaside Buffer (TLB). Suppose time taken to search a page entry in the TLB is 20 ns, time taken to access a byte/word from a main memory is 100 ns, and an effective memory access time is 160 ns. Assume, no page fault has occurred and two level page table is used .Determine the percentage of times that the page number of interest is found and not found in the TLB respectively. (3 Marks)

School of Computer Science and Engineering
Fall Semester 2023-24 UG Seniors (B.Tech 2021 Batch)

Continuous Assessment Test – 2

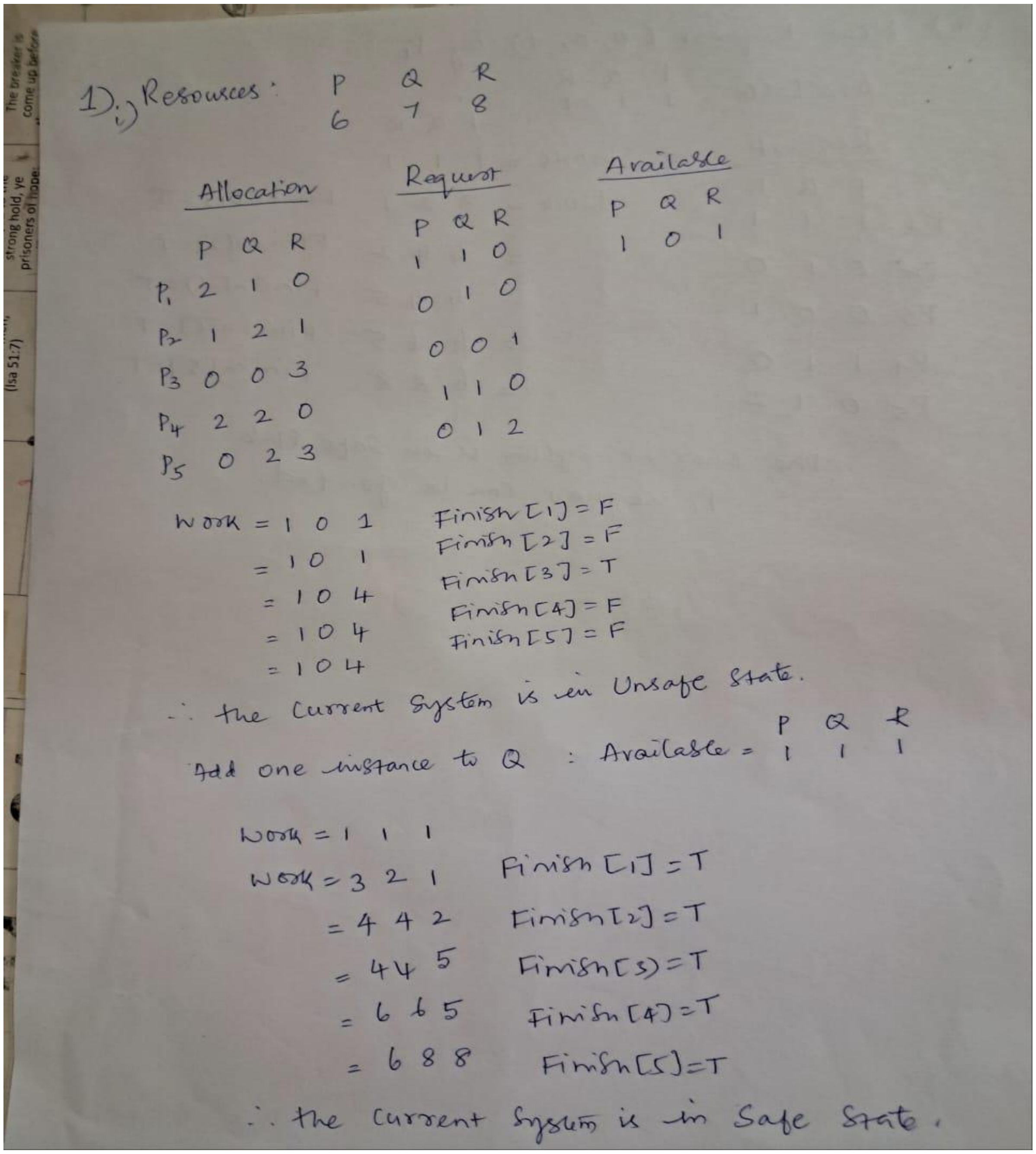
Course Code	: BCSE303L	**Duration**	**:** 90 Minutes.
Course Name	: OPERATING SYSTEMS		
Slot	: B1 + TB1, B2+TB2	**Max. Marks**	**:** 50M

Course Outcomes (CO):

CO 1	Interpret the evolution of OS functionality, structures, layers and apply various types of system calls of various process states.
CO2	Design scheduling algorithms to compute and compare various scheduling criteria.
CO 3	Apply and analyze communication between inter process and synchronization techniques.
CO 4	Implement page replacement algorithms, memory management problems and segmentation.
CO 5	Differentiate the file systems for applying different allocation, access technique, representing virtualization and providing protection and security to OS.

S. No	**Course Outcomes (CO's) Mapping (CO1-CO6)**	**Blooms Taxonomy (BL1 –BL6)**	**Marks Allotted**
Q1	CO3	BL3	**10**
Q2	CO3	BL4	**10**
Q3	CO3	BL3	**10**
Q4	CO4	BL4	**10**
Q5	CO4	BL3	**10**

1) Banker’s algorithm solution



1) i) Resources:

P	Q	R
6	7	8

	Allocation			Request			Available		
	P	Q	R	P	Q	R	P	Q	R
P_1	2	1	0	1	1	0	1	0	1
P_2	1	2	1	0	1	0			
P_3	0	0	3	0	0	1			
P_4	2	2	0	1	1	0			
P_5	0	2	3	0	1	2			

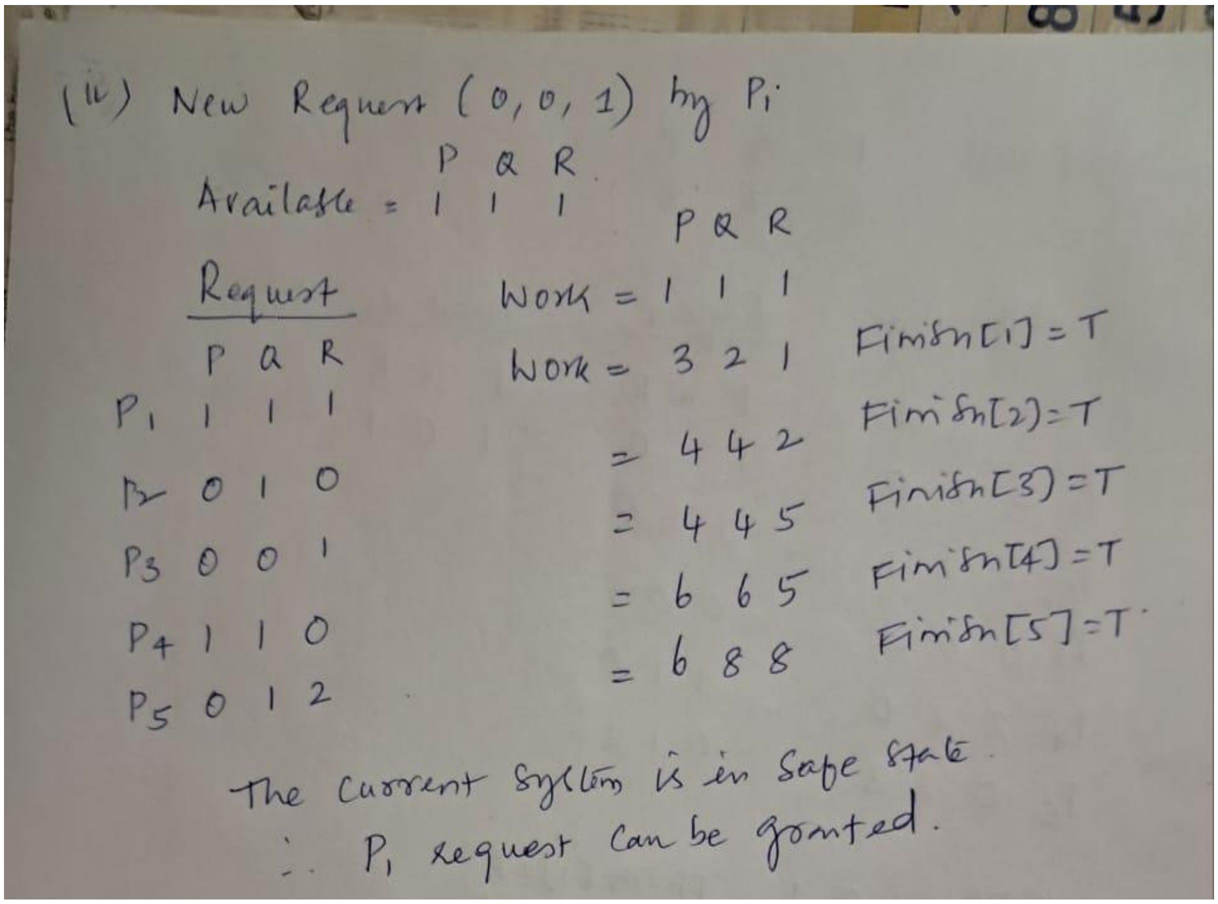
work = 1 0 1 Finish [1] = F
= 1 0 1 Finish [2] = F
= 1 0 4 Finish [3] = T
= 1 0 4 Finish [4] = F
= 1 0 4 Finish [5] = F
= 1 0 4

∴ the current System is in Unsafe State.

Add one instance to Q : Available = P Q R / 1 1 1

work = 1 1 1
work = 3 2 1 Finish [1] = T
= 4 4 2 Finish [2] = T
= 4 4 5 Finish [3] = T
= 6 6 5 Finish [4] = T
= 6 8 8 Finish [5] = T

∴ the current System is in Safe State.



(ii) New Request (0, 0, 1) by P_1

Available =

	P	Q	R
	1	1	1

Request

	P	Q	R
P_1	1	1	1
P_2	0	1	0
P_3	0	0	1
P_4	1	1	0
P_5	0	1	2

	P	Q	R	
Work =	1	1	1	
Work =	3	2	1	Finish[1] = T
=	4	4	2	Finish[2] = T
=	4	4	5	Finish[3] = T
=	6	6	5	Finish[4] = T
=	6	8	8	Finish[5] = T

The current system is in safe state.

∴ P_1 request can be granted.

Mark splitup

1) a) Allocation Table construction – 2 marks

To identify safe or unsafe – 2 marks

Identification of min resource – 1 mark

Conversion of unsafe to safe state – 2 marks

1) b) To identify whether P1 request can be granted or not - 3 marks

2) **Pseudo code of Peterson's Method** **(2 marks)**

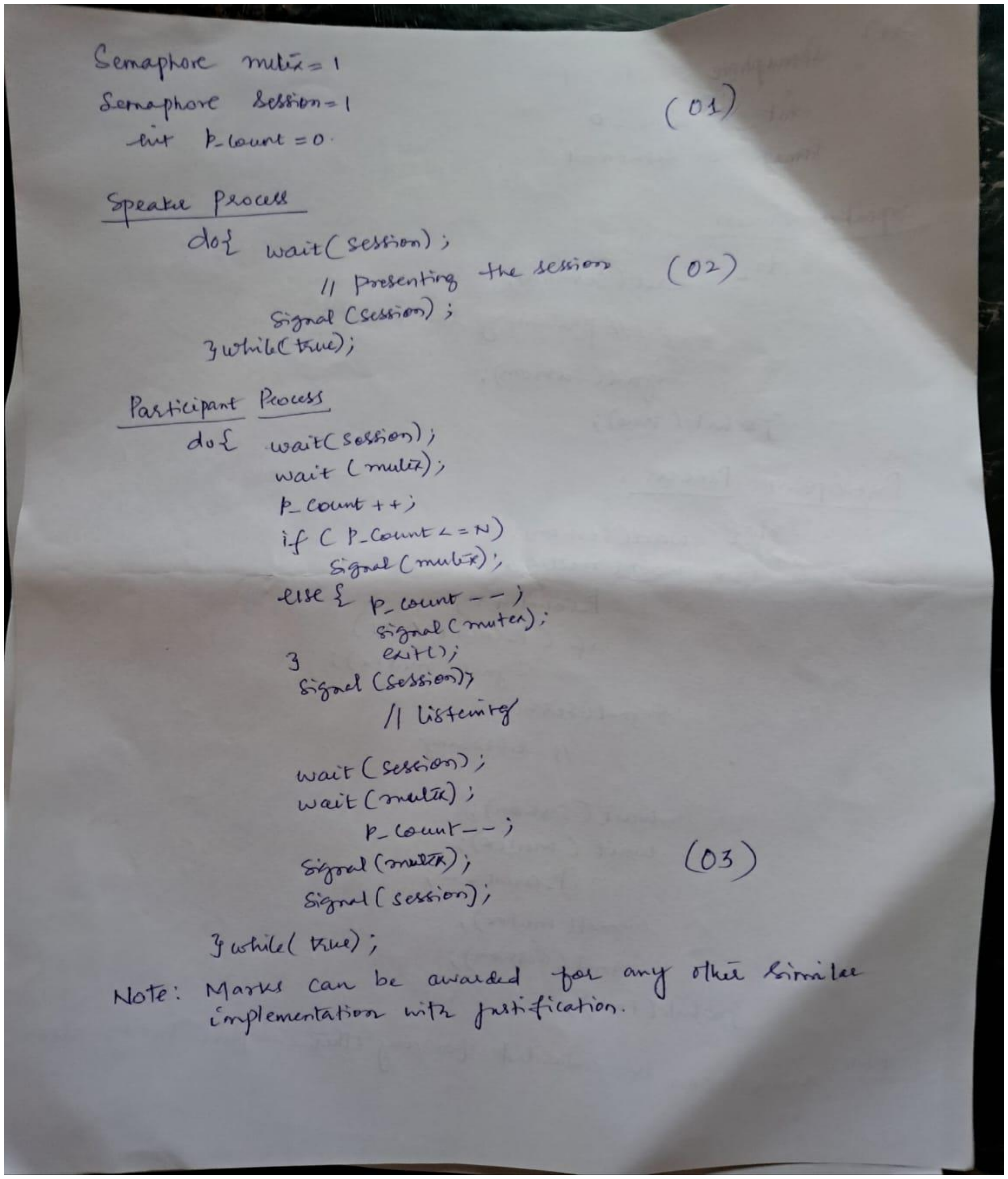
```
#define N2 // 'N' denotes total number of processes
#define TRUE 1
#define FALSE 0
int interested[N] = FALSE;
int turn;
void Entrysection (int process) {
   int other;
   other = 1 – process;
   interested [process] = TRUE;
   turn = process;
   while (interested [other] == TRUE  &&  turn = process);
 }
void Exitsection (int process) {
   interested [process] = FALSE;
}
```

S.No	**Interested[P0]**	**Interested [P1]**	**Turn**	**Output with justification (6 marks)**
1	True	False	P0	**Process P0 enters the critical section.** Justification: The other process P1 is not interested and it is the turn of P0, thus the while loop will break and the process P0 enters the critical section.
2	True	True	P0	**None of the processes will enter the critical section.** Justification:- It is the turn of process P0 and also the other process P1 is interested to enter the critical section. Thus, both the processes end up in an loop waiting for the other process to exit the critical section and change the interested value to FALSE.
3	True	True	P1	**None of the processes will enter the critical section.** Justification:- It is the turn of process P1 and also the other process P0 is interested to enter the critical section. Thus, both the processes end up in a loop waiting for the other process to exit the critical section and change the interested value to FALSE.
4	False	True	P1	**Process P1 enters the critical section.** Justification:- The other process P0 is not interested and it is the turn of P1, thus the while loop will break and the process P1 enters the critical section.

Mutual exclusion – Only one process must enter the critical section at a time.
Progress - If a process is not interested to enter the critical section, it should not stop the other process from entering the critical section.

Any example with justification given by the student which shows the mutual exclusion and progress is guaranteed for Peterson's method. **(2 marks)**

3) a)



```
Semaphore mutex = 1
Semaphore Session = 1                          (01)
int p_count = 0.

Speaker Process
    do{ wait(session);
            // Presenting the session          (02)
        Signal (session);
    }while(true);

Participant Process
    do{ wait(session);
        wait (mutex);
        p_count ++;
        if ( p_count <= N)
            Signal (mutex);
        else { p_count --;
               signal (mutex);
               exit();
        }
        Signal (session);
               // listening

        wait (session);
        wait (mutex);
             p_count--;
        Signal (mutex);                        (03)
        Signal (session);

    }while( true);
```

Note: Marks can be awarded for any other similar implementation with justification.

3) b) As per the concept of binary semaphore, the value of the semaphore is decremented by 1 to enter the critical section. Similarly when the process gets out of the critical section, the semaphore value is incremented back.

Iteration 1: a = 1; b = 0; 32
Iteration 2: a = 1; b = 0; 32
.
.
.

Thus, the output of the above example will be "32323232…….." (For multiple iterations) **(4 Marks)**

4)a)

Process execution chart when Best-fit algorithm is applied **(3 marks)**

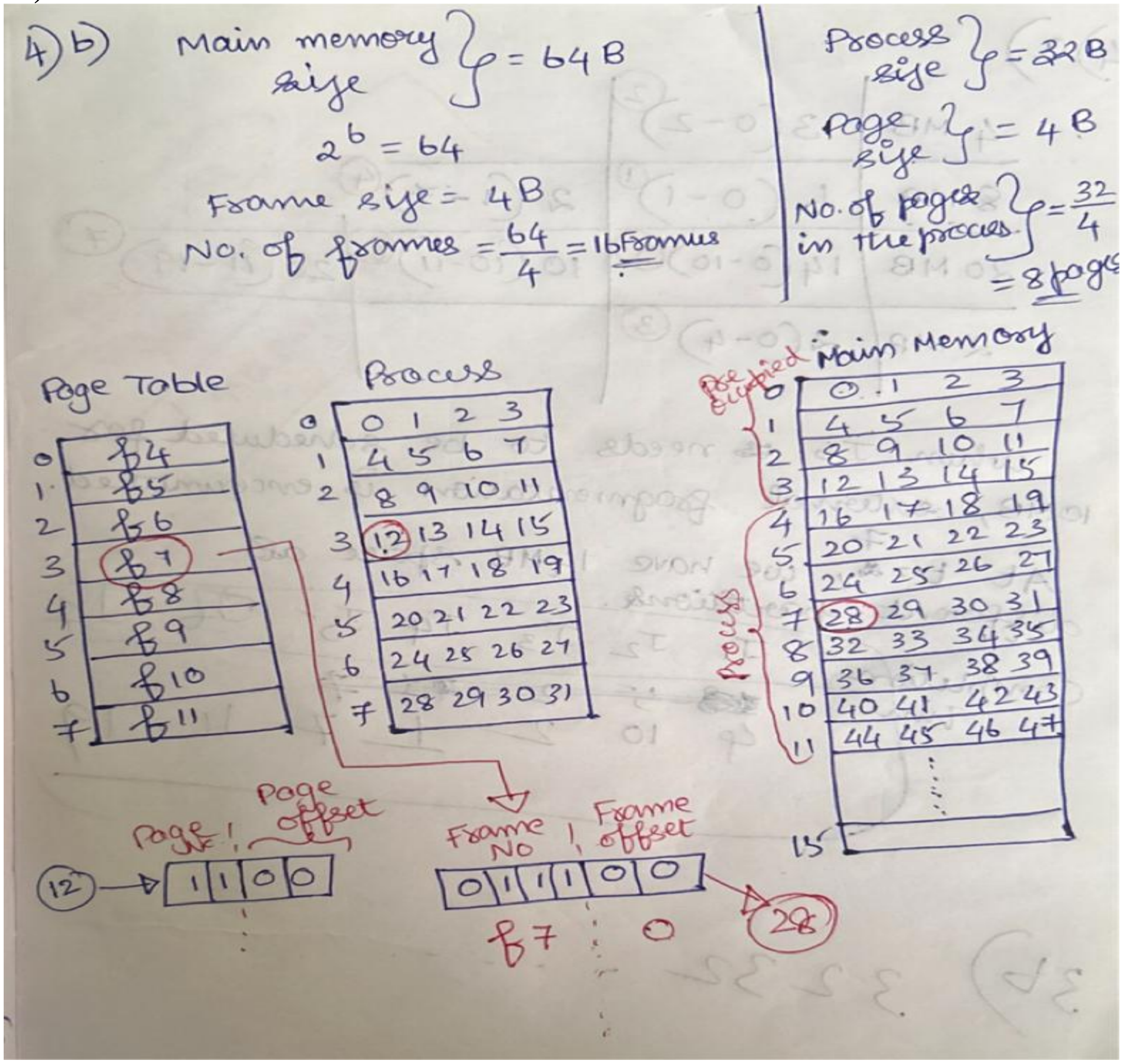
4 MB	J3 – 3 MB (0-2)		
8 MB	J4 – 6 MB (0-1)	J5 – 2 MB (1-7)	
20 MB	J2 – 14 MB (0-10)	J6 – 10 MB (10-11)	J7 – 20MB (11-19)
2 MB	J1 – 2 MB (0-4)		

When J6 – 10 MB needs to be scheduled, external fragmentation occurs. At time = 7, we have 14 MB of non-contiguous free space among all the partitions. (Or anyother instance given by the student) **(1 mark)**

Completion time based on Best-fit algorithm

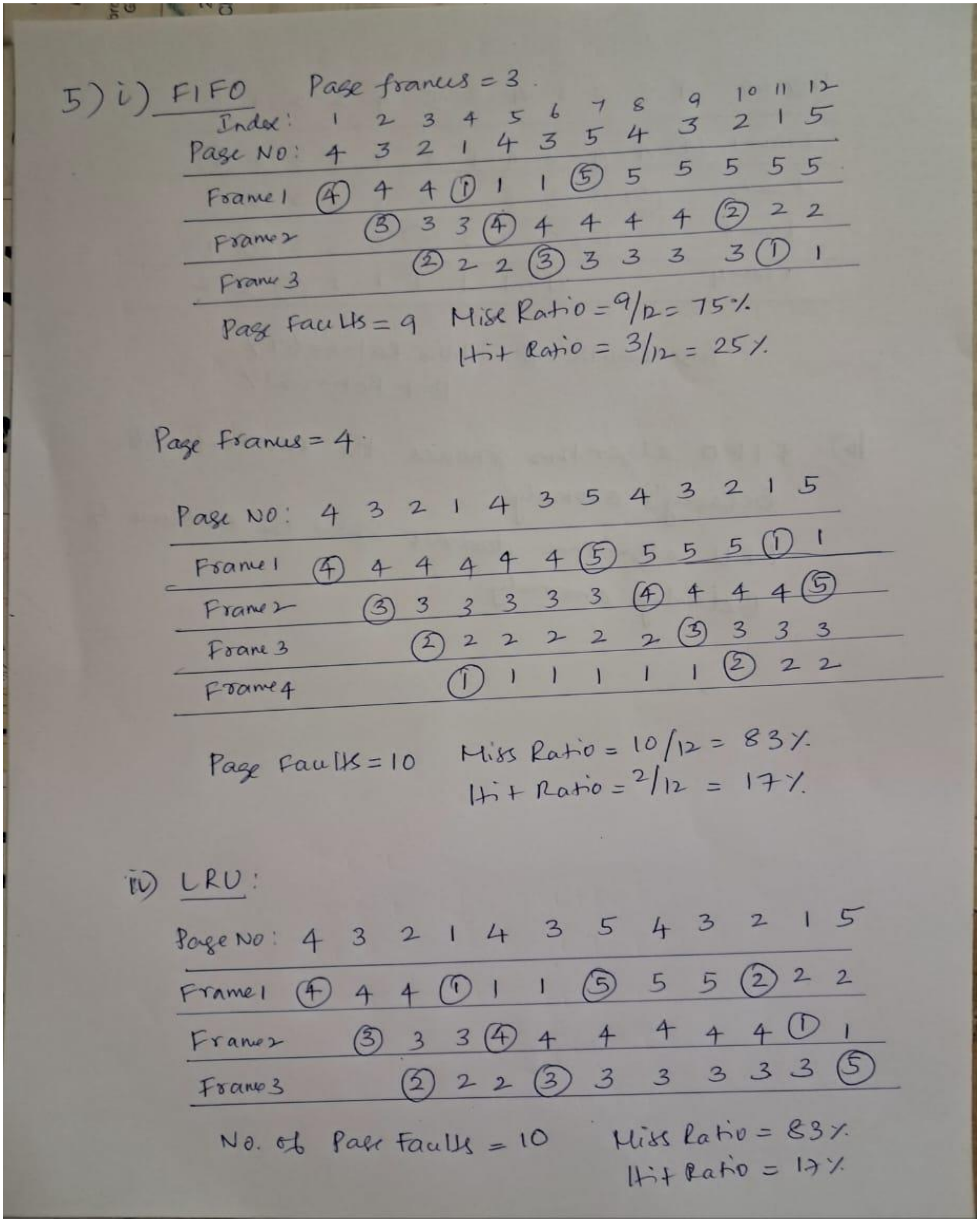
Job ID	J1	J2	J3	J4	J5	J6	J7
Job size (in MB)	2	14	3	6	2	10	20
Time for execution	4	10	2	1	6	1	8
Completion time	**4**	**10**	**2**	**1**	**7**	**11**	**19**

4 b)



- Finding out the number of frames and pages from the given data **(1 mark)**
- For the depiction of the detailed view (as given in the above image) of process table, page table and main memory. **(3 marks)**
- Identification of frame offset as 011100 – 28th byte in the main memory for 12th byte in the process. **(2 marks)**

5)a)



5) i) FIFO Page frames = 3.

Index:	1	2	3	4	5	6	7	8	9	10	11	12
Page No:	4	3	2	1	4	3	5	4	3	2	1	5
Frame 1	④	4	4	①	1	1	⑤	5	5	5	5	5
Frame 2		③	3	3	④	4	4	4	4	②	2	2
Frame 3			②	2	2	③	3	3	3	3	①	1

Page Faults = 9 Miss Ratio = 9/12 = 75%
Hit Ratio = 3/12 = 25%

Page Frames = 4.

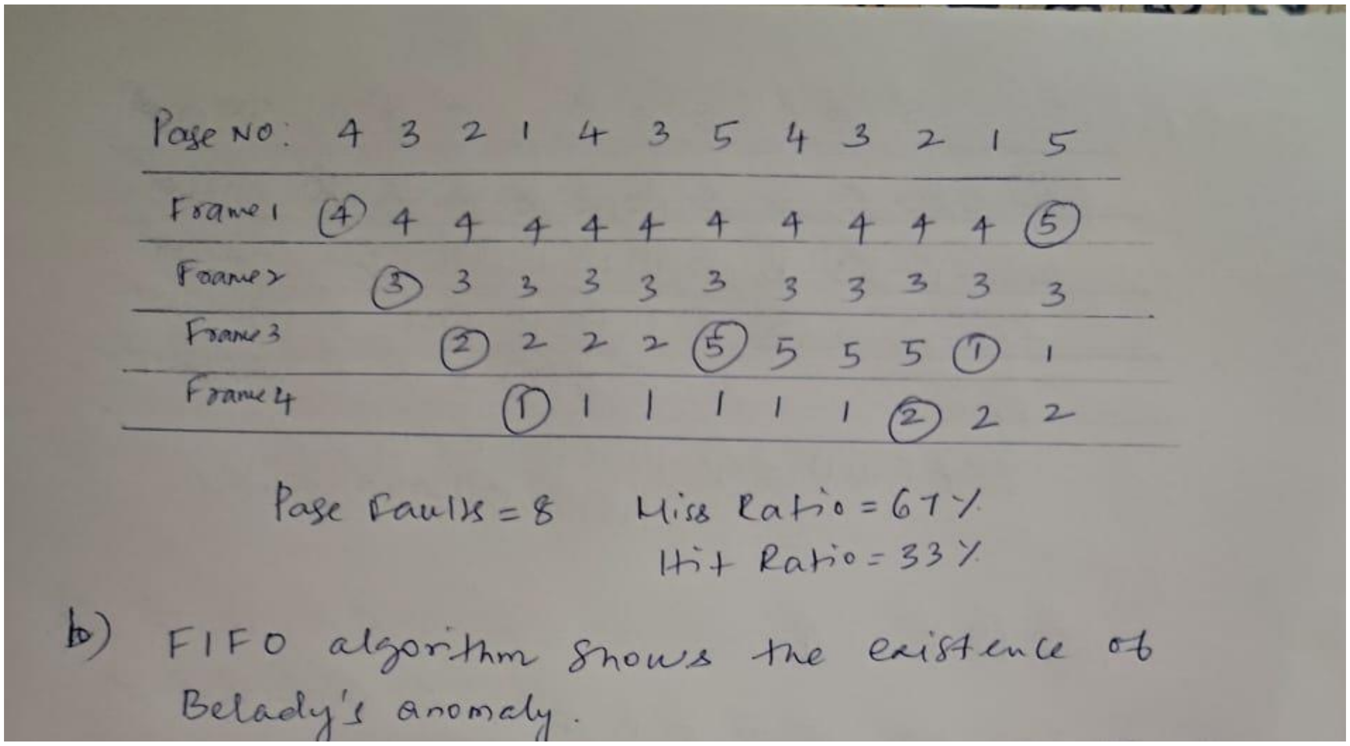
Page No:	4	3	2	1	4	3	5	4	3	2	1	5
Frame 1	④	4	4	4	4	4	⑤	5	5	5	①	1
Frame 2		③	3	3	3	3	3	④	4	4	4	⑤
Frame 3			②	2	2	2	2	2	③	3	3	3
Frame 4				①	1	1	1	1	1	②	2	2

Page Faults = 10 Miss Ratio = 10/12 = 83%
Hit Ratio = 2/12 = 17%

ii) LRU:

Page No:	4	3	2	1	4	3	5	4	3	2	1	5
Frame 1	④	4	4	①	1	1	⑤	5	5	②	2	2
Frame 2		③	3	3	④	4	4	4	4	4	①	1
Frame 3			②	2	2	③	3	3	3	3	3	⑤

No. of Page Faults = 10 Miss Ratio = 83%
Hit Ratio = 17%



Page No:	4	3	2	1	4	3	5	4	3	2	1	5
Frame 1	④	4	4	4	4	4	4	4	4	4	4	⑤
Frame 2		③	3	3	3	3	3	3	3	3	3	3
Frame 3			②	2	2	2	⑤	5	5	5	①	1
Frame 4				①	1	1	1	1	1	②	2	2

Page Faults = 8 Miss Ratio = 67%
Hit Ratio = 33%

b) FIFO algorithm shows the existence of Belady's anomaly.

Marks splitup

FIFO – Frames = 3 – 1.5 marks, Frames = 4 – 1.5 marks

LRU - Frames = 3 – 1.5 marks, Frames = 4 – 1.5 marks

Belady anomaly or not - 1 mark

5) b) Effective memory access time = p (t + m) + (1-p) (t + km + m) (1 mark)

160 = p (20+100) + (1-p)(20 + (2*100) + 100) (1 mark)

p = 0.8 => 80 % Hit 20 % Miss (1 mark)