Name: SHADAB IQBAL

ID: 1910172

Sec: 07

"I have not shared any answers with anyone, did not help anyone, and did not take help from anyone for this exam. The answers I have given here are done by myself only"

Ans. to QNo - 3 (b)

Time	Process 0	Process 1
0-3	flag[0] = true	
3-6	turn = 1	
6-9	while (false)	
	content switch	
9-12		flag[1] = true
12-15		turn = 0
15-18		while (True)
		content switch
18-21	Critical Section 1	
21-24	Critical Section 2	
24-27	flag[0] = false	
	content switch	
27-30		while (False)
30-33		Critical Section 1
33-36		Critical Section 2
		Content switch

36-39	Remainder Section 1
39-42	flag[0] = true
42-45	turn = 1
	content switch

Ans to QNo-3 (c)

Yes, we need to slightly modify the producer-consumer problem. Here, the waiters are the producer all the tables combined is the buffer and the guests are the consumers. In the given scenario all the tables are needed to be filled before the guests can start eating. So, basically all the indices of the buffer need to contain items first. After that, all the guests must complete eating, i.e all the items in the buffer are needed to be consum first, and only then, the food can be served again on all the tables, i.e, the producer can fill the whole buffer again. This process continues.

Ans to Q No-3 (d)

The three conditions:

1) **Mutual Exclusion:** If one process is in CS, no other process can't enter.

2) **Progress:** If CS is empty, one process requesting CS, will be able to enter CS.

3) **Bounded waiting:** A process can stay in CS for a limited time and can enter limited numer of times.

- **If mutual exclusion violated:**

  Many processes will be able to enter CS at a time which can conflict with the shared variable's value.

- **If progress requirement violated:**

  A process itself will not enter CS, neither will it allow any other processes to enter CS.

- **If Bounded waiting violated:** A process will keep entering CS again and again, so starvation will occur in other processes.

Ans to Q No- I (a)

i

Need matrix = Max - Allocated

T_0	0	2	2	0	0
T_1	1	0	0	2	0
T_2	0	2	0	3	0
T_3	1	0	0	2	0
T_4	3	0	0	1	0

Available resource matrix

A	B	C	D	E	
3	2	6	2	3	$\rightarrow T_1$
4	5	11	3	4	$\rightarrow T_3$
4	6	14	4	7	$\rightarrow T_4$
8	8	14	6	8	$\rightarrow T_0$
11	10	16	7	9	$\rightarrow T_2$

$\therefore$ Safe sequence $\Rightarrow T_1 \rightarrow T_3 \rightarrow T_4 \rightarrow T_0 \rightarrow T_2$

ii

T_1 requests → 0 4 2 0 2

T_1 needs → 1 0 0 2 0

So, as (requests > need), the request will be denied instantly. Because it violates the condit
so, if "requests" is added with "allocated," it will surely exceed "max". So, we won't ~~acc~~ accept the request.

Ans. to Q No - 1 (b)

i

	Allocation				Request				Available			
P_1	1	0	0	0	0	1	0	0	2	0	0	0
P_2	0	1	0	0	0	0	1	0	R_1	R_2	R_3	R_4
P_3	0	0	1	0	0	0	0	1				
P_4	0	1	0	1	1	0	0	0				
P_5	0	0	0	1	0	0	0	0				
	R_1	R_2	R_3	R_4	R_1	R_2	R_3	R_4				

ii

There is no deadlock.

Ans. to Q No - 1 (c)

Hold and wait ~~mmo~~ means that a process is being allocated by a resource, ~~att~~ and at that same time, it is requesting for another resource without pre-empting the previous resource.

Violating this is enough for absence of deadlock because, as violating this means releasing a resource, so another process will obviously be able to use that empty resource. Thus no deadlock.

Ans. to QNo - 2 (c)

- ☐ More flexibility on page size
- ☐ Uses less memory
- ☐ External fragmentation is avoided.
- ☐ Memory allocation simplified.

Ans. to QNo - 2 (b)

Page size = 2 KB = 2048 bytes

i) Page number = $\lfloor 1085 / 2048 \rfloor$

= 0

Page offset = 1085 % 2048

= 1085

ii) Page number = $\lfloor 30000 / 2048 \rfloor$

= 14

Page offset = 30000 % 2048

= 1328

iii) Page number = $\lfloor 19467 / 2048 \rfloor$

= 9

Page offset = 19467 % 2048

= 1035

iv) Page number = $\lfloor 15385 / 2048 \rfloor$

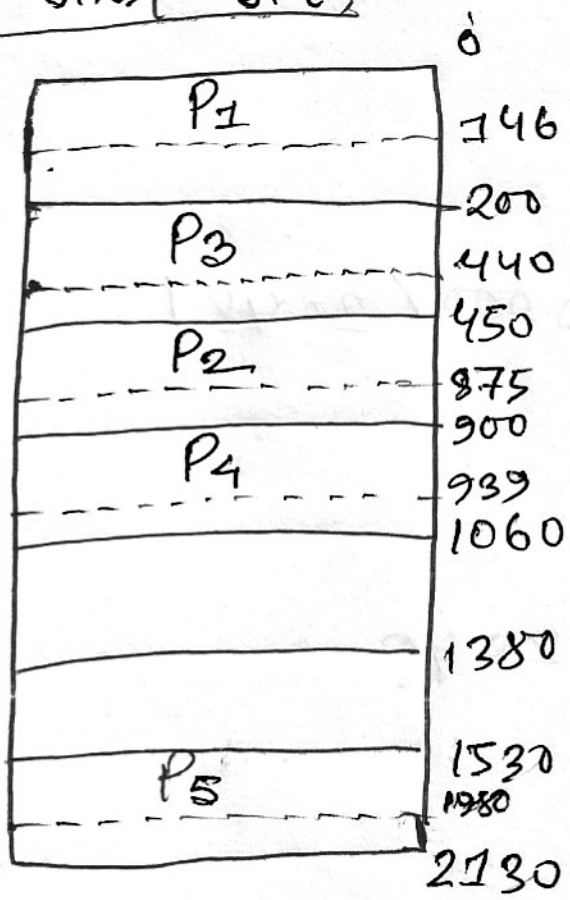
= 7

Page offset = 15385 % 2048

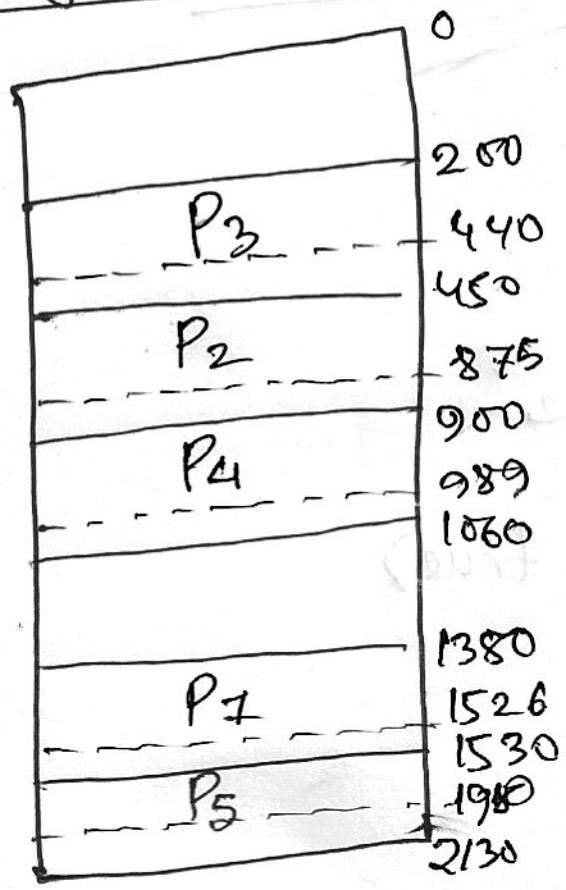
= 1049

Ans to QNo - 2 (a)

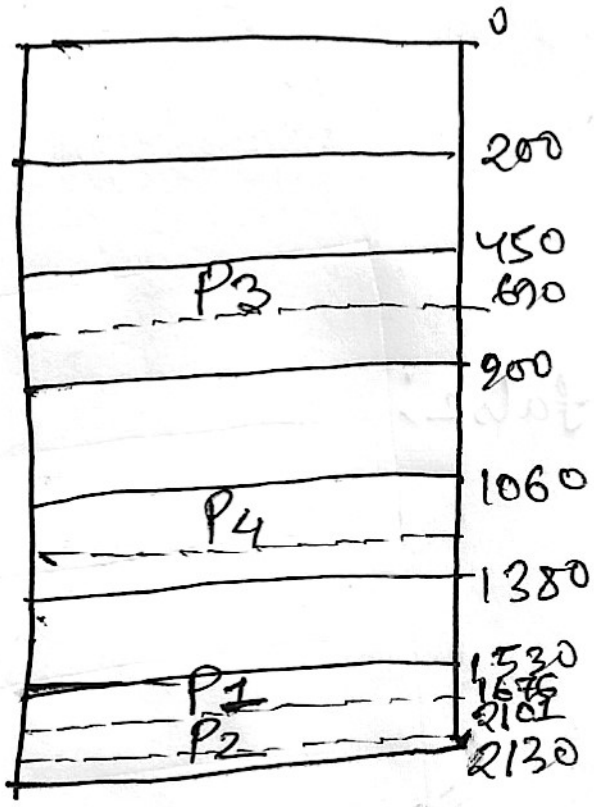
Applying first-fit,



Applying best-fit



Applying worst-fit



Here, best-fit is better because it takes up less space than first-fit.

and, worst-fit can't give space to P_5,

Ans. to QN:- 3(a)

```
do {
  wait(muten);
  ++read_count;
  i_kept_writer_waiting = true;
  if (i_kept_writer_waiting == true)
      wait(rw_muten);
  signal (muten);
  // reading done
  wait(muten);
--read_count;
  if (read_count == 0)
     signal (rw_muten);
  i_kept_writer_waiting = false;
  signal (muten);
} while (true);

// writer
  do {
     wait (rw_muten);
     writer_waiting = true;
     if (writer_waiting == true) {
        wait(muten);
     }
```

```
signal(rw_mutex);

// writing done

wait(rw_mutex);

writer_waiting = false;

signal(mutex);

} while(true);
```