Q. a

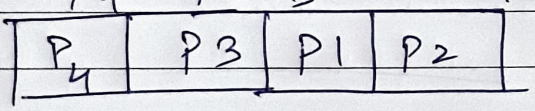
Process	Arrivaltime	Busst time	Finish time
P1	0	5	8
P2	0	10	18
P3	0	2	3
P4	0	1	1

SJF

	Turnaround time	Waiting time
P1	8	3
P2	18	8
P3	3	1
P4	1	0

Avg → 30/4 = 7.5 ms 12/4 = 3 ms

⇒



RR

Process	At	BT	FT	TAT	WT
P1	0	5	12	12	7
P2	0	10	18	18	8
P3	0	2	7	7	5
P4	0	1	4	4	3

RR

Avg. 41/4 23/4 = 5.75 ms

= 10.25 ms

5 10 2 1 4 9 1 3 8 2 7 1 6 5 4 3 2 1

P1 P2 P3 P4 P1 P2 P3 P1 P2 P1 P2 P1 P2 P2 P2 P2 P2 P2

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

4 7 12 18

	SJF	RR
Avg waiting time	3 ms	5.75 ms

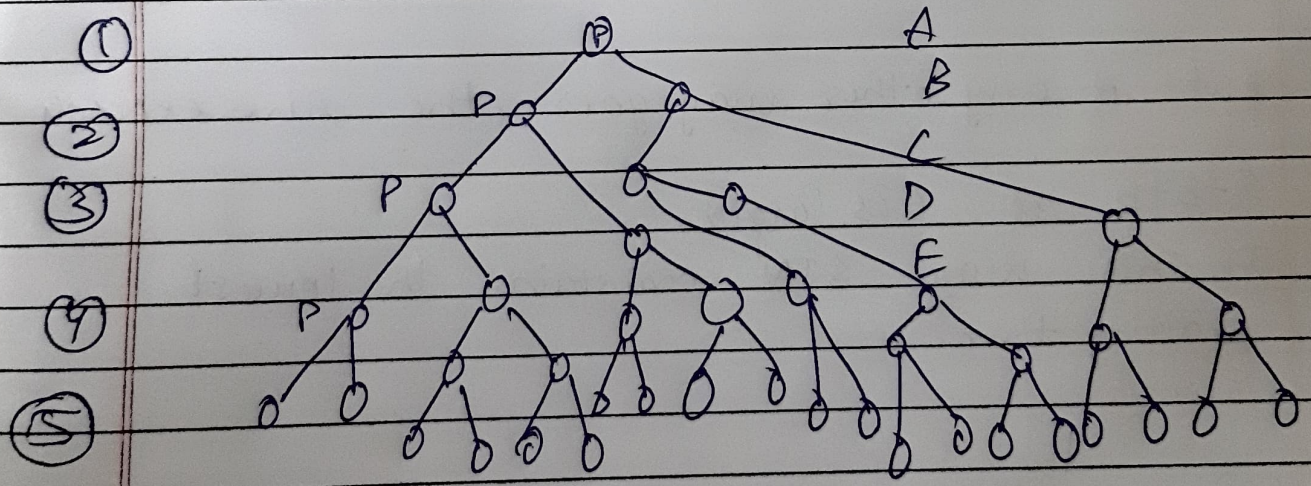
Q5-b

```
a()
a() && a()||a();
a().
```


-> A fork() call spawn processes as leaves grows binary tree.

∴ each call will ~~add~~ multiply by 2

```
a();          A
(a() &&       B
 a() ||       C
 a());        D
a();          E
```



-> At level 5 -> we will have 20 processes
The program will print "Created" 20 times

-> Overall 19 processes spawned

Q-c

In SJN or SJF Process we take the job/process with minimum required process time. as burst time, in the processes which are in ready queue.

Say we have three processes.

Process	bT	
A	a	a < b < c
B	b	
C	c	

In SJN -> input will be A -> B -> C

∴ waiting time will be of shortest processes i.e., a + b + b.

but if any other way goes the sum exceeds a + b + b as C is larger.

In this way SJN maintains the lowest waiting time.

Q-d

(a) In LRU -> page 1 will be replaced. as it is referenced at 256 (least recently).

(b). Page 1 again as it is loaded at 120 and the reference bit = 0. page 3 was considered first