

Q1) It is a classic example of deadlock. Neither P0 nor P1 can make progress because they are both waiting for a resource that the other process holds.

---

Q3) The initial system is in a safe state because all processes can complete their execution without deadlock. C has max claim of 9 instead of 7, the system would not be in a safe state. ~~the~~ because there would be enough resource to satisfy maximum claim.

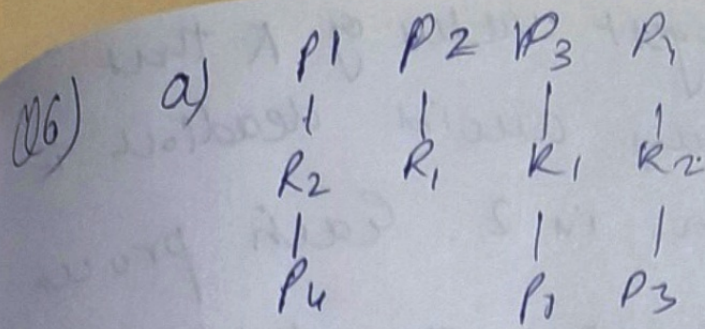
---

Q3) A and B share a semaphore.

A acquires, B waits but A never releases, causing both to wait indefinitely.

---





ii) Yes there is a cycle in graph.  
 $P_1 \rightarrow R_2 \rightarrow P_4 \rightarrow P_2 \rightarrow R_1 \rightarrow P_3$

c) Since there is a cycle it's deadlock.

Q8) To ensure that system is ~~deadlock~~ free, we need to make sure that it's impossible for any combination of process to simultaneously request all ~~available~~ available resource.

$$n = \frac{6 \text{ drivers}}{2 \text{ drivers/pro}} = 3 \text{ procs.}$$

So max value of  $n$  for the

system to be deadlock is 3

Q9) Deadlock is unlikely to occur because

the total maximum resource

needs of all processes are less than

the total number of resource available;

preventing circular waiting.



Q10) The largest value of  $K$  that will always avoid deadlock in this system is 2. Each process can request and hold a max of 2 instance, ensuring that at least one source remain available for other process to make progress and avoiding circular wait condition