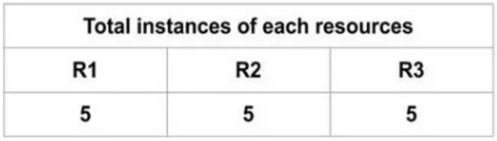
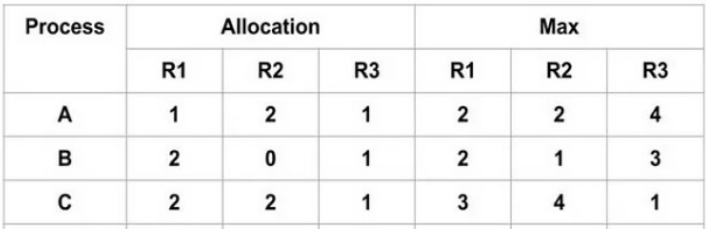
1. Compute TT and WT for the following processes using Hybrid (RR (TQ=2)+ Priority) scheduling. The selection of process is done as per the priority. After executing the process for the given time quantum, its priority is incremented by 1. Next process is selected based on the current priorities. Treat lowest value as highest priority. When same priorities found with more than once process, select next process by treating their PIDs as their priorities.

Process ID	**Burst time**	**Priority**
1	4	3
2	6	2
4	5	4
5	3	1
6	6	2

2. Determine whether the system is in safe state or not. Then find whether the requests for additional resources can be accepted. Each request should be evaluated based on the modified data after completing the previous requests and not based on the initial data given.



Total instances of each resources		
R1	R2	R3
5	5	5



Process	Allocation			Max		
	R1	R2	R3	R1	R2	R3
A	1	2	1	2	2	4
B	2	0	1	2	1	3
C	2	2	1	3	4	1

   a. C(1, 1, 0)
   b. B(0,1,0)
   c. A(0,0 1)

3. Consider a disk scheduling implementation in which the requests come dynamically at different times. So each request is also having an arrival time. Implement SSTF by choosing the nearest request among the currently arrived requests. Assume that it takes 1 unit of time to complete a request on disk. Initial head position is 55 and there are 200 tracks on the disk.

AT	0	0	1	1	1	2	3	3	4	5	5	6	7	7
Track	21	40	27	60	45	66	78	11	110	97	85	17	32	60

4. Implement page replacement based on FCFS with local replacement. Under this scheme, when a page needs to be replaced for loading a new page, one of the oldest pages of the same process will have to be replaced. Only if none the pages of that process is in memory, any page that is the oldest in memory has to be replaced.

Process	P1	P1	P2	P3	P1	P4	P4	P2	P5	P1	P6	P6	P1	P4	P4
Page no	1	2	3	4	1	5	6	3	7	2	8	7	1	5	6