Assignment 2

Problem 1: Implement preemptive priority scheduling with a single dispatch queue. Your scheduler should follow the below rules.

1. When a job is ready for execution (after arrival or after completion of I/O), it should be added to the ready queue and its priority should be set to **0** (zero).
2. Increase the priority of jobs in the ready queue at a constant rate ***alpha*** per unit time.
3. Increase the priority of jobs in the running state at a constant rate ***beta*** per unit time.
4. Your implementation should work for different values of ***alpha*** and ***beta***, for example-
    a. *alpha > beta, beta = 0*
    b. *beta > alpha > 0*
    c. *alpha > beta > 0*
5. While executing jobs, your scheduler should consider CPU bound and I/O bound.

Test your code with jobs given in the table below. Print the scheduling summary and number of context switches. Also print response time, turnaround time and waiting time (time spent in ready state) for each process. Analyze the output to understand merits and demerits of the three combinations of *alpha* and *beta* values.

Process Name	Arrival Time	CPU Burst	I/O Burst
P0	0 unit	18 units	5 units after every 5 units of CPU burst
P1	3 units	27 units	6 units after every 6 units of CPU burst
P2	8 units	44 units	4 units after every 6 units of CPU burst
P3	12 units	50 units	3 units after every 9 units of CPU burst

Problem 2: Implement multilevel queue with round robin scheduler within the individual queues. The scheduler will have three queues with different priorities and time slices as below.

Queue	Priority	Time slice
Q0	2	9 units
Q1	3	6 units
Q2	4	3 units

At the start, the scheduler should schedule the process from the highest-priority non-empty queue and allow the process to run for the time slice for that queue. When this time slice expires, select the next highest-priority process and repeat the process. Assume that a higher priority process that wakes up (after completing I/O) does not preempt a currently executing process. Also assume that your simple scheduler does not move process from one queue to another.

Test your scheduler with the same four processes as in problem 1, **with arrival time of all the processes as 0**, and with priorities as **P0 & P1 – 4 (highest priority)**, **P2 - 3** and **P3 - 2 (lowest priority)**. Print the scheduling summary and number of context switches. Also print response time, turnaround time and waiting time (time spent in ready state) for each process.