# CSE321
# CPU Scheduling
# Deadline: 14/08/2021

Given the list of processes, their CPU burst times, arrival times and priorities implement SJF, Priority and Round Robin scheduling algorithms on the processes with preemption. For each of the scheduling policies, compute and print the completion Time(CT), Turnaround Time(TAT), and Waiting Time(WT) for each process.

**Waiting time:** Processes need to wait in the process queue before execution starts and in execution while they get preempted.

**Turnaround time:** Time elapsed by each process to get completely served. (Difference between submission time and completion time).

## Task 1: SJF Scheduling
**You can use the following input as sample:**

| Process | Arrival Time | Burst Time |
|---------|--------------|------------|
| $P_1$ | 0 | 8 |
| $P_2$ | 1 | 4 |
| $P_3$ | 2 | 9 |
| $P_4$ | 3 | 5 |

**Solution in a Gantt chart:**

| $P_1$ | $P_2$ | $P_4$ | $P_1$ | $P_3$ |
|---|---|---|---|---|

0   1       5       10          17          26

## Task 2: Priority Scheduling
**Sample input:**

| Process | Burst Time | Priority |
|---------|------------|----------|
| $P_1$ | 10 | 3 |
| $P_2$ | 1 | 1 |
| $P_3$ | 2 | 4 |
| $P_4$ | 1 | 5 |
| $P_5$ | 5 | 2 |

**Solution shown by a Gantt chart:**

| $P_2$ | $P_5$ | $P_1$ | $P_3$ | $P_4$ |
|---|---|---|---|---|

0   1       6                   16      18  19

You may consult the following pseudocode to implement Priority Scheduling.

```
completed = 0
current_time = 0
while(completed != n) {
    find process with maximum priority time among process that are in ready queue at current_time
    if(process found) {
        if(process is getting CPU for the first time) {
            start_time = current_time
        }
        burst_time = burst_time - 1
        current_time = current_time + 1
        if(burst_time == 0) {
            completion_time  = current_time
            turnaround_time = completion_time - arrival_time
            waiting_time = turnaround_time - burst_time
            response_time = start_time - arrival_time
            mark process as completed
            completed++
        }
    }
    else {
        current_time++
    }
}
```
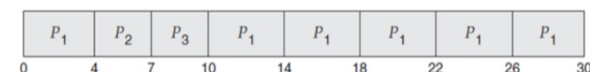
## Task 3: Round Robin
Time Quantum = 4 ms

Check for incoming processes after every time quantum (4 ms).

**Sample input:**

| Process | Burst Time |
|---------|------------|
| $P_1$ | 24 |
| $P_2$ | 3 |
| $P_3$ | 3 |

| $P_1$ | $P_2$ | $P_3$ | $P_1$ | $P_1$ | $P_1$ | $P_1$ | $P_1$ |
|---|---|---|---|---|---|---|---|

0       4       7       10      14      18      22      26      30

**N.B.**

Input for each task will be :

       Number of processes

       Arrival time of each process. If all processes arrive at the same time, this can be set to 0 for all processes.

       Burst time of each process

       Priority of each process (If required)

Output will be:

       Completion Time(CT), Turnaround Time(TAT), Waiting Time(WT)

       Average turnaround Time, average waiting time