Programming Project for Exam preparation

In this project you will implement the scheduling algorithm of the previously assigned Homework problem 5.18

for the following processes (the highest (best) priority is the largest integer number, P1 has highest priority.

The scheduling algorithm is preemptive, priority-based with round-robin scheduling for processes having the same

prority using time quantum q = 10.

Process Table (stored in a data file)

Process Priority Burst Arrival

------- -------- ----- -------

P1 8 15 0

P2 3 20 0

P3 4 20 20

P4 4 20 25

P5 5 5 45

P6 5 15 55

As in all past projects, you will submit the following:

1) Your source code all in one file and well-documented.

2) Compilation instructions.

3) Output of your program execution on your computer (details further below of exected output)

4) Write-up that includes your program design and issues you encountered and resolved in imlementation

Your program will do the following steps for the following input scenario:

0) The file with process table should be tab-delimitted with data in the order as above in the table.

1) Implement a loop that increments an integer time variable t by 1 from t=0 to t = 96

2) The body of your loop will check for arrival of a process at time t by reading the Process Table

from a file. You read the file and check for any processes at Arrival = t.

3) Inside the loop, if a new process has arrived at time t:

a) Insert the new process into the ready queue (implement the queue as a singly linked list).

b) If no process is in the run state, insert the new process into the run state.

Otherwise, if the new process P\_n has a higher priority than the process P\_r in the run state, preempt

the process in the run state (place it back in the ready queue) and assign the new process

to the run state. Output the following for the context switch: time t, P\_n, P\_r

4) Inside the loop, increment the turnaround time for each process, and, depending on the processes' state,

increment the wait time counter for each process.

5) When you finally terminate the loop:

Output the total turnaround and wait times for each process P\_i: P\_i, turnaround time, wait time