1 Homework Questions

1. **Compute the response time and turnaround time when running three jobs of length 200 with SJF and FIFO schedulers.**

   SJF

```
python scheduler.py -p SJF -l 200,200,200 -c
ARG policy SJF
ARG jlist 200,200,200

Here is the job list, with the run time of each job:
Job 0 ( length = 200.0 )
Job 1 ( length = 200.0 )
Job 2 ( length = 200.0 )

** Solutions **

Execution trace:
[ time   0 ] Run job 0 for 200.00 secs ( DONE at 200.00 )
[ time 200 ] Run job 1 for 200.00 secs ( DONE at 400.00 )
[ time 400 ] Run job 2 for 200.00 secs ( DONE at 600.00 )

Final statistics:
Job   0 -- Response: 0.00  Turnaround 200.00  Wait 0.00
Job   1 -- Response: 200.00  Turnaround 400.00  Wait 200.00
Job   2 -- Response: 400.00  Turnaround 600.00  Wait 400.00

Average -- Response: 200.00  Turnaround 400.00  Wait 200.00
```

   FIFO

```
python scheduler.py -p FIFO -l 200,200,200 -c
ARG policy FIFO
ARG jlist 200,200,200

Here is the job list, with the run time of each job:
Job 0 ( length = 200.0 )
Job 1 ( length = 200.0 )
Job 2 ( length = 200.0 )

** Solutions **

Execution trace:
[ time   0 ] Run job 0 for 200.00 secs ( DONE at 200.00 )
[ time 200 ] Run job 1 for 200.00 secs ( DONE at 400.00 )
```

```
[ time 400 ] Run job 2 for 200.00 secs ( DONE at 600.00 )

Final statistics:
Job   0 -- Response: 0.00  Turnaround 200.00  Wait 0.00
Job   1 -- Response: 200.00  Turnaround 400.00  Wait 200.00
Job   2 -- Response: 400.00  Turnaround 600.00  Wait 400.00

Average -- Response: 200.00  Turnaround 400.00  Wait 200.00
```

2. **Now do the same but with jobs of different lengths: 300, 200, and 100.**

   SJF

```
python scheduler.py -p SJF -l 300,200,100 -c
ARG policy SJF
ARG jlist 300,200,100

Here is the job list, with the run time of each job:
Job 0 ( length = 300.0 )
Job 1 ( length = 200.0 )
Job 2 ( length = 100.0 )


** Solutions **

Execution trace:
[ time   0 ] Run job 2 for 100.00 secs ( DONE at 100.00 )
[ time 100 ] Run job 1 for 200.00 secs ( DONE at 300.00 )
[ time 300 ] Run job 0 for 300.00 secs ( DONE at 600.00 )

Final statistics:
Job   2 -- Response: 0.00  Turnaround 100.00  Wait 0.00
Job   1 -- Response: 100.00  Turnaround 300.00  Wait 100.00
Job   0 -- Response: 300.00  Turnaround 600.00  Wait 300.00

Average -- Response: 133.33  Turnaround 333.33  Wait 133.33
```

   FIFO

```
python scheduler.py -p FIFO -l 300,200,100 -c
ARG policy FIFO
ARG jlist 300,200,100

Here is the job list, with the run time of each job:
```

```
Job 0 ( length = 300.0 )
Job 1 ( length = 200.0 )
Job 2 ( length = 100.0 )


** Solutions **

Execution trace:
[ time   0 ] Run job 0 for 300.00 secs ( DONE at 300.00 )
[ time 300 ] Run job 1 for 200.00 secs ( DONE at 500.00 )
[ time 500 ] Run job 2 for 100.00 secs ( DONE at 600.00 )

Final statistics:
Job   0 -- Response: 0.00  Turnaround 300.00  Wait 0.00
Job   1 -- Response: 300.00  Turnaround 500.00  Wait 300.00
Job   2 -- Response: 500.00  Turnaround 600.00  Wait 500.00

Average -- Response: 266.67  Turnaround 466.67  Wait 266.67
```

3. **Now do the same, but also with the RR scheduler and a time-slice of 1.**

```
python scheduler.py -p RR -q 1 -l 5,10,15 -c
ARG policy RR
ARG jlist 5,10,15

Here is the job list, with the run time of each job:
Job 0 ( length = 5.0 )
Job 1 ( length = 10.0 )
Job 2 ( length = 15.0 )

** Solutions **

Execution trace:
[ time   0 ] Run job   0 for 1.00 secs
[ time   1 ] Run job   1 for 1.00 secs
[ time   2 ] Run job   2 for 1.00 secs
[ time   3 ] Run job   0 for 1.00 secs
[ time   4 ] Run job   1 for 1.00 secs
[ time   5 ] Run job   2 for 1.00 secs
[ time   6 ] Run job   0 for 1.00 secs
[ time   7 ] Run job   1 for 1.00 secs
[ time   8 ] Run job   2 for 1.00 secs
[ time   9 ] Run job   0 for 1.00 secs
```

```
[ time  10 ] Run job   1 for 1.00 secs
[ time  11 ] Run job   2 for 1.00 secs
[ time  12 ] Run job   0 for 1.00 secs ( DONE at 13.00 )
[ time  13 ] Run job   1 for 1.00 secs
[ time  14 ] Run job   2 for 1.00 secs
[ time  15 ] Run job   1 for 1.00 secs
[ time  16 ] Run job   2 for 1.00 secs
[ time  17 ] Run job   1 for 1.00 secs
[ time  18 ] Run job   2 for 1.00 secs
[ time  19 ] Run job   1 for 1.00 secs
[ time  20 ] Run job   2 for 1.00 secs
[ time  21 ] Run job   1 for 1.00 secs
[ time  22 ] Run job   2 for 1.00 secs
[ time  23 ] Run job   1 for 1.00 secs ( DONE at 24.00 )
[ time  24 ] Run job   2 for 1.00 secs
[ time  25 ] Run job   2 for 1.00 secs
[ time  26 ] Run job   2 for 1.00 secs
[ time  27 ] Run job   2 for 1.00 secs
[ time  28 ] Run job   2 for 1.00 secs
[ time  29 ] Run job   2 for 1.00 secs ( DONE at 30.00 )

Final statistics:
Job   0 -- Response: 0.00  Turnaround 13.00  Wait 8.00
Job   1 -- Response: 1.00  Turnaround 24.00  Wait 14.00
Job   2 -- Response: 2.00  Turnaround 30.00  Wait 15.00

Average -- Response: 1.00  Turnaround 22.33  Wait 12.33
```

4. **For what types of workloads does SJF deliver the same turnaround times as FIFO?**
   When the lengths of all jobs are equal. Look at 1).

5. **For what types of worloads and quantum lengths does SJF deliver the same response times as RR?**
   The workloads themselves can be different lengths but as long as the set of them are the same for RR and SJF, you can get the same response time by setting the time-slice (quantum length) to be equal to the length of the longest job. So, if I have 3 processes of lengths 3, 2, and 1 (in that order) the RR time-slice has to be 3.

6. **What happens to the response time with SJF as job lengths increase? Can you use the similator to demonstrate this trend?**
   As job lengths increase, the response time also increases because you have to wait all other processes to finish before actually first running the current process and that just adds up.

7. **What happens to response time with RR as quantum lengths increase? Can you write an equation that gives the worst-case response time, given N jobs?**
   As quantum lenghts increase, RR response times increase. The worst-case response time would be $\frac{q(N-1)}{N}$.