**Objective**

The goal of this assignment is to deepen your understanding of CPU scheduling policies (FIFO, SJF, and RR) by manually computing scheduling metrics and verifying them using a Python-based scheduler simulation. Please compute these times manually and record your work and answers with pen and paper. Please take a photo or scan your hand-written calculations and answers. Submit your photo or scan as a file.

By the end of this assignment, you should be able to:

* Predict execution order based on different scheduling policies.
* Compute **Response Time, Turnaround Time, and Wait Time** for each job.
* Compare the efficiency of different scheduling algorithms.

**Part 1: First-In, First-Out (FIFO) Scheduling**

**Step 1: Generate the Job Set**

Run the following command to generate a **FIFO scheduling scenario** with 4 jobs:

./scheduler.py -p FIFO -j 4 -s 99

This will output a list of 4 jobs with their execution times.

👉 **Write down the job list (job IDs and their lengths).**

**Step 2: Manually Calculate Metrics**

Using FIFO scheduling:

* **Determine the order of execution.**
* **Compute manually:**
  + **Response Time** (When each job first starts)
  + **Turnaround Time** (Time from arrival to completion)
  + **Wait Time** (Time spent in the queue)

👉 **Write down your calculated values before verifying with the program.**  
👉 **DO NOT run with the -c flag until you finish your calculations!**

**Step 3: Verify Your Work**

./scheduler.py -p FIFO -j 4 -s 99 -c

* **Compare your manual calculations with the program’s output.**
* If there are discrepancies, analyze what went wrong and explain why.

👉 Compare your manually calculated values and correct as necessary.

**Part 2: Shortest Job First (SJF) Scheduling**

**Step 1: Generate a New Job Set**

Run the same job set, but use **SJF** instead of FIFO:

./scheduler.py -p SJF -j 4 -s 99

👉 **Write down the job list and predict the execution order (shortest jobs first).**

**Step 2: Compute and Compare**

* Compute **Response Time, Turnaround Time, and Wait Time** manually.
* Predict whether SJF reduces the average wait time compared to FIFO.

👉 **Write down your answers before verifying with:**

./scheduler.py -p SJF -j 4 -s 99 -c

* Compare your manual calculations with the program.
* Reflect: **Was SJF more efficient? Why or why not?**

**Part 3: Round Robin (RR) Scheduling**

**Step 1: Run RR with a Quantum of 3**

Execute:

./scheduler.py -p RR -j 4 -s 99 -q 3

👉 **Predict the execution order before proceeding!**  
👉 **How does RR differ from FIFO and SJF?**

**Step 2: Manually Compute Metrics**

* **Predict the execution order** under Round Robin with **quantum = 3**.
* Compute **Response Time, Turnaround Time, and Wait Time** for each job.

👉 **Write down your predictions before verifying with:**

./scheduler.py -p RR -j 4 -s 99 -q 3 -c

* Compare your work with the program’s results.
* Answer: **How does changing the time quantum affect performance?**