Operating system I
Assignment # 2

CPU Schedulers Simulator

Write a java program to simulate the following schedulers:

1. **preemptive** Shortest- Job First (SJF) Scheduling with context switching
2. Round Robin (RR) with context switching
3. **preemptive** Priority Scheduling (with the solving of starvation problem)
4. AG Scheduling :

    a. Each process is provided a static time to execute called quantum.

    b. Once a process is executed for given time period, it's called **FCFS till** the finishing of (ceil(52%)) of its Quantum time then it's converted to**non preemptive Priority till** the finishing of the next (ceil(52%)), after that it's converted to **preemptive** Shortest- Job First (SJF)**.**

    c. We have 3 scenarios of the running process

        i. The running process used all its quantum time and it still have job to do (add this process to the end of the **queue**, then increases its Quantum time by **Two**).

        ii. The running process was execute as **non-preemptive Priority** and didn't use all its quantum time based on another process converted from ready to running (add this process to the end of the **queue,** and then increase its Quantum time by ceil(**the remaining Quantum time/2**)).

        iii. The running process was execute as **preemptive** Shortest- Job First (SJF) and didn't use all its quantum time based on another process converted from ready to running (add this process to the end of the **queue,** and then increase its Quantum time by **the remaining Quantum time**).

iv. The running process didn't use all of its quantum time because it's no longer need that time and the job was completed (set it's quantum time to **zero**).

Example :

Processes	Burst time	Arrival time	Priority	Quantum
P1	17	0	4	7
P2	6	2	7	9
P3	11	5	3	4
P4	4	15	6	6

Answer:

ξ Quantum (7, 9, 4,6) -> ceil(25%) = (2,-,-,-) && ceil(50%) = (4,-,-,-) ξ

Quantum (7+3,9,4,6) -> ceil(25%) = (-,3,-,-) && ceil(50%) = (-,5,-,-) ξ

Quantum (10,9+3,4 ,6) -> ceil(25%) = (-,-,1,-) && ceil(50%) = (-,-,2,-) ξ

Quantum (10,12,4+2,6) -> ceil(25%) = (-,3,-,-) && ceil(50%) = (-,6,-,-) ξ

Quantum (10,0,6,6) -> ceil(25%) = (3,-,-,-) && ceil(50%) = (5,-,-,-) ξ

Quantum (10+4,0,6,6) -> ceil(25%) = (-,-,2,-) && ceil(50%) = (-,-,3,-) ξ

Quantum (14,0,6+3,6) -> ceil(25%) = (-,-,-,2) && ceil(50%) = (-,-,-,3) ξ

Quantum (14,0,9,6+2) -> ceil(25%) = (-,-,3,-) && ceil(50%) = (-,-,5,-) ξ

Quantum (14,0,0,8) -> ceil(25%) = (4,-,-,-) && ceil(50%) = (7,-,-,-) ξ

Quantum (14+7,0,0,8) -> ceil(52%) = (0,0,0,2) && ceil(50%) = (-,-,-,4) ξ

Quantum (21,0,0,0) -> ceil(25%) = (6,-,-,-) && ceil(50%) = (11,-,-,-)

P1	P2	P3	P2	P1	P3	P4	P3	P1	P4	P1

0 4 7 9 12 15 18 20 26 33 35 38

Program Input

- ③ Number of processes
- ③ Round robin Time Quantum
- ③ Context switching

For Each Process you need to receive the following parameters from the user:

- ③ Process Name
- ③ Process Arrival Time
- ③ Process Burst Time
- ③ Process Priority

Program Output

For each scheduler output the following:

- ③ Processes execution order
- ③ Waiting Time for each process
- ③ Turnaround Time for each process
- ③ Average Waiting Time
- ③ Average Turnaround Time
- ③ Print all history update of quantum time for each process (**AG Scheduling**)

ξ The assignment is submitted in group of max. 5 students and min. 4 students. ξ
Late submission is not allowed

Grading Criteria

BOUNS (15 grades)

	preemptive Shortest-Job First (SJF) Scheduling	Round Robin (RR) Scheduling	Priority Scheduling	AG Scheduling	Grade
Processes execution order	6	6	6	13	**31**

Waiting Time for each process	6	6	6	13	**31**
Turnaround Time for each process	2	2	2	4	**10**
Average Waiting Time	2	2	2	4	**10**
Average Turnaround Time	2	2	2	4	**10**
Print all history update of quantum time for each process (**AG Scheduling**)	0	0	0	8	**8**
Grade	**18**	**18**	**18**	**46**	**100**