

Operating System HW#2

Process Scheduling

各小題均寫在同一檔案中(.cpp)

第一小題： Shortest-Job-First

第二小題： Shortest-Remaining-Time-First

第三小題： Multilevel Feedback Queue:

Round-Robin(第一層) + First-Come, First-Served (第二層)

Input 須是一個.txt檔 (評分用不需繳交)

input.txt檔格式如圖 格式必須與範例一致, 沒有的地方就填0 e.g: 若沒有第二段CPU time就填0而不是空白

```
3
5
1 0 5 6 7
2 3 4 2 3
3 4 2 3 4
4 7 5 2 7
5 14 3 2 4
```

檔案中第一列為欲使用1 2 3小題的哪一題

第二列為有幾個process (3~8之間)

第三列後為processes(每行間隔為一空格)

第一欄數字表示process id(正整數)

第二欄數字表示process arrival time(正整數)

第三欄數字表示process需要花的第一段CPU time(正整數)

第四欄數字表示process需要花的 I/O time(context switch time忽略不計)(正整數)

第五欄數字表示process需要花的第二段CPU time(正整數)

Output 須是一個(學號).txt檔 (同樣不需繳交)

e.g: 0456666.txt

學號.txt 格式如圖

2:17

3:21

5:30

4:37

1:44

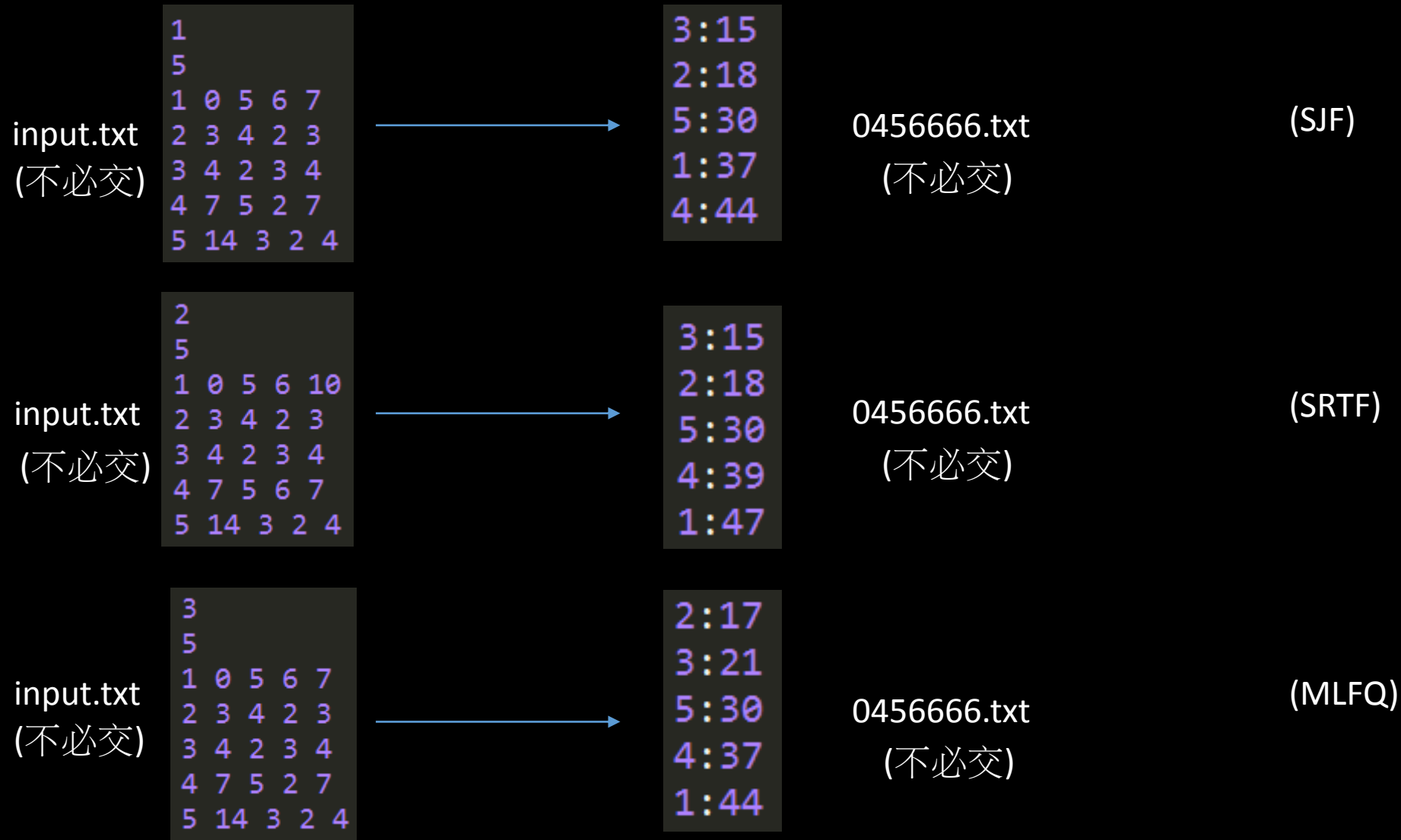
輸出格式為:

(process id):(process finish time)

(中間不必留空格)

輸出先後順序由process finish time決定，先完成的process在愈上方

整體Input即Output結果範例



Question 2-1

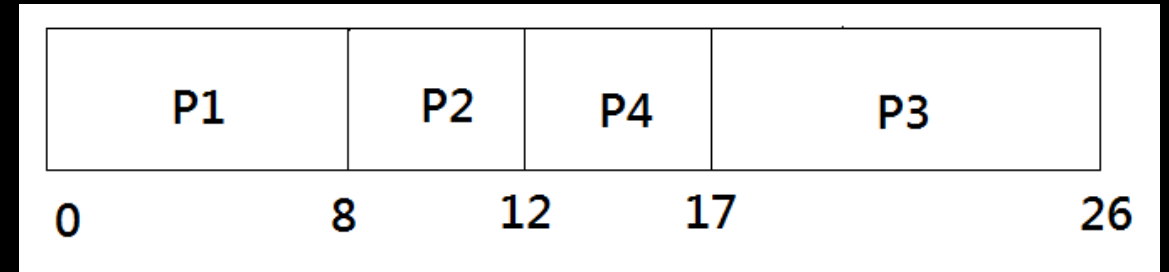
Shortest-Job-First(SJF)

Question 2-2

Shortest-Remaining-Time-First(SRTF)

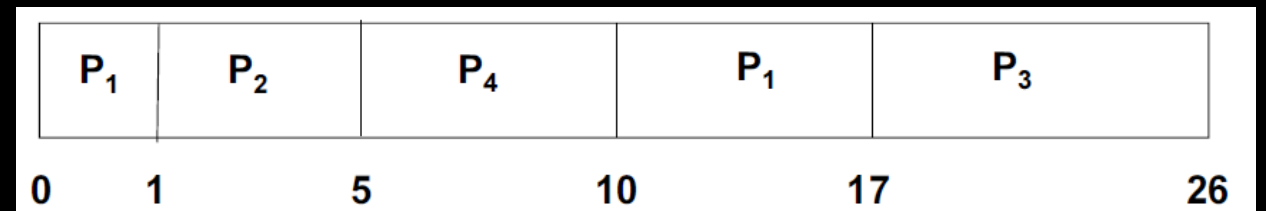
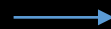
Shortest-Job-First (SJF)

<u>Process</u>	<u>Arrival Time</u>	<u>Burst Time</u>
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5



Shortest-Remaining-Time-First (SRTF)

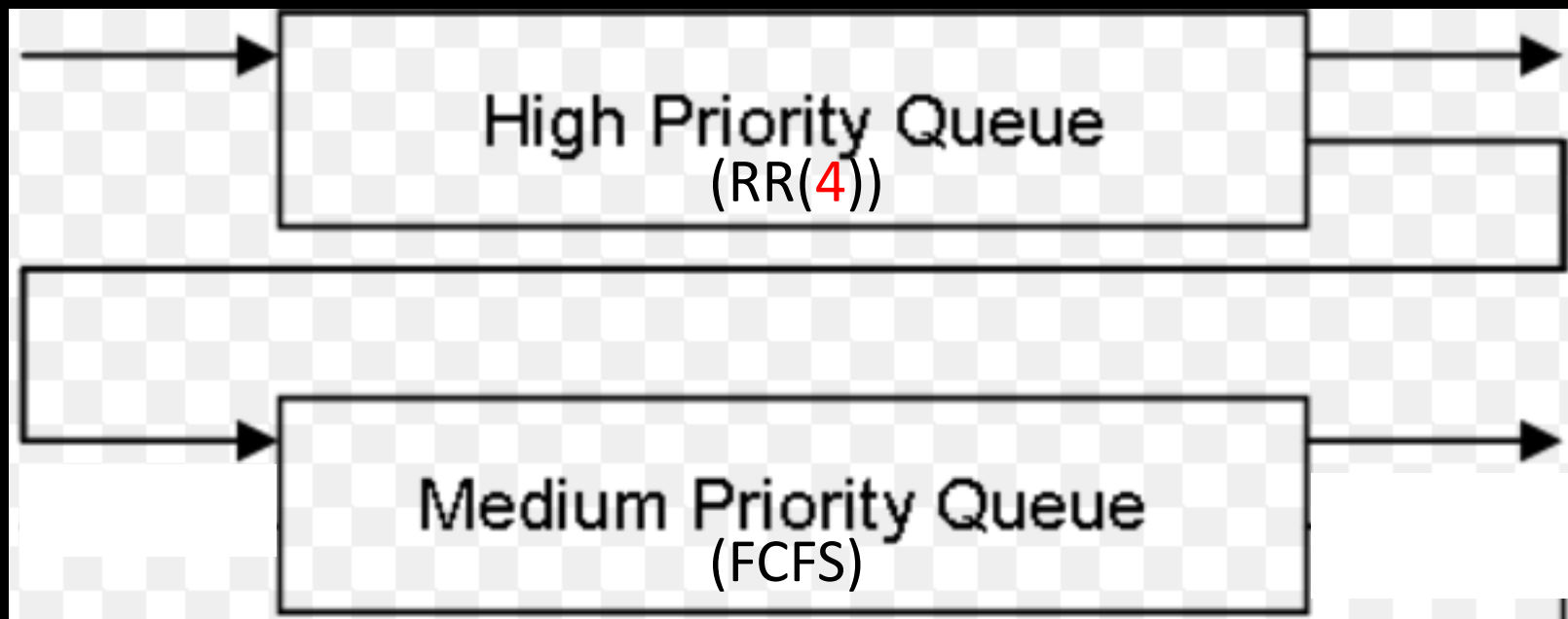
<u>Process</u>	<u>Arrival Time</u>	<u>Burst Time</u>
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5



Question 2-3

Multilevel feed back queue: Round-Robin(RR)+First come, First serve(FCFS)

Input



Output

Output

Processes 一律都先進 High Priority Queue(RR)，超過4個執行時間而未執行完的才進入FCFS Queue

RR Queue中process做完I/O會重新進入RR Queue；FCFS Queue中process做完I/O亦是重新進入FCFS Queue中，若在此同時有新process進入，須排在新進process之後

RR Queue完全空時，才會執行FCFS Queue，不過當再有process進入RR Queue時，正在用CPU 的FCFS process 會立即被preempted，換RR queue 的process 執行

注意事項

注意事項

1. 請繳交一個檔案至e3即可(.cpp) (請用C++)
檔名為"學號_OS_HW2"(ex:1234567_OS_HW2.cpp)
2. 注意input.txt檔案的第一列 1: SJF 2: SRTF 3: MLFQ(RR+FCFS)
input.txt 及 學號.txt 檔都不需繳交自己測試時用即可
3. 繳交期限11/4(五) 23:59:59 遲交0分
4. 抓到抄襲一律0分 輸出格式不符 -10, 檔名不正確 -5, compile error 0分
5. Compile Error的問題請盡量不要問助教，練習debug