1) Calculating the performance of both the two scheduling algorithms :

FCFS - Non Preemptive - (Default Data) :

The Gantt Chart for FCFS :

```
|__P2__|__P5__|__P1__|__P4__|__P3__|
0      20     32     37     44     57
```

a) Waiting Time P2 = 0
Waiting Time P5 = 20 - 10 = 10
Waiting Time P1 = 32 - 16 = 16
Waiting Time P4 = 37 - 18 = 19
Waiting Time P3 = 44 - 20 = 24
Then, Average Waiting Time will be :
= (Waiting Time P2 + Waiting Time P5 + Waiting Time P1 + Waiting Time P4 + Waiting Time P3) / 5
= (0 + 10 + 16 + 19 + 24) / 5
= 69 / 5
= 13.8

b) Response Time P2 = 0
Response Time P5 = 20 - 10 = 10
Response Time P1 = 32 - 16 = 16
Response Time P4 = 37 - 18 = 19
Response Time P3 = 44 - 20 = 24
Then, Average Response Time will be :
= (Response Time P2 + Response Time P5 + Response Time P1 + Response Time P4 + Response Time P3) / 5
= (0 + 10 + 16 + 19 + 24) / 5
= 69 / 5
= 13.8

c) Turn-Around Time P2 = 20 - 0 = 20
Turn-Around Time P5 = 32 - 10 = 22
Turn-Around Time P1 = 37 - 16 = 21
Turn-Around Time P4 = 44 - 18 = 26
Turn-Around Time P3 = 57 - 20 = 37
Then, Average Turn-Around Time will be :
= (Turn-Around Time P2 + Turn-Around Time P5 + Turn-Around Time P1 + Turn-Around Time P4 + Turn-Around Time P3) / 5
= (20 + 22 + 21 + 26 + 37) / 5
= 126 / 5
= 25.2

SJF - Non Preemptive - (Default Data) :

The Gantt Chart for SJF :

```
|__P2__|__P1__|__P4__|__P5__|__P3__|
0      20     25     32     44     57
```

a) Waiting Time P2 = 0

Waiting Time P1 = 20 - 16 = 4
Waiting Time P4 = 25 - 18 = 7
Waiting Time P5 = 32 - 10 = 22
Waiting Time P3 = 44 - 20 = 24
Then, Average Waiting Time will be :
= (Waiting Time P2 + Waiting Time P1 + Waiting Time P4 + Waiting Time P5 + Waiting Time P3) / 5
= (0 + 4 + 7 + 22 + 24) / 5
= 57 / 5
= 11.4

b) Response Time P2 = 0
Response Time P1 = 20 - 16 = 4
Response Time P4 = 25 - 18 = 7
Response Time P5 = 32 - 10 = 22
Response Time P3 = 44 - 20 = 24
Then, Average Response Time will be :
= (Response Time P2 + Response Time P1 + Response Time P4 + Response Time P5 + Response Time P3) / 5
= (0 + 4 + 7 + 22 + 24) / 5
= 57 / 5
= 11.4

c) Turn-Around Time P2 = 20 - 0 = 20
Turn-Around Time P1 = 25 - 16 = 9
Turn-Around Time P4 = 32 - 18 = 14
Turn-Around Time P5 = 44 - 10 = 34
Turn-Around Time P3 = 57 - 20 = 37
Then, Average Turn-Around Time will be :
= (Turn-Around Time P2 + Turn-Around Time P1 + Turn-Around Time P4 + Turn-Around Time P5 + Turn-Around Time P3) / 5
= (20 + 9 + 14 + 34 + 37) / 5
= 114 / 5
= 22.8

2) Comparison between FCFS (First-Come First-Served) and SJF (Shortest Job First) in Non-Preemptive scheduling :

First, we should understand what is non-preemptive scheduling. We can say a scheduling is non-preemptive if once
a process has been give to the CPU, the CPU cannot be taken away from that process. It means once the process
has been taken by the CPU, it cannot be interrupted or stopped by another process until it has finished.
So, some characteristics of non-preemptive scheduling are :
a) Short jobs can probably wait for longer jobs execution but the overall treatment of all processes is fair
b) Response times are more predictable because incoming high priority jobs can not displace waiting jobs
c) There are only two situations the schedular executes jobs :
i) When a process switches from running state to the waiting state
ii) When a process has been terminated

Then, we should about FCFS and SJF. FCFS is the simplest scheduling algorithm. It simply queues processes in the order that they arrive in ready queue. Similar to its name, who comes first to ready queue will be processed first.

In human sense, we can say FCFS is fair, but it is unfair in the sense that long jobs can make the shorter jobs wait or less imporant job is being processed first rather than the important jobs. FCFS scheme is not really useful in scheduling interactive because it cannot guarantee good response time, turn-around time, and waiting time (spend too much time).

Advantage of FCFS :

a) It is easy to understand and easy to be programmed.

b) It is fair, since a single linked list will keep track of all ready process in ready queue.

Disadvantage of FCFS :

a) Throughput can be low, since long process can hog the CPU

b) The average turn-around time, waiting time, and response time is often too long

c) Processing time of each job must be known in advance (only suitable for batch process)

SJF is the scheduler which arranges the processes with the least estimated processing time remaining to be processed next in the queue. In SJF scheduling format, waiting time and response time will increase as the computational requirements increase. Since turn-around time is based on waiting time plus processing time, longer processes are significantly affected by this scheduling format. However, the overall waiting time will always be smaller than FCFS because no process has to wait for the termination of the longest process. This algorithm is designed for maximum throughput in almost every scenario, especially the preemptive type (since in this project, it is non-preemptive).

Advantage of SJF :

a) It can be said as the best scheduling algorithm for shortest jobs

b) Waiting time and turn-around time will always be less compared to FCFS

Disadvantage of SJF :

a) Starvation can be occurred especially in a busy system with many small processes being run. Starvation itself is a condition where long jobs has to wait the execution of shorter jobs. When a shorter job keeps coming to the ready queue, the long jobs can be waited in a very long time.

From the default data, we can see the average waiting time, the average response time, and the average turn-around time of SJF scheduling is always smaller compared to FCFS scheduling. It is because SJF is optimized the execution time of ready queue by choosing the process that have the lowest burst

time to be executed first rather than the longer process. As I already explained above, SJF will arrange the processes with the least burst time first and make the throughput is always better than FCFS. If we only conclude these two scheduling model by looking to the average waiting time, response time, or turn-around time, we can say SJF is always give a better performance rather than FCFS.

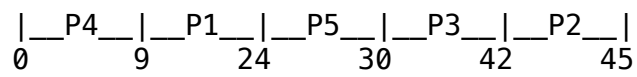
Shortly, it can be concluded :
Average waiting time in SJF < Average waiting time in FCFS ==> Faster, More Efficient
Average response time in SJF < Average response time in FCFS ==> Faster, More Efficient
Average turn-around time in SJF < Average turn-around time in FCFS ==> Faster, More Efficient
Result : SJF is better than FCFS.

However, It is not always a hundred percent correct to say SJF is always better. Imagine if we have a case where there is a process that has the longest burst time, but it is also the most important process that should be executed. If we are using SJF, that longest job will wait in a very long time and probably starvation is occurred there. However, in FCFS, that job can be executed first, or at least faster than in SJF if and only if it has fast arrival time. In this case, we can say FCFS is better in executing these set of processes by considering the most important job is executed faster.

In order to make my statement is stronger, I am using a different test data in this program.

Process	Burst Time	Arrival
P1	15	5
P2	3	20
P3	12	15
P4	9	0
P5	6	10

The Gantt Chart for FCFS :



a) Waiting Time P4 = 0
Waiting Time P1 = 9 - 5 = 4
Waiting Time P5 = 24 - 10 = 14
Waiting Time P3 = 30 - 15 = 15
Waiting Time P2 = 42 - 20 = 22
Then, Average Waiting Time will be :
= (0 + 4 + 14 + 15 + 22) / 5
= 55 / 5
= 11

b) Response Time P4 = 0

Response Time P1 = 9 - 5 = 4
Response Time P5 = 24 - 10 = 14
Response Time P3 = 30 - 15 = 15
Response Time P2 = 42 - 20 = 22
Then, Average Response Time will be :
= (0 + 4 + 14 + 15 + 2) / 5
= 55 / 5
= 11

c) Turn-Around Time P4 = 9 - 0 = 9
Turn-Around Time P1 = 24 - 5 = 19
Turn-Around Time P5 = 30 - 10 = 20
Turn-Around Time P3 = 42 - 15 = 27
Turn-Around Time P2 = 45 - 20 = 25
Then, Average Turn-Around Time will be :
= (9 + 19 + 20 + 27 + 25) / 5
= 100 / 5
= 20

The Gantt Chart for SJF :

```
|__P4__|__P1__|__P2__|__P5__|__P3__|
0      9     24     27     33     45
```

a) Waiting Time P4 = 0
Waiting Time P1 = 9 - 5 = 4
Waiting Time P2 = 24 - 20 = 4
Waiting Time P5 = 27 - 10 = 17
Waiting Time P3 = 33 - 15 = 18
Then, Average Waiting Time will be :
= (0 + 4 + 4 + 17 + 18) / 5
= 43 / 5
= 8.6

b) Response Time P4 = 0
Response Time P1 = 9 - 5 = 4
Response Time P2 = 24 - 20 = 4
Response Time P5 = 27 - 10 = 17
Response Time P3 = 33 - 15 = 18
Then, Average Response Time will be :
= (0 + 4 + 4 + 17 + 18) / 5
= 43 / 5
= 8.6

c) Turn-Around Time P4 = 9 - 0 = 9
Turn-Around Time P1 = 24 - 5 = 19
Turn-Around Time P2 = 27 - 20 = 7
Turn-Around Time P5 = 33 - 10 = 23
Turn-Around Time P3 = 45 - 15 = 30
Then, Average Turn-Around Time will be :
= (9 + 19 + 7 + 23 + 30) / 5
= 88 / 5
= 17.6

We can see, all the average of waiting time, response time, and turn-around time of SJF is always smaller than FCFS.

As I said earlier, it is because SJF is optimized the execution time of ready queue by choosing the process that have the lowest burst time to be executed first rather than the longer process. Again, we can conclude that SJF is better than FCFS in executing this set of processes since it is faster and more efficient. But, from my new example, if we consider that P3 is the most important process that should be executed as soon as possible, we can see FCFS is executing it faster than SJF. In this condition, we can say FCFS is better than SJF.

***