**Operating System Lab**

**CS342**

Tanishq Malu Lab:6 1901CS63

Q1 Consider the n processes, P1, P2.. Pn. Write a program to find out the average

waiting time (WT), turn-around time(TAT) and the completion order of the

processes using **FCFS** scheduling algorithm (in case of conflict, the process with

smaller process id will execute first).

FCFS is a non-preemptive scheduling algorithm.

Compilation:

g++ -o q1 q1.cpp

Syntax:

./q1

Sample Input and Output:

Input:

g++ -o q1 q1.cpp

./q1

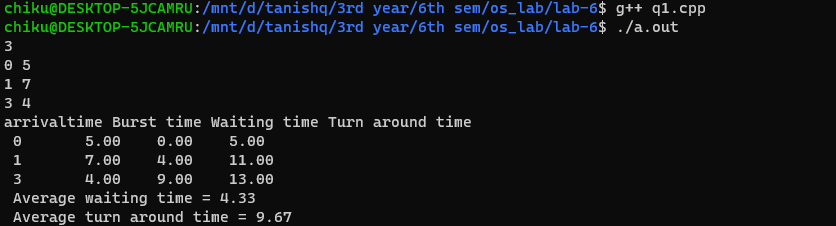
3

0 5

1 7

3 4

Output:



Q 2. Consider the n processes, P1, P2.. Pn. Write a program to find out the average

waiting time (WT), turn-around time (TAT) and the completion order of the processes

using **Round Robin** (RR) scheduling.

Compilation:

g++ -o q2 q2.cpp

Syntax:

./q2

Sample Input and Output:

Input:

g++ -o q2 q2.cpp

./q2

3

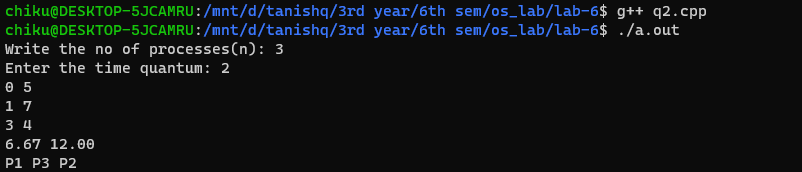
2

0 5

1 7

3 4

Output:



Q3 Consider the n processes, P1, P2.. Pn. Write a program to find out the average waiting time

(WT), turn-around time (TAT) and the completion order of the processes using **Non preemptive**

**Priority** scheduling algorithm (the lower the priority number, the higher is the priority of the

process).

Compilation:

g++ -o q3 q3.cpp

Syntax:

./q3

Sample Input and Output:

Input:

g++ -o q3 q3.cpp

./q3

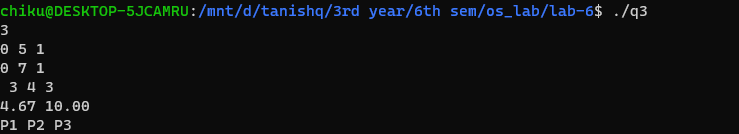
3

0 5 1

0 7 1

3 4 3

Output:



Q4 Consider n processes, P1, P2 .. Pn. Write a program to find out the average

Waiting Time (WT), the Turn-around Time(TAT) and the completion order of the

processes using **Highest remaining time first** (preemptive) scheduling

algorithm (in case of conflict, the process with smaller process id will execute

first).

Compilation:

g++ -o q4 q4.cpp

Syntax:

./q4

Sample Input and Output:

Input:

g++ -o q4 q4.cpp

./q4

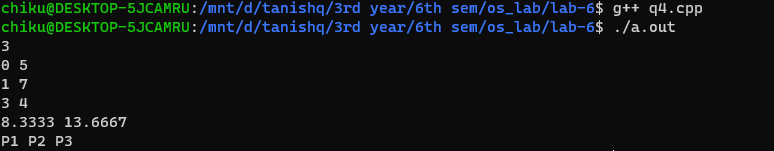
3

0 5

1 7

3 4

Output:



-------------------------------------------------- The End -----------------------------------------------