**Practical - 01**:First Come First Serve(FCFS) Algorithm

**Practical Date**:17­­th July, 2021

**Practical Aim :** Implement FCFS scheduling algorithm using Java

**Algorithm**

Solved Example

Gnatt Chart

Implementation

Input

Output

Sample Output – 01

Sample Output - 02

Sample Output - 03

Sample Output - 04

CPU Scheduling algorithm are used for scheduling different processes present int the ready queue with available resources (CPU cores) in an optimal way so that each and every process get executed by CPU.

Scheduling algorithm are broadly classified into two main types namely Preemptive and Non-preemptive. **First Come First Serve(FCFS)** is an **Non-preemptive Scheduling algorithm** where each process is executed according to its arrival time.

**First Come First Serve (FCFS)** is also known as **First In First out (FIFO)** scheduling algorithm is the easiest and simplest CPU scheduling algorithm where the process which arrives first in the queue is executed first by the CPU. New process is executed only when current the current process is executed fully by the CPU.

**Algorithm**

**Step 1:** Input the number of processes required to be scheduled using FCFS, burst time for each process and its arrival time.

**Step 2:** Using enhanced bubble sort technique sort all the given processes in the ascending order according to arrival time in a ready queue

**Example of First Come First Serve Algorithm**

Consider the following example containing five process arrive at same time

|  |  |
| --- | --- |
| **Process Id** | **Burst Time** |
| P0 | 6 |
| P1 | 3 |
| P2 | 8 |
| P3 | 3 |
| P4 | 4 |

**Step 1:** Processes get executed according to their arrival.

**Step 2:** Following shows the scheduling and execution of the processes.

**Step 2.1:** At start P0 arrives and gets executed for 6(i.e., 0-6) seconds

|  |  |
| --- | --- |
| System time | 0 |
| Process Scheduled | P0 |
| Turn around time | 6-0 |
| Waiting Time | 6-6=0 |

**Step 2.2:** P1 arrive after completion of p0, p1 is execute for 3.

|  |  |
| --- | --- |
| System time | 6 |
| Process Scheduled | P0, P1 |
| Turn around time | 9-0=9 |
| Waiting Time | 9-3=6 |

**Step 2.3:** P2 arrive after completion of process P1 for 8.

|  |  |
| --- | --- |
| System time | 9 |
| Process Scheduled | P0, P1, P2 |
| Turn around time | 17-0=17 |
| Waiting Time | 17-8=9 |

**Step 2.4:** P3 arrive and get execute for 3

|  |  |
| --- | --- |
| System time | 17 |
| Process Scheduled | P0, P1, P2, P3 |
| Turn around time | 20-0=20 |
| Waiting Time | 20-3=17 |

|  |  |
| --- | --- |
| System time | 20 |
| Process Scheduled | P0, P1, P2, P3, P4 |
| Turn around time | 24-0=24 |
| Waiting Time | 24-4=20 |

**Step 2.5:** P4 arrives gets execute for 4.

**Step 3:** Calculate average waiting time and average turn around time.

|  |
| --- |
| Average waiting time = (0+6+9+17+20)/5  = 52/5  = 10.4  Average turn around time = (6+9+17+20+24)/5  = 76/5  = 15.2 |

**Gnatt Chart.**

**Step 4:** After scheduling of all provided processes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process id** | **Burst time** | **Arrival time** | **Finish time** | **Turn Around Time** | **Waiting time** |
| P0 | 6 | 0 | 0 + 6 = 6 | 6 – 0 = 6 | 6 - 6=0 |
| P1 | 3 | 0 | 6 + 3 = 9 | 9 – 0 = 9 | 9 - 3=6 |
| P2 | 8 | 0 | 9 + 8 = 17 | 17 – 0 = 17 | 17 - 8=9 |
| P3 | 3 | 0 | 17 + 3 = 20 | 20 – 0 = 20 | 20 - 3=17 |
| P4 | 4 | 0 | 20 + 4 = 24 | 24 – 0 = 24 | 24 - 4=20 |
| Average |  |  |  | 15.20 | 10.40 |

Consider the following example containing five process arrive at same time

|  |  |  |
| --- | --- | --- |
| **Process Id** | **Burst time** | **Arrival time** |
| P0 | 6 | 2 |
| P1 | 3 | 5 |
| P2 | 8 | 1 |
| P3 | 3 | 0 |
| P4 | 4 | 4 |

**Step 1:** Processes get executed according to their arrival.

**Step 2:** Following shows the scheduling and execution of the processes.

**Step 2.1:** At start p3 arrive and get execute for 0-3 second.

|  |  |
| --- | --- |
| System time | 0 |
| Process Scheduled | P3 |
| Turn around time | 3-0=3 |
| Waiting Time | 3-3=0 |

**Step 2.2:** P0 arrives at time 4 sec but gets resource of CPU at 17 second for execution its execution period is 17-21 second.

|  |  |
| --- | --- |
| System time | 11 |
| Process Scheduled | P3, P2, P0 |
| Turn around time | 17-2=15 |
| Waiting Time | 15-2=13 |

**Step 2.3**: P4 arrives at time 4 sec but gets resource of CPU at 17 second for execution period is 17-21 second.

|  |  |
| --- | --- |
| System time | 11 |
| Process Scheduled | P3, P2, P0 |
| Turn around time | 15-6=9 |
| Waiting time | 17-2=15 |

**Step 2.4:** p4 arrives at time 4 sec but gets resource of cpu at 17 second for execution period is 17-21 second

|  |  |
| --- | --- |
| System time | 17 |
| Process Scheduled | P3,P2, P0, P4 |
| Turn around time | 17-4=13 |
| Waiting Time | 21-4=17 |

**Step 2.5:** : similarly p1 arrives at time 5 sec but its execution gets started turn around time 21 second and last for a period 21-24 second

|  |  |
| --- | --- |
| System time | 21 |
| Process Scheduled | P3, P2, P0, P4, P1 |
| Turn around time | 19-3=16 |
| Waiting Time | 24-5=19 |

**Step 3:** Calculate average waiting time and average turn around time.

|  |
| --- |
| Average waiting time = (0+2+9+13+16)/5  = 40/5  = 8  Average turn around time = (3+10+15+17+19)/5  = 64/5  = 12.8 |

**Gnatt Chart.**

**Step 4:** After scheduling of all provided processes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process id** | **Burst time** | **Arrival time** | **Finish time** | **Turn Around Time** | **Waiting time** |
| P3 | 3 | 0 | (-+3=)3 | 3 - 0 = 3 | 3-3=0 |
| P2 | 8 | 1 | (3+8 =) 9 | 11 – 1 = 10 | 10-8=2 |
| P0 | 6 | 2 | (11+ 6 =) 17 | 17 – 2= 15 | 15 – 6= 9 |
| P4 | 4 | 4 | (17 + 4 =)21 | 21 – 4 = 17 | 17- 4 = 13 |
| P1 | 3 | 5 | (21 + 3 =) 24 | 24 – 5 = 19 | 19 - 3= 16 |
| Average |  |  |  | 12.800 | 8.000 |

**Step 5:** Stop

|  |  |
| --- | --- |
| **Process Id** | **Arrival time** |
| P0 | 2 |
| P1 | 1 |
| P2 | 6 |

Consider the following example containing five processes arrive at the Same time.

**Step 1:** Processes get executed according to their arrival.

**Step 2:** Following shows the scheduling and execution of the processes.

**Step 2.1:** At start P0 arrives and gets executed for 2(i.e., 0-2) seconds

|  |  |
| --- | --- |
| System time | 0 |
| Process Schedule | P0 |
| Turn around time | 2-0=2 |
| Waiting time | 2-2=0 |

Step 2.2: p1 arrive after completion of P0, P1 is execute for 1

|  |  |
| --- | --- |
| System time | 2 |
| Process Schedule | P0,P1 |
| Turn around time | 3-0=2 |
| Waiting time | 3-1=2 |

Step2.3: P2 arrive after complete execution of process P1 for 6.

|  |  |
| --- | --- |
| System time | 3 |
| Process Schedule | P0, P1, P2 |
| Turn around time | 9-0=9 |
| Waiting time | 9-6=3 |

**Step 3:** Calculate average waiting time and average turn around time.

|  |
| --- |
| Average waiting time = (0+2+3)/3  = 5/3  = 1.6666  Average turn around time = (2 + 3+9)/3  = 14/3  = 4.6666 |

**Gnatt chart**

Step 4: After scheduling of all provided processes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process id** | **Burst time** | **Arrival time** | **Finish time** | **Turn Around Time** | **Waiting time** |
| P0 | 2 | 0 | (0+2=)2 | 3- 1 = 2 | 2-2=0 |
| P1 | 1 | 0 | (2+1 =) 3 | 6- 3 = 3 | 3-1=2 |
| P2 | 6 | 0 | (3+ 6 =) 9 | 9- 0 =0 | 9 – 6 = 3 |
| Average |  |  |  | 4.6666 | 1.6666 |

Example 4: Consider the following example containing five processes with varied arrival time.

|  |  |  |
| --- | --- | --- |
| Process Id | Burst time | Arrival time |
| P0 | 4 | 3 |
| P1 | 3 | 5 |
| P2 | 2 | 0 |
| P3 | 1 | 5 |
| P4 | 3 | 4 |

**Step 3:** Calcluate the average waiting time and average turn around time.

|  |
| --- |
| Average wating time = (3+1+7+4+6)/5  =21/5  =4.2  Average turn around time = (1+2+9+5+9)/5  =26/5  =5.2 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process id** | **Burst time** | **Arrival time** | **Finish time** | **Turn Around Time** | **Waiting time** |
| P0 | 4 | 3 | 6 | 3 | 1 |
| P1 | 3 | 5 | 12 | 7 | 4 |
| P2 | 2 | 0 | 2 | 2 | 0 |
| P3 | 1 | 5 | 13 | 8 | 7 |
| P4 | 3 | 4 | 9 | 4 | 9 |
| Average |  |  |  | 5.0000 | 2.4 |

**Implementation:**

import java.util.Scanner;

public class P1\_FCFS\_AJ

{

int burstTime[];

int arrivalTime[];

String[] processId;

int numberOfProcess;

void getProcessData(Scanner input){

System.out.print("Enter the number of process for schedule:");

int inputNumberOfProcess=input.nextInt();

numberOfProcess = inputNumberOfProcess;

burstTime = new int[numberOfProcess];

arrivalTime = new int[numberOfProcess];

processId = new String[numberOfProcess];

String st = "P";

for (int i = 0;i<numberOfProcess;i++){

processId[i]=st.concat(Integer.toString(i));

System.out.print("Enter the burst time for process -"+(i)+":");

burstTime[i] = input.nextInt();

System.out.print("Enter the arrival time for the proccess:");

arrivalTime[i] = input.nextInt();

}

}

void sortAccordingArrivalTime(int[] at,int[] bt,String[] pid){

boolean swapped;

int temp;

String stemp;

for (int i = 0;i< numberOfProcess;i++){

swapped=false;

for(int j = 0;j<numberOfProcess-i-1;j++){

if (at[j]>at[j+1]){

temp = at[j];

at[j]=at[j+1];

at[j+1]=temp;

temp =bt[j];

bt[j] =bt[j+1];

bt[j+1] = temp;

stemp = pid[j];

pid[j] = pid [j+1];

pid[j +1]=stemp;

swapped = true;

}

}

if (swapped == false){

break;

}

}

}

void firstComeFirstServeAlgorithm(){

int finishTime[] = new int[numberOfProcess];

int bt[] = burstTime.clone();

int at[] = arrivalTime.clone();

String pid[] = processId.clone();

int waitingTime[] = new int[numberOfProcess];

int turnAroundTime[] = new int[numberOfProcess];

sortAccordingArrivalTime(at,bt,pid);

finishTime[0] =at[0] =bt[0];

turnAroundTime[0]=finishTime[0]- at[0];

waitingTime[0]=turnAroundTime[0]-bt[0];

for (int i=1;i<numberOfProcess;i++){

finishTime[i]=bt[i]+finishTime[i-1];

turnAroundTime[i]=finishTime[i]-at[i];

waitingTime[i]=turnAroundTime[i]-bt[i];

}

float sum = 0;

for (int n:waitingTime){

sum+=n;

}

float averageWaitingTime=sum/numberOfProcess;

sum=0;

for(int n:turnAroundTime){

sum +=n;

}

float averageTurnAroundTime = sum/numberOfProcess;

System.out.println("FCFS Scheduling Algoritm:");

System.out.format("%20s%20s%20s%20s%20s%20s\n", "ProcessId","BurstTime","ArrivalTime","FinishTime","TurnAroundTime","WaitingTime");

for (int i =0;i<numberOfProcess;i++){

System.out.format("%20s%20s%20s%20s%20s%20s\n", pid[i],bt[i],at[i],finishTime[i],turnAroundTime[i],waitingTime[i]);

}

}

public static void main(String[] args){

Scanner input = new Scanner(System.in);

P1\_FCFS\_AJ obj = new P1\_FCFS\_AJ();

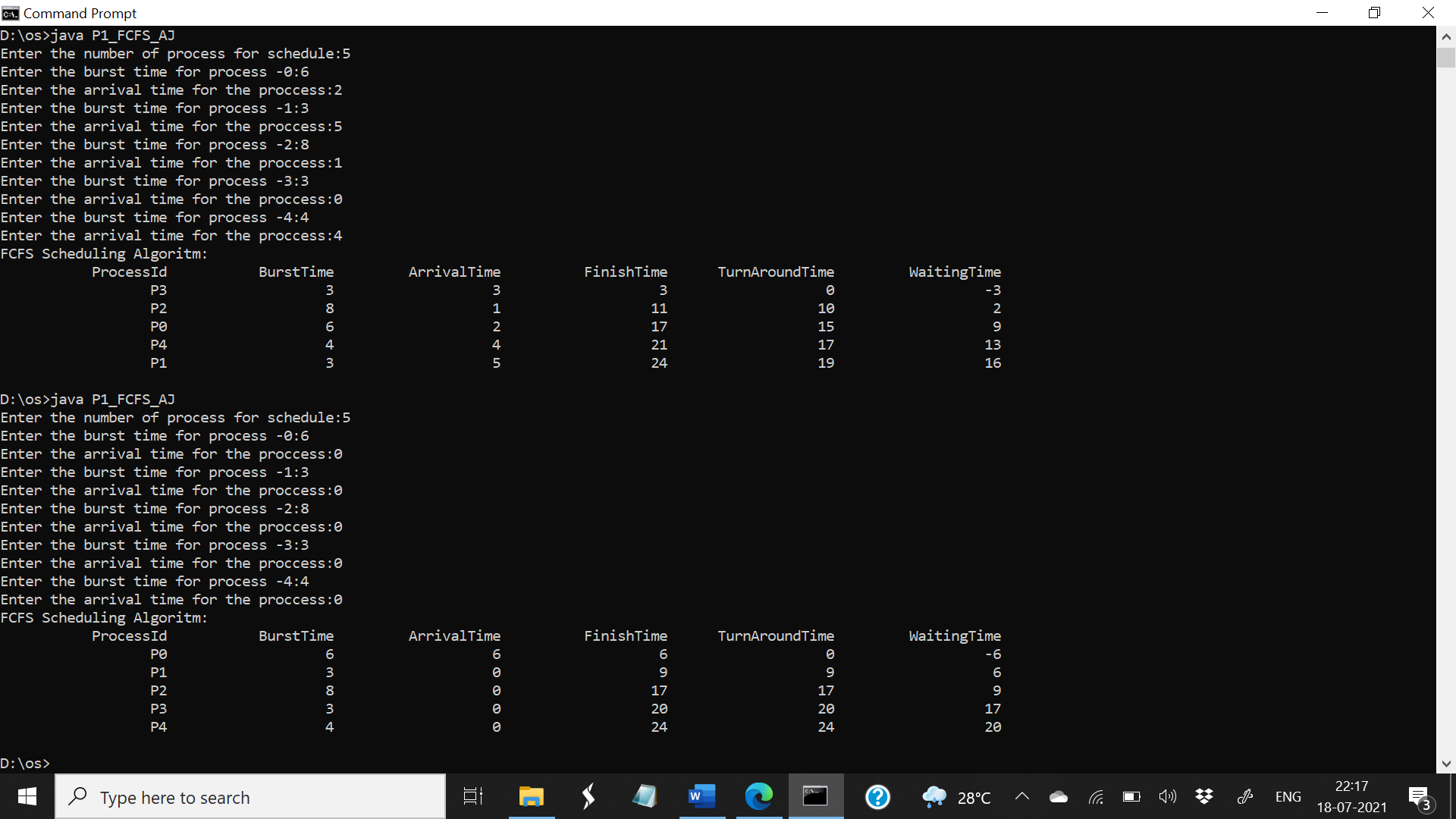
obj.getProcessData(input);

obj.firstComeFirstServeAlgorithm();

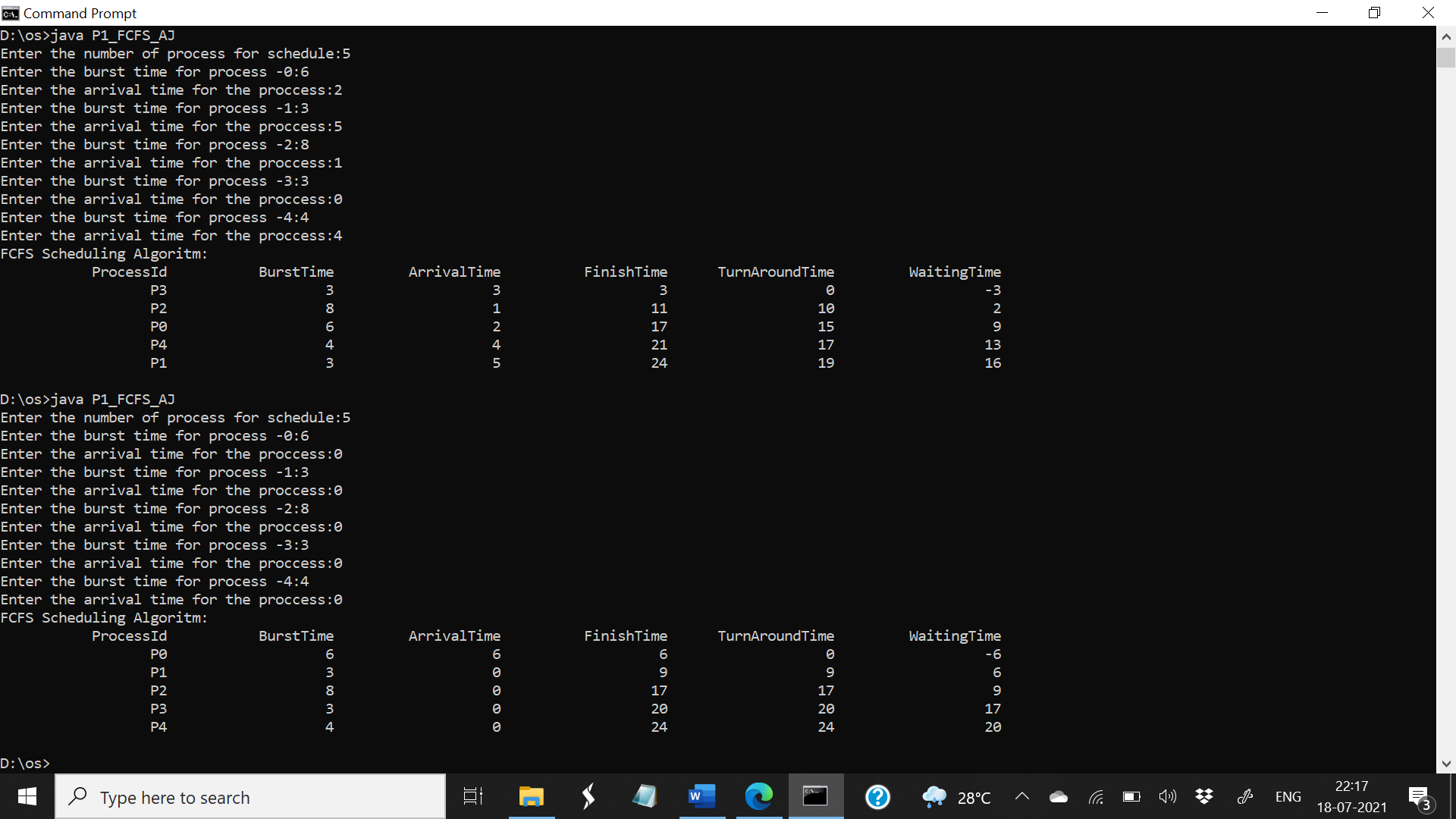
}

}

**Input:**

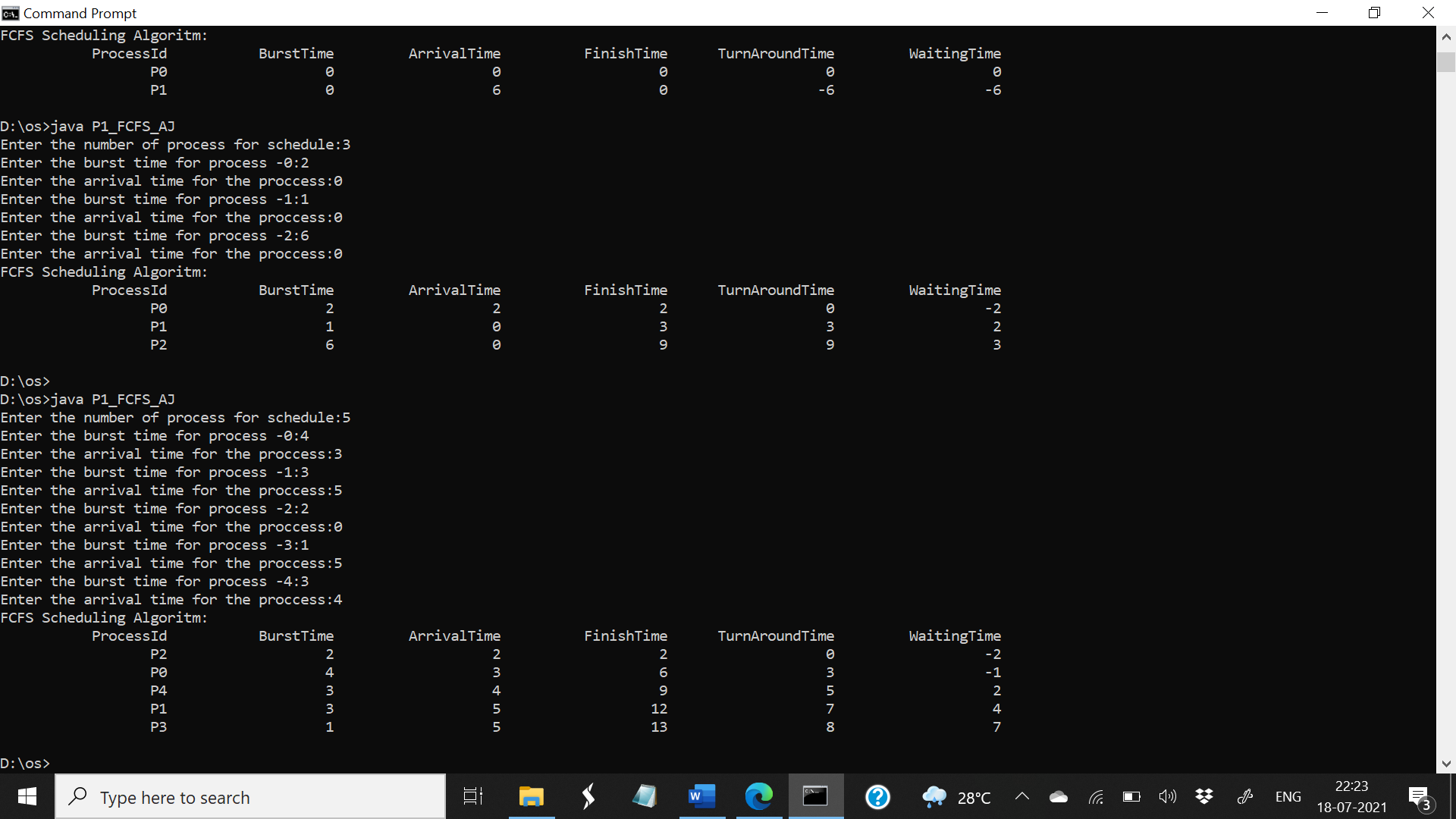


**Output:**

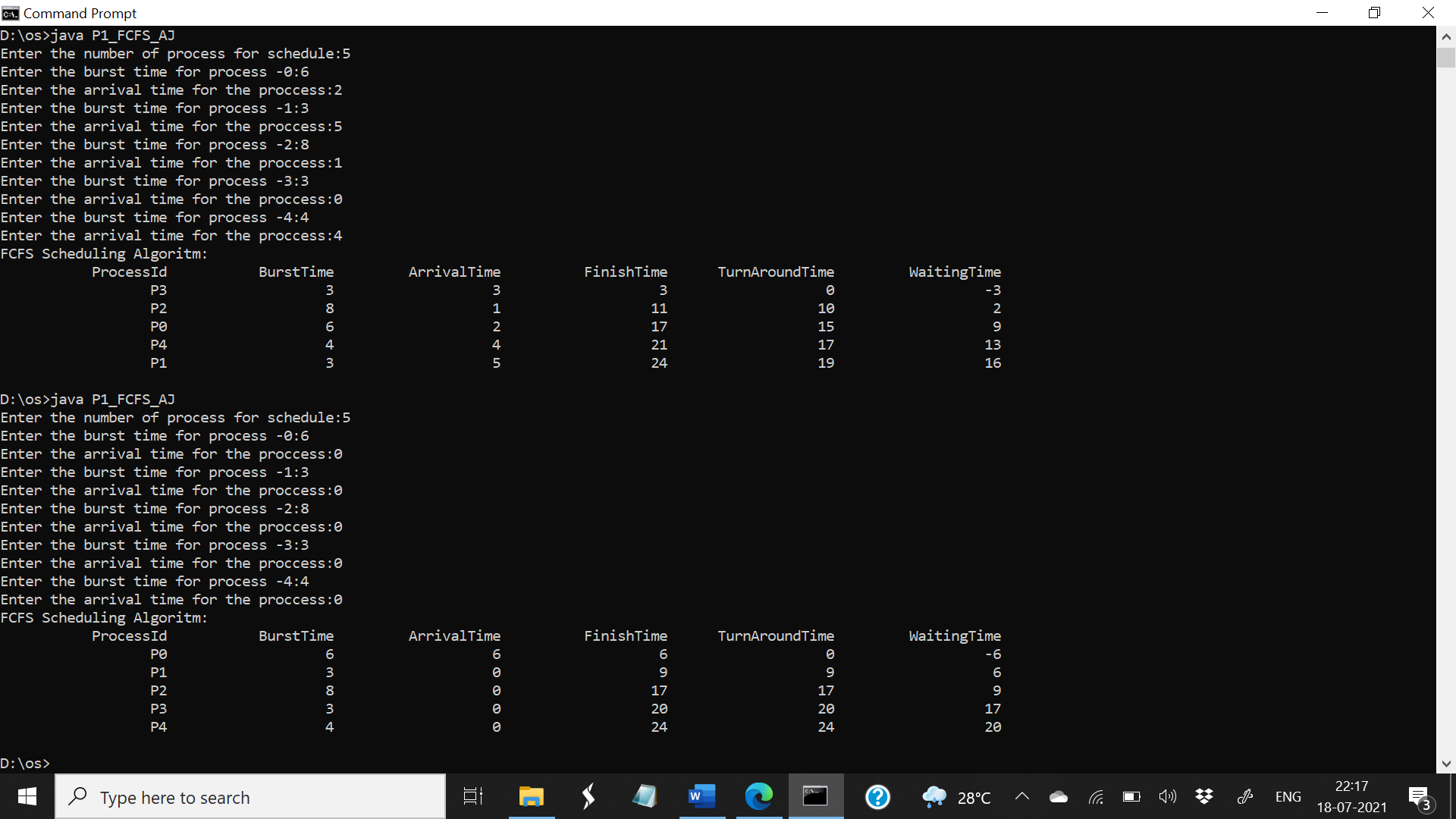
****

**Sample output 01:**

**Input:**

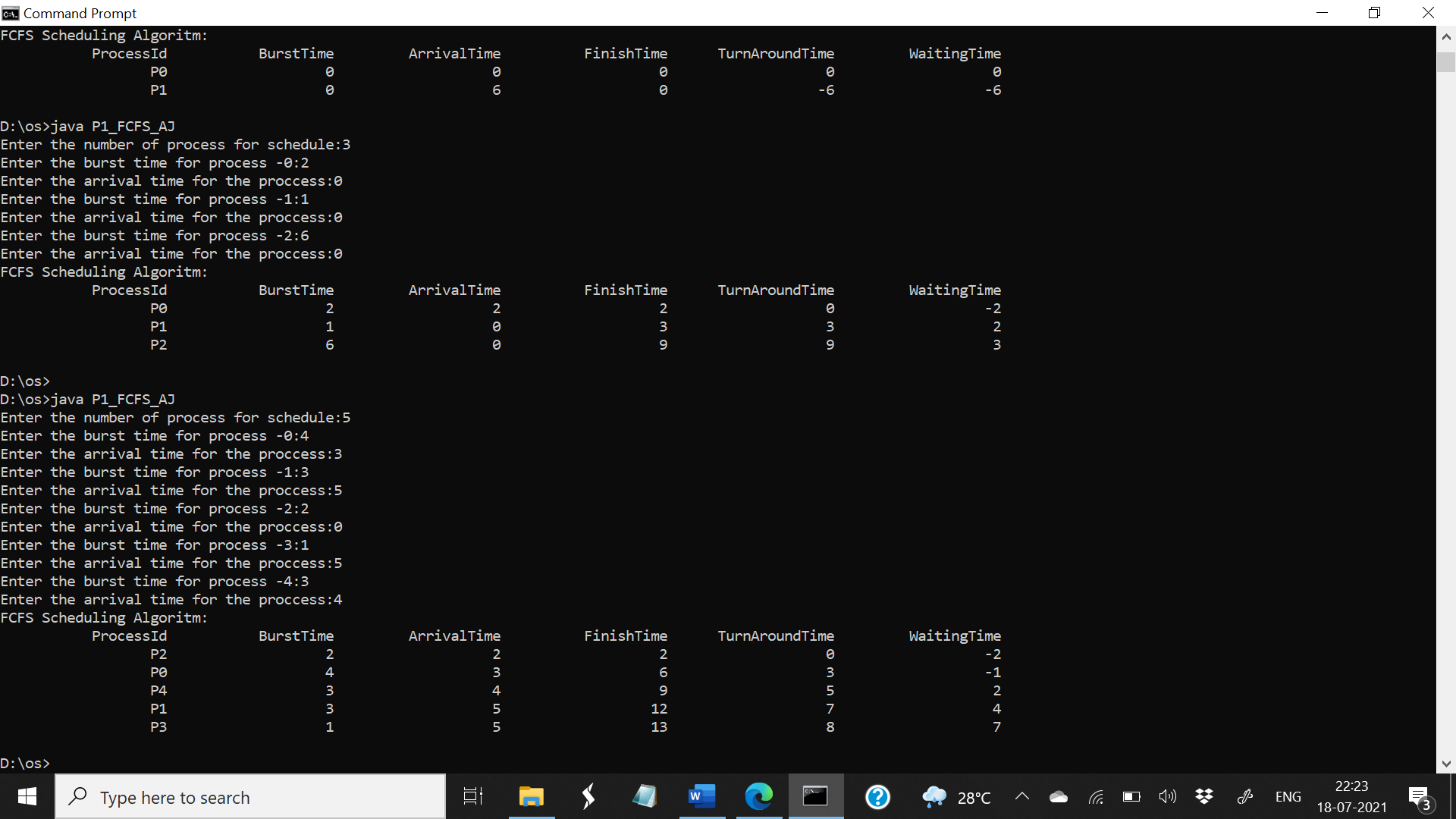


**Output:**

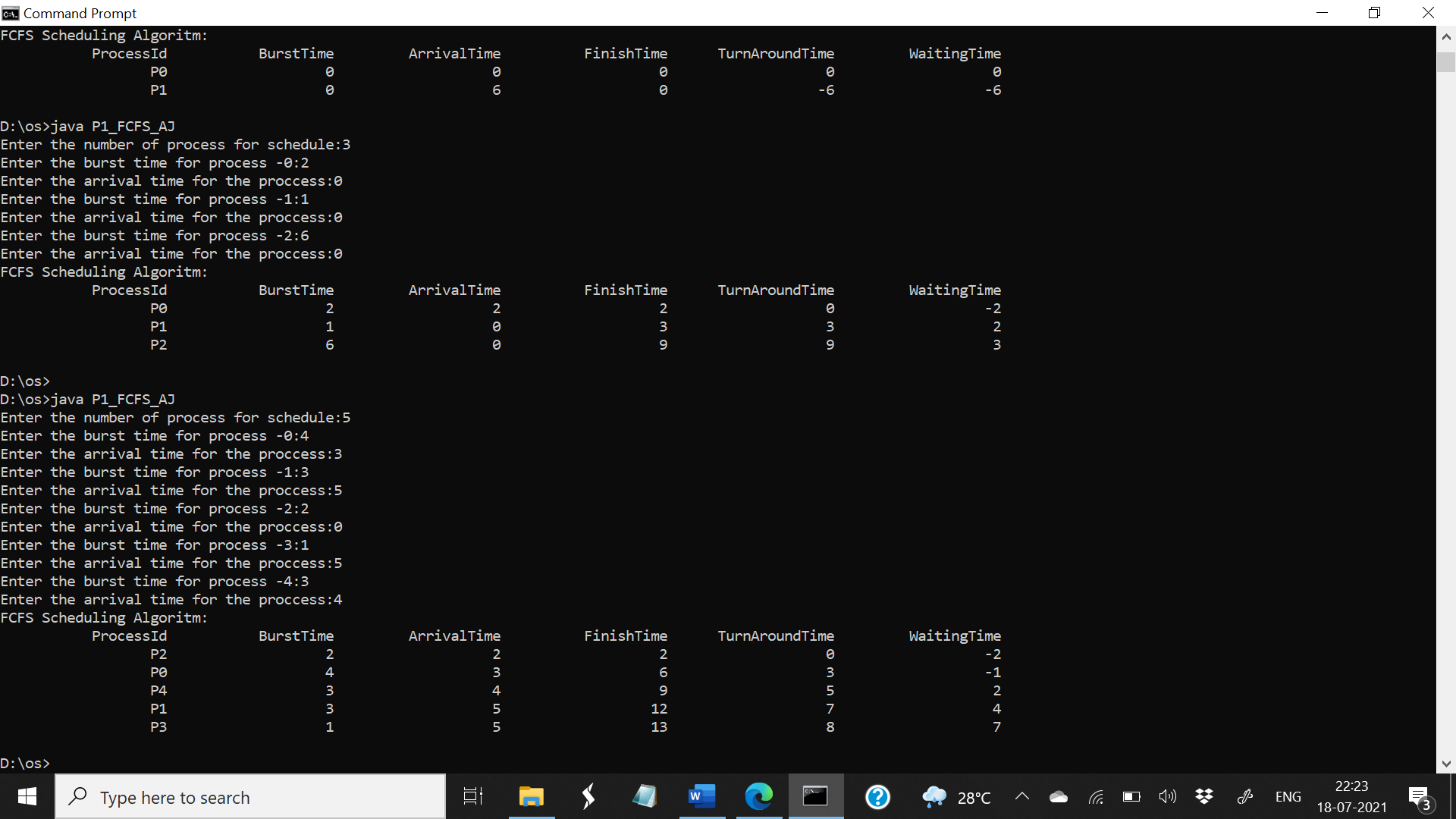


**Sample 2**

**Input:**

****

**Output:**

****