EXPERIMENT NO : 4

**Problem Statement:** Write a program to simulate CPU Scheduling Algorithms:

1. FCFS
2. SJF(Preemptive)
3. Priority(Non- Preemptive)
4. Round Robin(Preemptive)
5. FCFS

# Source Code:

**package** lab;

**import** java.io.\*;

**import** java.util.Scanner;

**public class** FCFS

{

**public static void** main(String args[])

{

**int** i,no\_p,burst\_time[],TT[],WT[]; **float** avg\_wait=0,avg\_TT=0; burst\_time=**new int**[50];

TT=**new int**[50]; WT=**new int**[50];

WT[0]=0;

Scanner s=**new** Scanner(System.***in***); System.***out***.println("Enter the number of process: "); no\_p=s.nextInt();

System.***out***.println("\nEnter Burst Time for processes:");

**for**(i=0;i<no\_p;i++)

{

System.***out***.print("\tP"+(i+1)+": "); burst\_time[i]=s.nextInt();

}

**for**(i=1;i<no\_p;i++)

{

WT[i]=WT[i-1]+burst\_time[i-1]; avg\_wait+=WT[i];

}

avg\_wait/=no\_p;

**for**(i=0;i<no\_p;i++)

{

TT[i]=WT[i]+burst\_time[i]; avg\_TT+=TT[i];

}

avg\_TT/=no\_p;

System.***out***.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*"); System.***out***.println("\tProcesses:");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println(" Process\tBurst Time\tWaiting Time\tTurn Around Time");

**for**(i=0;i<no\_p;i++)

{

System.***out***.println("\tP"+(i+1)+"\t "+burst\_time[i]+"\t\t "+WT[i]+"\t\t "+TT[i]);

}

System.***out***.println("\n

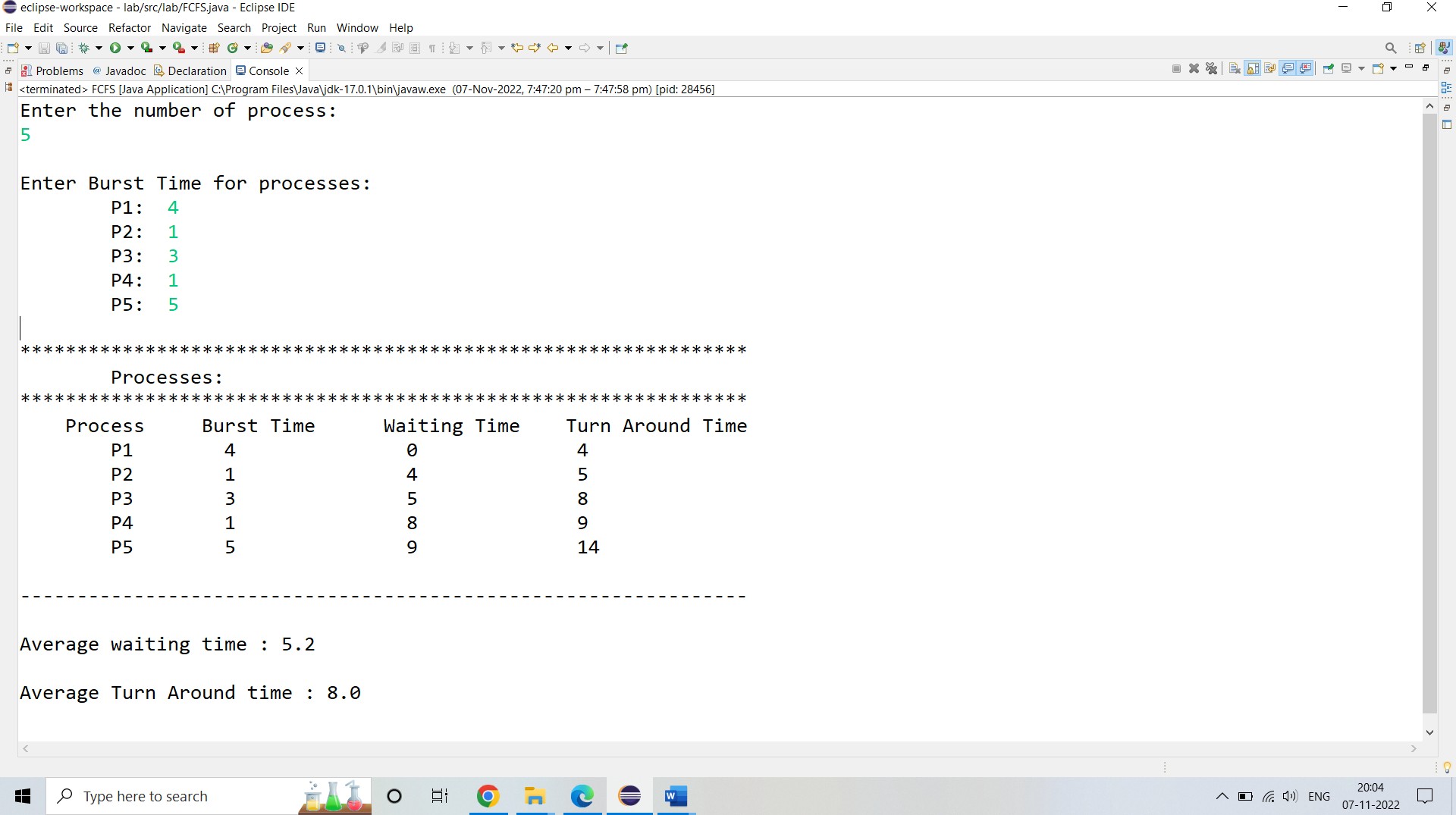
");

System.***out***.println("\nAverage waiting time : "+avg\_wait); System.***out***.println("\nAverage Turn Around time : "+avg\_TT+"\n");

}

}

**OUTPUT:**



1. SJF(Preemptive)

**Source Code:**

**package** lab;

**import** java.util.Scanner;

**class** sjf{

**public static void** main(String args[])

{

**int**\_time[],i,j,n,total=0,total\_comp=0,pos,temp; **float** wait\_avg,TAT\_avg;

Scanner s = **new** Scanner(System.***in***); System.***out***.print("Enter number of process: "); n = s.nextInt();

process = **new int**[n]; burst\_time = **new int**[n]; waiting\_time = **new int**[n]; arr\_time=**new int**[n];

tat = **new int**[n]; completion\_time=**new int**[n];

//burst time

System.***out***.println("\nEnter Burst time:");

**for**(i=0;i<n;i++)

{

System.***out***.print("\nProcess["+(i+1)+"]: "); burst\_time[i] = s.nextInt();;

process[i]=i+1; //Process Number

}

//arrival time

System.***out***.println("\nEnter arrival time:");

**for**(i=0;i<n;i++)

{

System.***out***.print("\nProcess["+(i+1)+"]: "); arr\_time[i] = s.nextInt();;

process[i]=i+1; //Process Number

}

//Sorting

**for**(i=0;i<n;i++)

{

pos=i;

**for**(j=i+1;j<n;j++)

{

**if**(burst\_time[j]<burst\_time[pos]) pos=j;

}

temp=burst\_time[i]; burst\_time[i]=burst\_time[pos]; burst\_time[pos]=temp;

temp=process[i]; process[i]=process[pos]; process[pos]=temp;

System.***out***.println("process"+process[i]);

}

//completion time new

**for**(i=1;i<n;i++)

{

completion\_time[i]=0;

**for**(j=0;j<i;j++)

completion\_time[i]+=burst\_time[j];

total\_comp+=completion\_time[i];

}

//First process has 0 waiting time waiting\_time[0]=0;

//calculate

**for**(i=1;i<n;i++)

{

waiting\_time[i]=0; **for**(j=0;j<i;j++) waiting\_time[i]+=burst\_time[j]; total+=waiting\_time[i];

}

//Calculating Average waiting time wait\_avg=(**float**)total/n;

total=0;

System.***out***.println("\nPro\_number\t Burst Time \tcompletion\_time\tWaiting Time\tTurnaround Time");

**for**(i=0;i<n;i++)

{

tat[i]=burst\_time[i]+waiting\_time[i];

//Calculating Turnaround Time total+=tat[i];

System.***out***.println("\n"+process[i]+"\t\t "+burst\_time[i]+"\t\t "+completion\_time[i]+"\t\t"+waiting\_time[i]+"\t\t "+tat[i]);

}

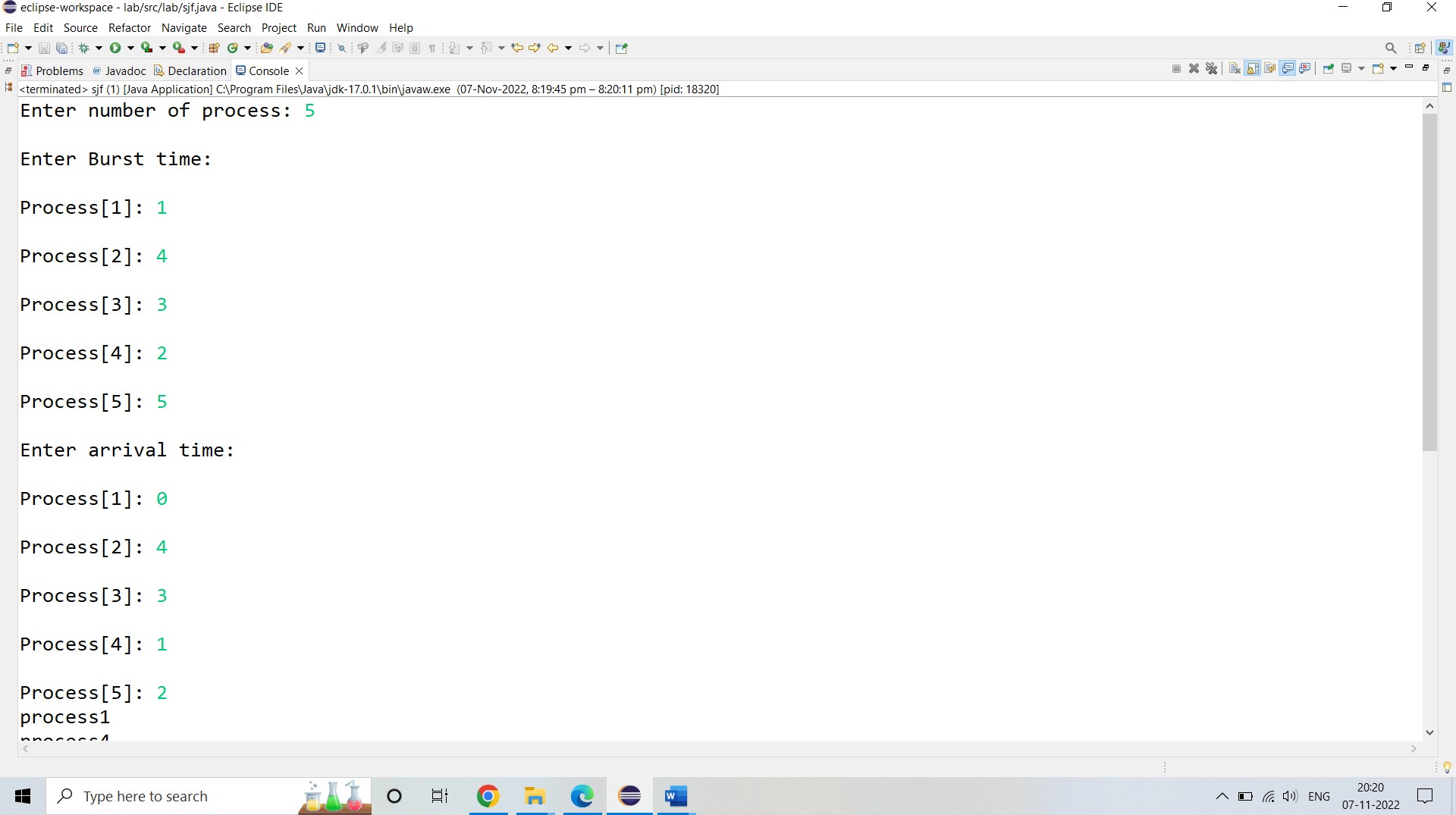
//Calculation of Average Turnaround Time TAT\_avg=(**float**)total/n;

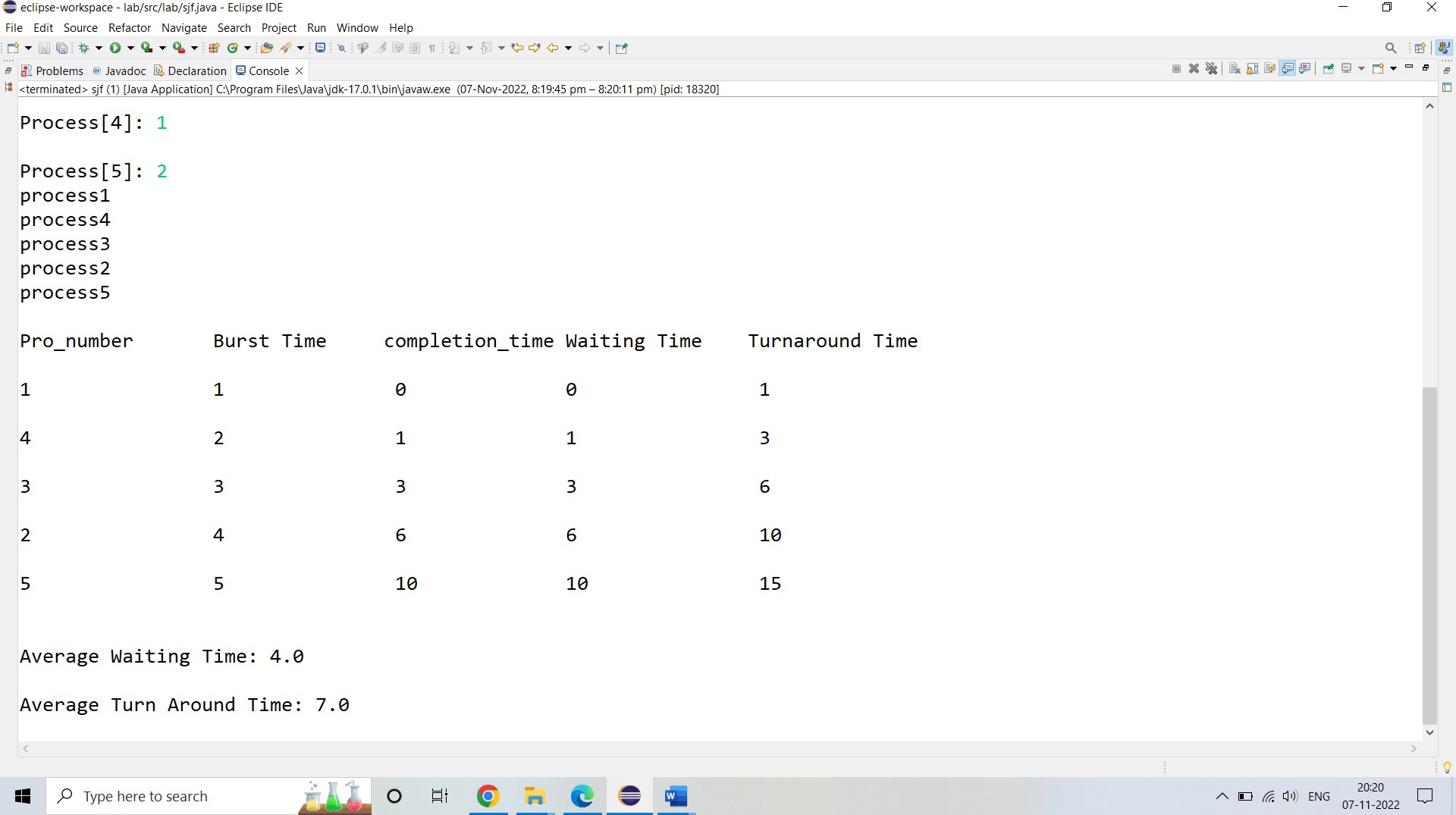
System.***out***.println("\n\nAverage Waiting Time: "+wait\_avg); System.***out***.println("\nAverage Turn Around Time: "+TAT\_avg);

}

}

## OUTPUT:





1. **Priority (Preemtive) Source Code:**

**package** lab;

**import** java.util.Scanner;

**package** lab;

**import** java.util.Scanner;

**public class** Priority

{

**int** burstTime[];

**int** priority[];

**int** arrivalTime[]; String[] processId; **int** numberOfProcess;

**void** getProcessData(Scanner input)

{

System.***out***.print("Enter the number of Process for Scheduling : ");

**int** inputNumberOfProcess = input.nextInt(); numberOfProcess = inputNumberOfProcess; burstTime = **new int**[numberOfProcess]; priority = **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 Process - " + (i) + " : "); arrivalTime[i] = input.nextInt();

System.***out***.print("Enter the priority for Process - " + (i) + " : "); priority[i] = input.nextInt();

}

}

**void** sortAccordingArrivalTimeAndPriority(**int**[] at, **int**[] bt, **int**[] prt, String[] pid)

{

**int** temp; String stemp;

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

{

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

{

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

{

//swapping arrival time temp = at[j];

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

//swapping burst time temp = bt[j];

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

//swapping priority temp = prt[j]; prt[j] = prt[j + 1]; prt[j + 1] = temp;

//swapping process identity stemp = pid[j];

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

}

//sorting according to priority when arrival timings are same

**if** (at[j] == at[j + 1])

{

**if** (prt[j] > prt[j + 1])

{

//swapping arrival time temp = at[j];

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

//swapping burst time temp = bt[j];

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

//swapping priority temp = prt[j]; prt[j] = prt[j + 1]; prt[j + 1] = temp;

//swapping process identity stemp = pid[j];

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

}

}

}

}

}

**void** priorityNonPreemptiveAlgorithm()

{

**int** finishTime[] = **new int**[numberOfProcess]; **int** bt[] = burstTime.clone();

**int** at[] = arrivalTime.clone(); **int** prt[] = priority.clone(); String pid[] = processId.clone();

**int** waitingTime[] = **new int**[numberOfProcess];

**int** turnAroundTime[] = **new int**[numberOfProcess]; sortAccordingArrivalTimeAndPriority(at, bt, prt, pid);

//calculating waiting & turn-around time for each process 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;

//print on console the order of processes along with their finish time & turn around time System.***out***.println("Priority Scheduling Algorithm : "); System.***out***.format("%20s%20s%20s%20s%20s%20s%20s\n", "ProcessId", "BurstTime",

"ArrivalTime", "Priority", "FinishTime", "WaitingTime", "TurnAroundTime");

**for** (**int** i = 0; i < numberOfProcess; i++) { System.***out***.format("%20s%20d%20d%20d%20d%20d%20d\n", pid[i], bt[i], at[i], prt[i],

finishTime[i], waitingTime[i], turnAroundTime[i]);

}

System.***out***.format("%100s%20f%20f\n", "Average", averageWaitingTime, averageTurnAroundTime);

}

**public static void** main(String[] args)

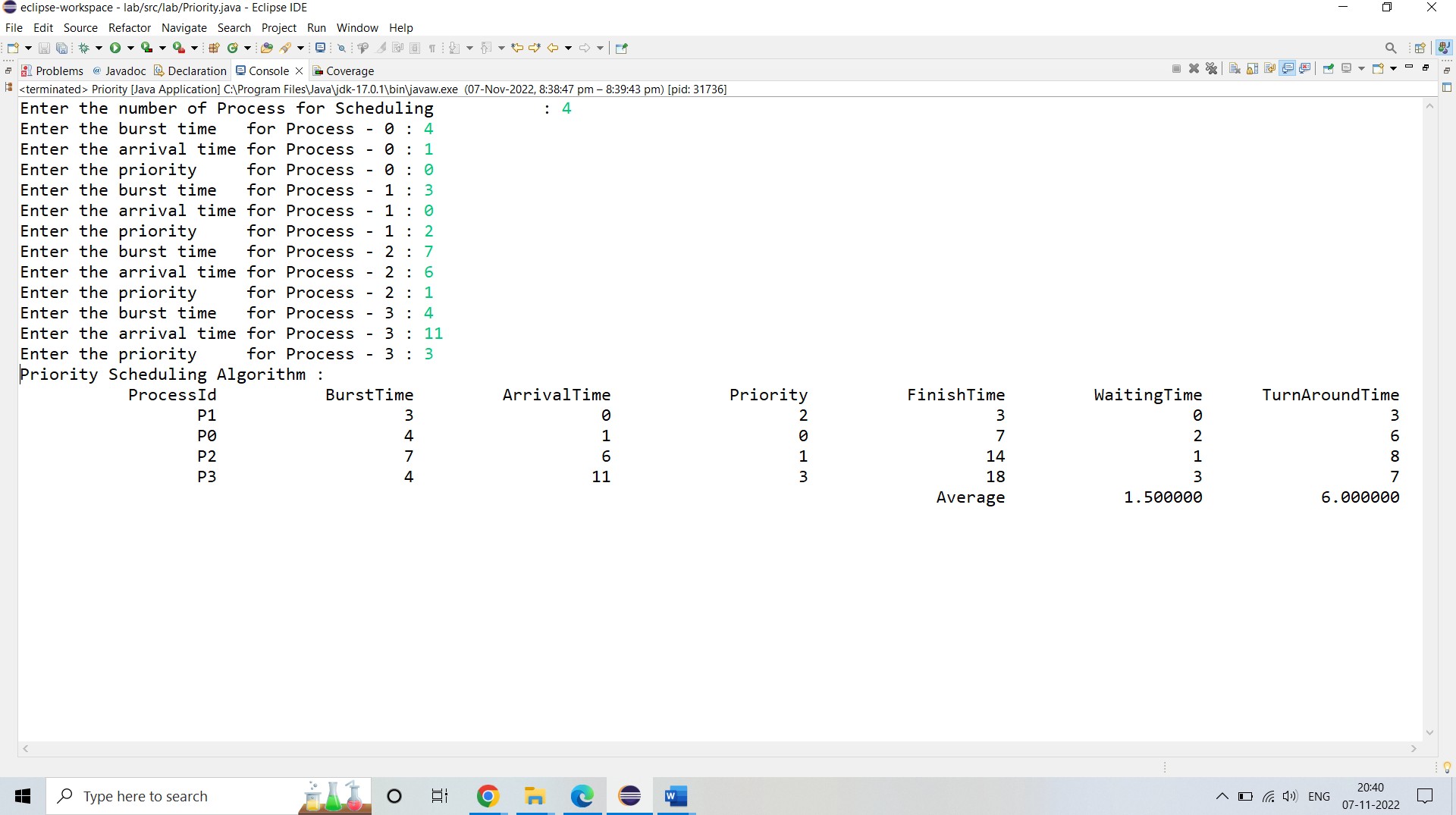
{

Scanner input = **new** Scanner(System.***in***); Priority obj = **new** Priority(); obj.getProcessData(input); obj.priorityNonPreemptiveAlgorithm();

}

}

# OUTPUT:



## Round Robin(Preemtive):

**Source Code: package** lab; **import** java.util.\*; **import** java.io.\*; **class** RoundRobin

{

**public static void** main(String args[])

{

**int** Process[]=**new int**[10]; **int** a[]=**new int**[10];

**int** Arrival\_time[]=**new int**[10]; **int** Burst\_time[]=**new int**[10]; **int** WT[]=**new int**[10];

**int** TAT[]=**new int**[10]; **int** Pno,sum=0;;

**int** TimeQuantum;

System.***out***.println("\nEnter the no. of Process::"); Scanner sc=**new** Scanner(System.***in***); Pno=sc.nextInt();

System.***out***.println("\nEnter each process::");

**for**(**int** i=0;i<Pno;i++)

{

Process[i]=sc.nextInt();

}

System.***out***.println("\nEnter the Burst Time of each process::");

**for**(**int** i=0;i<Pno;i++)

{

}

**do**{

Burst\_time[i]=sc.nextInt();

System.***out***.println("\nEnter the Time Quantum::"); TimeQuantum=sc.nextInt();

**for**(**int** i=0;i<Pno;i++)

{

**if**(Burst\_time[i]>TimeQuantum)

{

Burst\_time[i]-=TimeQuantum;

**for**(**int** j=0;j<Pno;j++)

{

}

**else**

{

**if**((j!=i)&&(Burst\_time[j]!=0)) WT[j]+=TimeQuantum;

}

**for**(**int** j=0;j<Pno;j++)

{

**if**((j!=i)&&(Burst\_time[j]!=0)) WT[j]+=Burst\_time[i];

}

Burst\_time[i]=0;

}

}

sum=0;

**for**(**int** k=0;k<Pno;k++)

sum=sum+Burst\_time[k];

} **while**(sum!=0);

**for**(**int** i=0;i<Pno;i++) TAT[i]=WT[i]+a[i];

System.***out***.println("process\t\tBT\tWT\tTAT");

**for**(**int** i=0;i<Pno;i++)

{

System.***out***.println("process"+(i+1)+"\t"+a[i]+"\t"+WT[i]+"\t"+TAT[i]);

}

**float** avg\_wt=0; **float** avg\_tat=0; **for**(**int** j=0;j<Pno;j++)

{

avg\_wt+=WT[j];

}

**for**(**int** j=0;j<Pno;j++)

{

avg\_tat+=TAT[j];

}

System.***out***.println("average waiting time "+(avg\_wt/Pno)+"\n

Average turn around time"+(avg\_tat/Pno));

}

}

**OUTPUT:**

