

```

import java.util.Scanner;

class Process {
    int pid, burstTime, arrivalTime, priority, completionTime, waitingTime,
    turnaroundTime;

    public Process(int pid, int burstTime, int arrivalTime, int priority) {
        this.pid = pid;
        this.burstTime = burstTime;
        this.arrivalTime = arrivalTime;
        this.priority = priority;
    }
}

class PriorityScheduling {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of processes: ");
        int n = sc.nextInt();
        Process[] processes = new Process[n];

        for (int i = 0; i < n; i++) {
            System.out.print("Enter burst time, arrival time, and priority for process
" + (i + 1) + ": ");
            int burstTime = sc.nextInt();
            int arrivalTime = sc.nextInt();
            int priority = sc.nextInt();
            processes[i] = new Process(i + 1, burstTime, arrivalTime, priority);
        }

        int currentTime = 0, completed = 0, totalWaitTime = 0,
        totalTurnaroundTime = 0;

        while (completed != n) {
            int currentProcessIndex = -1;
            for (int i = 0; i < n; i++) {
                if (processes[i].arrivalTime <= currentTime && processes[i].burstTime
                > 0 &&
                    (currentProcessIndex == -1 || processes[i].priority <
                    processes[currentProcessIndex].priority)) {
                        currentProcessIndex = i;
                    }
            }

            if (currentProcessIndex != -1) {

```

```

        currentTime += processes[currentProcessIndex].burstTime;
        processes[currentProcessIndex].completionTime = currentTime;
        processes[currentProcessIndex].turnaroundTime = currentTime -
processes[currentProcessIndex].arrivalTime;
        processes[currentProcessIndex].waitingTime =
            processes[currentProcessIndex].turnaroundTime -
processes[currentProcessIndex].burstTime;
        totalWaitTime += processes[currentProcessIndex].waitingTime;
        totalTurnaroundTime +=
processes[currentProcessIndex].turnaroundTime;
        processes[currentProcessIndex].burstTime = 0;
        completed++;
    } else {
        currentTime++;
    }
}

System.out.println("PID\tWaiting Time\tTurnaround Time");
for (Process p : processes) {
    System.out.println(p.pid + "\t" + p.waitingTime + "\t" +
p.turnaroundTime);
}

System.out.println("Avg waiting time: " + (float) totalWaitTime / n);
System.out.println("Avg turnaround time: " + (float) totalTurnaroundTime /
n);
}
}
//Output

```

```

hiteshikukreja@192 javapractice % javac priority.java
hiteshikukreja@192 javapractice % java PriorityScheduling
Enter number of processes: 5
Enter burst time, arrival time, and priority for process 1: 10 0 3
Enter burst time, arrival time, and priority for process 2: 1 0 1
Enter burst time, arrival time, and priority for process 3: 2 0 4
Enter burst time, arrival time, and priority for process 4: 1 0 5
Enter burst time, arrival time, and priority for process 5: 5 0 2
PID  Waiting Time  Turnaround Time
1    6             16
2    0             1
3    16            18
4    18            19
5    1             6
Avg waiting time: 8.2
Avg turnaround time: 12.0

```