import java.util.\*;

public class FCFS\_Scheduling {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Input number of processes

System.out.print("Enter number of processes: ");

int n = sc.nextInt();

Process[] processes = new Process[n];

// Input arrival and burst time for each process

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

System.out.println("\nEnter details for Process " + (i + 1));

System.out.print("Arrival Time: ");

int at = sc.nextInt();

System.out.print("Burst Time: ");

int bt = sc.nextInt();

processes[i] = new Process(i + 1, at, bt);

}

// Sort processes by arrival time

Arrays.sort(processes, Comparator.comparingInt(p -> p.arrivalTime));

int currentTime = 0;

for (Process p : processes) {

if (currentTime < p.arrivalTime) {

currentTime = p.arrivalTime; // CPU idle

}

p.completionTime = currentTime + p.burstTime;

p.turnaroundTime = p.completionTime - p.arrivalTime;

p.waitingTime = p.turnaroundTime - p.burstTime;

currentTime = p.completionTime;

}

// Display process table

System.out.println("\nPID\tAT\tBT\tCT\tTAT\tWT");

for (Process p : processes) {

System.out.printf("P%d\t%d\t%d\t%d\t%d\t%d\n",

p.pid, p.arrivalTime, p.burstTime, p.completionTime,

p.turnaroundTime, p.waitingTime);

}

// Display Gantt Chart

System.out.println("\nGantt Chart:");

System.out.print("|");

for (Process p : processes) {

System.out.print(" P" + p.pid + " |");

}

System.out.println();

System.out.print(processes[0].arrivalTime);

for (Process p : processes) {

System.out.printf("%6d", p.completionTime);

}

sc.close();

}

}

class Process {

int pid, arrivalTime, burstTime, completionTime, waitingTime, turnaroundTime;

public Process(int pid, int arrivalTime, int burstTime) {

this.pid = pid;

this.arrivalTime = arrivalTime;

this.burstTime = burstTime;

}

}