RoundRobin-6

```
import java.util.*;

class RoundRobin {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter number of processes: ");
        int n = sc.nextInt();

        int pid[] = new int[n], at[] = new int[n], bt[] = new int[n], rem[] = new int[n];
        int wt[] = new int[n], tat[] = new int[n], ct[] = new int[n];

        for(int i = 0; i < n; i++) {
            System.out.print("Process ID: ");
            pid[i] = sc.nextInt();
            System.out.print("Arrival Time: ");
            at[i] = sc.nextInt();
            System.out.print("Burst Time: ");
            bt[i] = sc.nextInt();
            rem[i] = bt[i];
            System.out.println();
        }

        System.out.print("Enter Time Quantum: ");
        int tq = sc.nextInt();

        int time = 0, completed = 0;
        Queue<Integer> q = new LinkedList<>();
        boolean inQueue[] = new boolean[n];
        ArrayList<String> gantt = new ArrayList<>();
```

```
// Add first process
for(int i = 0; i < n; i++)
    if(at[i] == 0) { q.add(i); inQueue[i]=true; break; }

while(completed < n) {

    if(q.isEmpty()) {
        time++;
        for(int i = 0; i < n; i++)
            if(at[i] <= time && !inQueue[i] && rem[i] > 0) {
                q.add(i); inQueue[i] = true; break;
            }
        continue;
    }

    int i = q.poll();
    int exec = Math.min(tq, rem[i]);
    rem[i] -= exec;
    time += exec;
    gantt.add("P" + pid[i]);

    // check arrivals
    for(int j = 0; j < n; j++) {
        if(at[j] <= time && rem[j] > 0 && !inQueue[j]) {
            q.add(j); inQueue[j] = true;
        }
    }

    if(rem[i] > 0) q.add(i);
    else {
```

```
                completed++;
                ct[i] = time;
                tat[i] = ct[i] - at[i];
                wt[i] = tat[i] - bt[i];
            }
        }

        double avgWT = 0, avgTAT = 0;
        System.out.println("\nPID\tAT\tBT\tWT\tTAT");
        for(int i = 0; i < n; i++) {
            avgWT += wt[i];
            avgTAT += tat[i];
            System.out.println(pid[i] + "\t" + at[i] + "\t" + bt[i] + "\t" + wt[i] + "\t" + tat[i]);
        }

        System.out.println("\nGantt Chart:");
        for(String g : gantt) System.out.print("| " + g + " ");
        System.out.println("|");

        System.out.println("\nAverage Waiting Time: " + avgWT/n);
        System.out.println("Average Turnaround Time: " + avgTAT/n);
    }
}
```