

```

import java.util.*;
public class RoundRobin {
    static class Task
    {
        int arrive;
        int remain;
        public Task(int arrive, int remain) {
            this.arrive = arrive;
            this.remain = remain;
        }
    }
    public static double roundRobin(int[] arriveTime, int[] runTime, int slot)
    {
        Queue<Task> queue = new LinkedList<>();
        int i = 0, t = 0, wait = 0;
        while(i < arriveTime.length || !queue.isEmpty()) {
            if(!queue.isEmpty()) {
                Task peek = queue.poll();
                wait += (t - peek.arrive);
                if(peek.remain > slot) {
                    t += slot;
                    peek.remain -= slot;
                    peek.arrive = t;
                } else {
                    t += peek.remain;
                    peek.remain = 0;
                    peek.arrive = t;
                }
                while(i < arriveTime.length && arriveTime[i] <= t) {
                    queue.offer(new Task(arriveTime[i], runTime[i]));
                    i++;
                }

                if(peek.remain != 0) queue.offer(peek);
            } else {
                queue.offer(new Task(arriveTime[i], runTime[i]));
                t = arriveTime[i];
                i++;
            }
        }
        return (wait + 0.0) / arriveTime.length;
    }
}

```