Problem H
A Kind of Multicore Virtual Server

Time Limit: 3 seconds

Memory Limit: 512 Megabytes

Problem description

Nam, Bac, Dong, and Tay are students of the computer science department at ACM Lab, and they have each one a classical computer with one core. Recently, they studied multi-cores CPU, and they would like to build up a simulation multi-cores CPU Server by creating a software run on a computer owned by the department to combine their own 4 classical computers into a virtual one.

The simulation software will receive tasks raised by the department computer's operating system in each time-sliced unit of computation. Some tasks reached simulation software at the same time. Each task will be assigned a PID for identification. When a task reached the simulation software in a time-sliced unit, it will be processed in the next time-sliced unit in the simulation software. The simulation software will go around to assigned tasks to CPUs based on its PID when it reached to assigning phase, and the assignment process will start from the CPU, which is the next one with the last assigned CPU in turning around, i.e.

If the last assigned CPU is 1st, the start one is 2nd CPU.

If the last assigned CPU is 2nd, the start one is the 3rd CPU.

If the last assigned CPU is 3rd, the start one is 4th CPU.

If the last assigned CPU is 4th, the next start is 1st.

And so on.

In each CPU, a task's processing time consumption – called burst time - will also be sliced into some unit pieces, and it will be allocated into the CPU computation time slots in around.

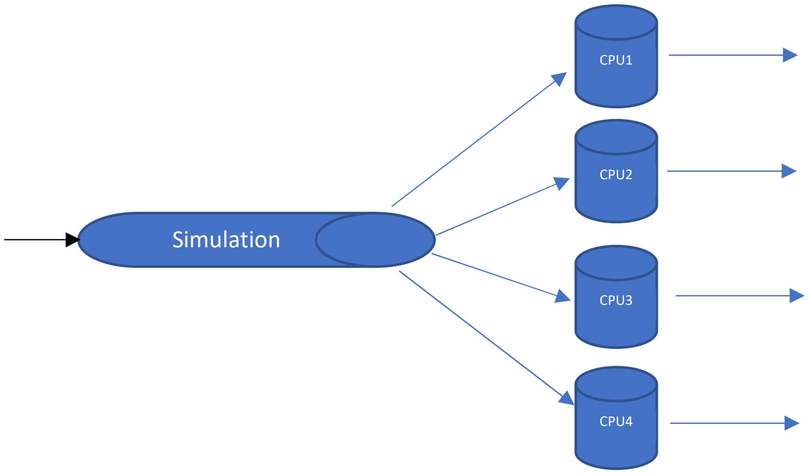

Figure 2. Simulation and CPUs structure

When a task reaches a CPU, it will be allocated in processing waiting list, and the CPU will assign them a fixed time slot in a cyclic way for processing. In case a Task/Process had been completed earlier than the end of time slot, the CPU will immediately pick the next Task/Process in the processing wait list to execute.

1. Completion time: Time at which process completes its execution.
2. Turn Around Time: Time Difference between completion time and CPU arrival time. Turn Around Time = Completion Time – CPU Arrival Time
3. Waiting Time: Time Difference between turn around time and burst time. Waiting Time = Turn Around Time – Burst Time
4. Total Turn Around Time: is computed by sum all of the turn around time of processes.
5. Total Waiting Time is the sum of the waiting time of all processes.
6. Average Turn Around Time in each CPU is equal to the total average turn around time divide with the number of tasks/processes.
7. Average Waiting Time in each CPU is equal to the total waiting time divide with the number of tasks/processes.
8. Average Turn Around Time of a task in the Virtual Server is equal to the sum of all Average Turn Around Time in each CPU devide with number of CPUs
9. Average Waiting Time of a task in the Virtual Server is equal to the sum of all Average Waiting Time in each CPU devide with number of CPUs

Assume that the timer in all CPUs and simulation machine are synchronized. The simulation software doesn't care about busy or leisure state of CPUs while assigning processes to them.

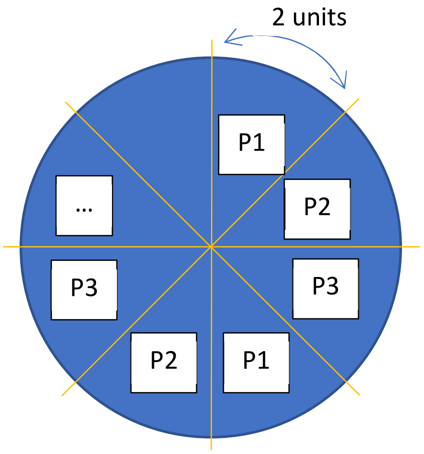

Figure 3. a time slot in CPU processing

Input

First line:

N: number of Tasks/Processes will be executed – integer number ($1 < N <= 10^3$)

slot: length of a CPU processing slot in time sliced unit – integer number. ($1 < slot <= 10^2$)

Next N lines contain the information of each task. i^{th} line contains 3 values, separated by a space

- pid – tasks/processes id - integer number ($0 < pid < 10^6$)
- bt_i - burst time of task i^{th}, integer number ($0 < bt_i < 10^6$) - in time sliced unit
- rt_i - reached time of task ith on simulation software, integer number ($-1 <= rti < 10^6$) - in time sliced unit. In case rt_i is equal -1, the process i^{th} will be assigned to a CPU immediately at the first assigning phase of simulation software.

Output

One lines with 2 values in time sliced unit, separated by a space.

Wt: a floating number - Average waiting time of a task in the Virtual Server

Tt: a floating number - Average turn around time of a task in the Virtual Server

Floating point value accuracy is 10^{-6}.

Example 1:

Input	**Output**
1 2 1 4 5	0.250000 1.250000

Example 2:

Input	**Output**

2 2
1 4 8
2 2 9
	0.500000 2.000000

Example 3:

Input	**Output**

2 2
1 4 0
2 4 0
	0.500000 2.500000

Example 4:

Input	**Output**

2 2
1 4 -1
2 8 -1
	0.500000 3.500000