

There are  $n$  processes in a queue. Each process has  $name_i$  and  $time_i$ . The round-robin scheduling handles the processes in order. A round-robin scheduler gives each process a *quantum* (a time slot) and interrupts the process if it is not completed by then. The process is resumed and moved to the end of the queue, then the scheduler handles the next process in the queue.

For example, we have the following queue with the quantum of 100ms.

A(150) - B(80) - C(200) - D(200)

First, process A is handled for 100ms, then the process is moved to the end of the queue with the remaining time (50ms).

B(80) - C(200) - D(200) - A(50)

Next, process B is handled for 80ms. The process is completed with the time stamp of 180ms and removed from the queue.

C(200) - D(200) - A(50)

Your task is to write a program which simulates the round-robin scheduling.

### Input

```
 $n$   $q$ 
 $name_1$   $time_1$ 
 $name_2$   $time_2$ 
...
 $name_n$   $time_n$ 
```

In the first line the number of processes  $n$  and the quantum  $q$  are given separated by a single space.

In the following  $n$  lines, names and times for the  $n$  processes are given.  $name_i$  and  $time_i$  are separated by a single space.

### Output

For each process, prints its name and the time the process finished in order.

### Constraints

- $1 \leq n \leq 100000$
- $1 \leq q \leq 1000$
- $1 \leq time_i \leq 50000$
- $1 \leq \text{length of } name_i \leq 10$
- $1 \leq \text{Sum of } time_i \leq 1000000$

### Sample Input 1

```
5 100
p1 150
p2 80
p3 200
p4 350
p5 20
```

### Sample Output 1

```
p2 180
p5 400
p1 450
p3 550
p4 800
```

### Notes

[Template in C](#)