A. SMSC

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Some large corporation where Polycarpus works has its own short message service center (SMSC). The center's task is to send all sorts of crucial information. Polycarpus decided to check the efficiency of the SMSC.

For that, he asked to give him the statistics of the performance of the SMSC for some period of time. In the end, Polycarpus got a list of n tasks that went to the SMSC of the corporation. Each task was described by the time it was received by the SMSC and the number of text messages to send. More formally, the i-th task was described by two integers t_i and c_i — the receiving time (the second) and the number of the text messages, correspondingly.

Polycarpus knows that the SMSC cannot send more than one text message per second. The SMSC uses a queue to organize its work. Consider a time moment x, the SMSC work at that moment as follows:

1. If at the time moment x the queue is not empty, then SMSC sends one message from the queue (SMSC gets the message from the head of the queue). Otherwise it doesn't send messages at the time moment x.
2. If at the time moment x SMSC receives a task, then it adds to the queue all the messages from this task (SMSC adds messages to the tail of the queue). Note, that the messages from the task cannot be send at time moment x. That's because the decision about sending message or not is made at point 1 before adding these messages to the queue.

Given the information about all n tasks, Polycarpus wants to count two values: the time when the last text message was sent and the maximum size of the queue at some time. Help him count these two characteristics he needs to evaluate the efficiency of the SMSC.

Input

The first line contains a single integer n $(1 \le n \le 10^3)$ — the number of tasks of the SMSC. Next n lines contain the tasks' descriptions: the i-th line contains two space-separated integers t_i and c_i $(1 \le t_i, c_i \le 10^6)$ — the time (the second) when the i-th task was received and the number of messages to send, correspondingly.

It is guaranteed that all tasks were received at different moments of time. It is guaranteed that the tasks are sorted in the chronological order, that is, $t_i < t_{i+1}$ for all integer i $(1 \le i < n)$.

Output

In a single line print two space-separated integers — the time when the last text message was sent and the maximum queue size at a certain moment of time.

Examples

input

```
2
1 1
2 1
```

output

```
3 1
```

input

```
1
1000000 10
```

output

```
1000010 10
```

input

```
3
3 3
4 3
5 3
```

output

```
12 7
```

Note

In the first test sample:

- second 1: the first message has appeared in the queue, the queue's size is 1;
- second 2: the first message is sent, the second message has been received, the queue's size is 1;
- second 3: the second message is sent, the queue's size is 0,

Thus, the maximum size of the queue is 1, the last message was sent at the second 3.