

At the hairdresser there is one master. He spends exactly 2020 minutes on one client, and then immediately proceeds to the next one, if someone is in the queue, or is waiting for the next client to arrive.

Given the times of customers arriving at the hairdresser (in the order in which they arrived).

Each client also has a characteristic called the degree of impatience. It shows how many people can be in the line in front of the client as much as possible so that he waits for his turn and does not leave earlier. If at the moment of arrival of the client there are more people in the queue than the degree of his impatience, then he decides not to wait for his turn and leaves. The client who is currently being served is also considered to be in the queue.

It is required for each client to indicate the time of his exit from the hairdresser.

The first line introduces a positive integer N not exceeding 10^5 – number of clients.

Each of the following N lines contains three integers, the first two of which specify the time of arrival of the client, indicating hours and minutes (hours – from 0 to 16 000 000, minutes – from 0 to 59), the third number sets the degree of his impatience (non-negative integer no greater than 10^5) – the maximum number of people that he is ready to wait in front of himself in the queue. Times are shown in ascending order (all times are different).

If for some clients the time of the end of service of one client and the time of arrival of another coincide, then we can assume that at the beginning the service of the first client ends, and then the second client arrives.

Print N pairs of numbers: exit times from the hairdresser on the 1th, 2nd, ..., N th client (hours and minutes). If at the time of the client's arrival, the person in the queue is more than the degree of his impatience, then we must assume that the time of his departure is equal to the time of arrival.

Sample input 1:

```
3
10 0 0
10 1 1
10 2 1
```

Sample output 1:

```
10 20
10 40
10 2
```

Sample input 2:

```
5
1 0 100
2 0 0
2 1 0
2 2 3
2 3 0
```

Sample output 2:

```
1 20
2 20
2 1
2 40
2 3
```