Programming task Hairdressers

Memory limit: 4 MB
Time limit: 0.2 second
Input file: `hair.in`
Output: `hair.out`

Description

In a big city (more than one million of inhabitants, though no more than a billion) there is a one and only hair studio, where only a few hairdressers work $[1..9]$. Each of the hairdressers has its own number $[1..9]$, in order to organize the studio work more efficiently. The studio counts the time in certain time units $[1\,..\,2\,000\,000\,000]$, and counting starts at the studio opening moment.

As there is an immense number of customers, and the demand for hairdressers is very high, each of them obligatory has to go for a break. The time of a mandatory break for each hairdresser is when a digit in the hundred position in time number coincide with the hairdresser's number, i.e. the hairdresser with number 5 has to go for a break in the time moments $[500..599]$, $[1500..1599]$, $[2500..2599]$, etc. During a break, hairdresser is forbidden to serve a client. In addition, customer appointment cannot be divided in stages, i.e. customer can only be served by one hairdresser without any breaks. Consequently, a hairdresser cannot start to serve a client, if the service cannot be finished before the moment of the break.

A customer should be served without delay if there is an unoccupied hairdresser and she/he does not have any limitations regarding this work. Upon finishing work with a current client, a hairdresser in the next moment should try to start working with the next one. More precisely: client C_1 shows up at the time moment T_1 and his appointment needs time (serving duration) D_1. The hairdresser H_1 is free. Consequently, this appointment will take place during time interval $[T_1\,..\,T_1 + D_1 - 1]$. The appointment is finished at the time moment $T_1 + D_1 - 1$. If customer C_2 has already shown up before or exactly during time moment $T_1 + D_1$, then hairdresser H_1 can start working with customer C_2 at the time moment $T_1 + D_1$.

Customers wait in the exact (honest) queue. During one time moment, only one customer can show up. Administrator immediately assigns a number in queue $[1\,..\,200\,000]$ and appointment duration $[1..900]$.

If there is an unserved customer and there are multiple hairdressers that lay claim on her, then:

1. The hairdresser that has spent more time without a customer (counting from the end of the appointment), has the priority;
2. If two hairdressers had spent the same time without the client, the hairdresser with the smaller number has the priority

Knowing the time, when customers have shown up, their number in the queue, the duration of the appointment, print the customer appointment end time, the number of serving hairdresser and the customer number in the queue. The records should be printed ordered by time in the ascending order. If two client appointments were finished at the same time, results should be ordered by the hairdresser number.

Input:

The first line has the integer number that is a number of hairdressers. Then follows the information about the customers in the order of showing up (Time values ascend).

```
Hairdressers
Time Customer Duration
...
0
```

- `Hairdressers` denotes the number of hairdressers $[1..9]$
- `Time` denotes the moment in time, when customer has shown up and is ready to have an appointment $[1\,..\,2\,000\,000\,000]$
- `Customer` denotes the customer number in the queue $[1\,..\,200\,000]$
- `Duration` denotes the necessary length of the appointment $[1..900]$
- `0` means the end of input data. In this case, `Customer` and `Duration` fields are not provided

Input data are in precise chronological order.

The input file is correct regarding the input data format and given conditions.

Output:

According to the input file, each of the customer appointment end time is written in the following format:

```
Time Hairdresser Customer
```

- `Time` denotes the moment in time, which is the last customer appointment moment $[1\,..\,2\,000\,000\,000]$
- `Hairdresser` denotes the hairdresser that worked with the customer, i.e. the hairdresser number $[1..9]$
- `Customer` denotes the customer that had the appointment, number in queue $[1\,..\,200\,000]$

In the result, none of the appointment end time will not be greater than $2\,000\,000\,000$.

Example:

The content of input file `hair.in`:

```
2
11 1 10
21 2 50
31 3 20
0
```

The content of output file `hair.out`:

```
20 1 1
50 1 3
70 2 2
```

Additional explanations:

Customers must be served in a strict queue

There should not be a situation when a customer **starts** to be served **faster** than a customer who arrived before that.

One hairdresser

The content of input file `hair.in`:

```
1
10 1 100
20 2 10
0
```

The content of output file `hair.out`:

```
299 1 1
309 1 2
```

Two hairdressers compete in time 300

The content of input file `hair.in`:

```
2
110 1 100
120 2 10
0
```

The content of output file `hair.out`:

```
299 1 1
309 2 2
```

Two hairdressers

The content of input file `hair.in`:

```
2
110 1 99
120 2 10
0
```

The content of output file `hair.out`:

```
298 1 1
308 1 2
```

Several customer service can start at the same time

It is allowed to start serving several customers at the same time. This is not considered a queue violation.

Two hairdressers are freeing at the same time and there are waiting customers

The content of input file `hair.in`:

```
2
1 1 10
2 2 9
3 3 10
4 4 10
0
```

The content of output file `hair.out`:

```
10 1 1
10 2 2
20 1 3
20 2 4
```