# H. Heap Manager

In this problem, you're to implement a heap manager.

There are $n$ memory units in the heap, whose address range from 0 to $n$-1. Each memory unit is either free or occupied. If there are $k$ consecutive free memory units $a$, $a$+1, $a$+2, ..., $a$+$k$-1, we call it a free memory slice of length $k$, starting from address $a$.

There will be some processes which use the manager to allocate memory. We use a triple ($t$, $m$, $p$) to represent a process who requests for a memory slice of length $m$ at time $t$, and needs $p$ time units to run. Let $t'$ be the time when this process successfully allocated memory, then the memory slice will be free at time $t'$+$p$.

Processes will be sorted in ascending order of $t$, and no two processes are allocating memory simultaneously. For each process ($t$, $m$, $p$), we do the following:
∎ If there is a free memory slice of length $m$ at time $t$, it is allocated to the process. If there are more than one such slice, use the one with smallest starting address.
∎ If there is no such slice, place the process in a waiting queue. *Whenever* (for example, just after freeing some memory) there is a suitable memory slice for the first process in the queue, we remove the process from the queue and allocate memory for it. Other processes in the queue will not be considered until they become the first process in the queue. Note that new requests are processed only when the first process in the queue cannot allocate memory (or the queue is empty).

## Input
There are multiple test cases. Each test case begins with two integers $n$ ($10<=n<=10^9$) and $b$ ($0<=b<=1$), where $n$ is the number of memory cells, and $b$ is whether or not the events should be printed. There will be no more than 200,000 lines followed, each containing three integers $t$, $m$, $p$ ($m<=n$, $0<=t<=10^9$, $0<p<=10^9$). The processes are sorted in increasing order of $t$, and no two processes have identical $t$. The process list in each test case terminates with three zeros. The size of input does not exceed 10MB.

## Output
For each test case, first print all the events, in the order they happened, if $b$=1 (if $b$=0, the events should not be printed). Each event is formatted as "T i a", that means at time T, the i-th process (counting from 1) has successfully allocated a memory slice beginning at $a$. Then two lines contains two integers (one for each line), the time that all the processes have finished, and the number of processes that are ever placed in the queue. Print an empty line after each test case.

## Sample Input
```
10 1
1 3 10
2 4 3
3 4 4
4 1 4
5 3 4
0 0 0
4 1
0 3 5
1 1 4
2 2 2
```

```
3 1 1
4 2 1
5 1 3
0 0 0
4 1
0 3 5
1 1 1
2 2 2
3 1 1
4 2 1
5 1 3
0 0 0
```

## Output for Sample Input

```
1 1 0
2 2 3
4 4 7
5 3 3
8 5 7
12
2

0 1 0
1 2 3
5 3 0
5 4 2
5 6 3
7 5 0
8
3

0 1 0
1 2 3
3 4 3
5 3 0
5 5 2
6 6 2
9
3
```