

Memory Distribution

EMS memory (called **memory** for short) is some important resource of a computer. When a program is running, the computer must distribute the memory for it.

The classical memory distribution process is like the following:

- The basic unit of the memory is called a **cell**, each cell is assigned to a fixed integer number called its **address**. The address starts from number 0. If two cell's address numbers are two consecutive integer numbers, the two cells are called (logically) consecutive. We name the s consecutive cells starting from address i a piece whose length is s and first address is i .
- Many programs need memory during their running. For each program, we need a application time X , a number of cells needed M and a running time P to describe it. When the program is running(during it starts running (time T) to $T+P$, including the left end and excluding the right end),The M cells cannot be used by other programs.
- Suppose a program apply M cells at time X and its running time is P ,then
 - (A) If there is a piece in the memory at time X , each cell of which is not being used, and whose length is M , the computer will distribute these M cells to this program. If there are more pieces, the computer will choose the one whose first address is minimum.
 - (B) If such piece does not exist at time X , the program will be put into a queue.If after some time, there exist a piece whose length is M and the corresponding program is at the head of the queue, the computer will pop this program and distribute memory for it like (A) immediately. During the process of memory distribution, the programs which are not at the head of the queue cannot start to run before the one at the head of the queue.

Input

Ten test cases (given one after another, you have to process all!).For each test case:

The first line is a number N , which shows the number of memory cells. There addresses are $0..n-1$. Less than 10000 lines follow, each contains 3 integers $X, M(M \leq N), P$ describing the programs. A line containing three zeroes denotes the end of a test case. The programs have been sorted by there application time X .

All numbers in the input and output file will be less than 10^9 .

Output

For each test case output two lines. The first line contains a single integer, which shows the time when all the programs have been run and stops normally. The second line contains a single integer, which is the number of programs which has been put into the queue.

Example

Input:

10

1 3 10
2 4 3
3 4 4
4 1 4
5 3 4
0 0 0

[and 9 test cases more]

Output:

12
2

[and 9 test cases more]