

## E. Printer

time limit per test: 4 seconds  
memory limit per test: 256 megabytes

**input:** input.txt  
**output:** output.txt

Let's consider a network printer that functions like that. It starts working at time 0. In each second it can print one page of a text. At some moments of time the printer receives printing tasks. We know that a printer received  $n$  tasks. Let's number the tasks by consecutive integers from 1 to  $n$ . Then the task number  $i$  is characterised by three integers:  $t_i$  is the time when the task came,  $s_i$  is the task's volume (in pages) and  $p_i$  is the task's priority. The priorities of all tasks are distinct.

When the printer receives a task, the task goes to the queue and remains there until all pages from this task are printed. The printer chooses a page to print each time when it either stops printing some page or when it is free and receives a new task. Among all tasks that are in the queue at this moment, the printer chooses the task with the highest priority and next second prints an unprinted page from this task. You can assume that a task goes to the queue immediately, that's why if a task has just arrived by time  $t$ , the printer can already choose it for printing.

You are given full information about all tasks except for one: you don't know this task's priority. However, we know the time when the last page from this task was finished printing. Given this information, find the unknown priority value and determine the moments of time when the printer finished printing each task.

### Input

The first line contains integer  $n$  ( $1 \leq n \leq 50000$ ). Next  $n$  lines describe the tasks. The  $i$ -th of these lines contains three integers  $t_i$ ,  $s_i$  and  $p_i$ , separated by single spaces ( $0 \leq t_i \leq 10^9$ ,  $1 \leq s_i, p_i \leq 10^9$ ). Exactly one task (let's assume that his number is  $x$ ) has number  $-1$  written instead of the priority. All priorities are different. The last line contains integer  $T$  — the time when the printer finished printing the last page of task  $x$  ( $1 \leq T \leq 10^{15}$ ). Numbers  $t_i$  are not necessarily distinct. The tasks in the input are written in the arbitrary order.

### Output

In the first line print integer  $p_x$  — the priority of the task number  $x$  ( $1 \leq p_x \leq 10^9$ , remember that all priorities should be distinct). Then print  $n$  integers, the  $i$ -th of them represents the moment of time when the last page of the task number  $i$  finished printing.

It is guaranteed that at least one solution exists. If there are multiple solutions, print any of them.

### Examples

input
3 4 3 -1 0 2 2 1 3 3 7
output
4 7 8 4

input
3 3 1 2 2 3 3 3 1 -1 4
output

## Note

Let's consider the first test case. Let's assume that the unknown priority equals 4, then the printer's actions for each second are as follows:

- the beginning of the 1-st second (time 0). The queue has task 2. The printer prints the first page of this task;
- the beginning of the 2-nd second (time 1). The queue has tasks 2 and 3. The printer prints the first page of task 3;
- the beginning of the 3-rd second (time 2). The queue has tasks 2 and 3. The printer prints the second page of task 3;
- the beginning of the 4-th second (time 3). The queue has tasks 2 and 3. The printer prints the third (last) page of task 3. Thus, by the end of the 4-th second this task will have been printed;
- the beginning of the 5-th second (time 4). The queue has tasks 2 and 1. The printer prints the first page of task 1;
- the beginning of the 6-th second (time 5). The queue has tasks 2 and 1. The printer prints the second page of task 1;
- the beginning of the 7-th second (time 6). The queue has tasks 2 and 1. The printer prints the third (last) page of task 1. Thus, by the end of the 7-th second this task will have been printed;
- the beginning of the 8-th second (time 7). The queue has task 2. The printer prints the second (last) page of task 2. Thus, by the end of the 8-th second this task will have been printed.

In the end, task number 1 will have been printed by the end of the 7-th second, as was required. And tasks 2 and 3 are printed by the end of the 8-th and the 4-th second correspondingly.