

NOI '99 P6 - Memory Allocation

Time limit: 0.6s **Memory limit:** 16M

National Olympiad in Informatics, China, 1999

Memory is an important resource of computers, and running processes require memory to be allocated for them.

The classic memory allocation process is as follows:

1. The minimally addressable *storage unit* is the most basic unit for memory. Each storage unit is identified using a certain integer which is known as the *address*. Addresses begin at 0 and go up consecutively. Storage units with consecutive addresses are considered logically *contiguous*. We shall call the s contiguous storage units starting at address i the *memory block* at i of length s .
2. While the computer is running, there may be several *processes* that will need to take up memory. Each process will have a requested time T , the number of storage units M and the running time P . Within the running duration P (the process starts at T and terminates at $T + P$), the M occupied storage units will not be accessible to any other process.
3. If at time T there is a process requesting M storage units with a running time P , then:
 1. If at time T there exists an unused memory block of length M , then the system will assign these M storage units to the process. If there are multiple length M blocks, then the system will assign the process the block with the smallest initial address.
 2. If at time T there does not exist an unused memory block of length M , then the process will be placed in a waiting queue. Regarding the process at the head of the waiting queue, if at any time there exists an unused memory block of its requested size M , then the system will *immediately* remove the process from the waiting queue and allocate this block to it. Note that while allocating memory, the process at the head of the queue has the highest priority, and no other processes in the queue may be dealt with before this process.

Given information about the processes, please write a program that simulates the memory allocation of the system.

Input Specification

The first line of input is an integer N denoting the total number of storage units in memory (the range of available addresses is from 0 to $N - 1$).

Starting at the second line of input, each line will describe a single process with three integers T , M , and P ($M \leq N$). The processes will be given in increasing order of T .

The last line of input is denoted by three zeros.

There are at most 10 000 lines of input, and all values will be less than 10^9 .

Adjacent values on the same line of the input will be separated by one or more spaces.

Output Specification

The output should contain two lines.

The first line should contain the time at which all processes terminate.

The second line should contain the total number of processes that have been placed in the waiting queue.

Sample Input

```

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

Sample Output

```

12
2
    
```

Explanation

Time T	Memory Usage Status									Description	
	0	1	2	3	4	5	6	7	8		9
1	A										Process A requests memory ($M = 3, P = 10$) <Succeeded>
2	A			B							Process B requests memory ($M = 4, P = 3$) <Succeeded>
3	A			B							Process C requests memory ($M = 4, P = 4$) <Failed, added to queue>
4	A			B			D				Process D requests memory ($M = 1, P = 4$) <Succeeded>
5	A			C			D				Process B terminated with its memory freed up. Process C is removed from the queue and memory is allocated. Process E requests memory ($M = 3, P = 4$) <Failed, added to queue>
6	A			C			D				
7	A			C			D				
8	A			C			E				Process D terminated with its memory freed up. Process E is removed from the queue and memory is allocated.
9	A						E				Process C terminated with its memory freed up.
10	A						E				
11							E				Process A terminated with its memory freed up.
12											Process E terminated with its memory freed up.

