PROBLEM 5 – LOLLIES

Every day on his way home, little Billy passes by his great aunt Clara Mitchum's house. Generally he stops in for a chat with the great ACM (as he lovingly refers to her) and sometimes he asks for some lollies. When he does, she generally gives him some, but then adds "now don't be asking for any more for another N days" where N is some positive integer. If $N = 1$ that means he can ask for some on the next day, but for example if it is April 6 and $N = 4$ then he must wait until April 10 or later before asking for more lollies.

One day Billy happened to catch sight of the great ACM's calendar, and noted that each day was marked with two integers. He also noted that the first of these referred to the number of lollies the great ACM would give him on a particular day, and the second to the delay that would then be required before making another request. He copied down as much of the information as he could, and has passed it to you to analyse. His objective, of course, is to get as many lollies as he can.

Your task is to write a program which will report the total number of lollies that can be obtained by Billy, and provide a schedule for obtaining that amount. In the event that there are two or more ways to obtain the maximum number of lollies, Billy will choose the one where his first collection is as late as possible, and among all collections with that first date, his second collection is as late as possible, and so on.

INPUT FORMAT

The input text consists of a number of sets of unrelated problems. The first line of a set is a problem title consisting of a string of 1 to 20 letters. A single # on a line indicates the end of input.

The "title" line is followed by a sequence of "day" lines. Each problem set contains between 1 and 100 days, including the limits. In the given order, the first "day" line corresponds to day number 1, the second line to day number 2, the n-th line to day number n. Each "day" line consists of two integers separated by a single space:

- an integer L, which is the number of lollies available on that day ($1 \leq L \leq 100$),

- an integer N, which is the associated delay ($1 \leq N \leq 100$).

Conventionally, a delay N pointing to a day beyond the end of the current problem refers to a day with zero lollies and zero further delays ($L = 0$, $N = 0$).

SAMPLE INPUT:

```
January
1 1
2 2
3 3
February
10 3
7 1
5 2
1 1
March
2 3
1 1
3 7
2 7
#
```

OUTPUT FORMAT

Each report must follow the following format (use single spaces for spacing):

```
In <problem_title> <total_amount> <lollies> can be obtained:
On day <day_number> collect <day_amount> <lollies>.
On day <day_number> collect <day_amount> <lollies>.
…
```

In this notation, *<problem_title>* represents the actual problem title, *<total_amount>*, *<day_amount>*, and *<day_number>* are numbers with self-described meaning, and *<lollies>* stands for either "lolly" or "lollies", as required by the context (the singular and plural forms must be used appropriately). Days must be given in increasing sequence numbers. Each group report should be separated from the next by a blank line.

SAMPLE OUTPUT:

```
In January 4 lollies can be obtained:
On day 1 collect 1 lolly.
On day 3 collect 3 lollies.

In February 12 lollies can be obtained:
On day 2 collect 7 lollies.
On day 3 collect 5 lollies.

In March 4 lollies can be obtained:
On day 2 collect 1 lolly.
On day 3 collect 3 lollies.
```