Automatized Mineral Classification

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

This task is interactive. After printing each line, you should flush the output buffer. You can use `cout << flush` in C++, `System.out.flush()` in Java, and `sys.stdout.flush()` in Python. You must strictly follow the instructions in the *Interaction* section; otherwise **you may receive verdicts like *wrong answer*, *time limit exceeded*, or others**.

—

Because work in the mine is hard and time-consuming, the dwarves decided to seek help in automating some of their tasks. Their current task is to classify N minerals (identified by numbers from 1 to N) into different classes. To assist with this, Dwarf the Worker has installed a new conveyor belt. The dwarves can add samples of minerals to the beginning of the belt and remove them from its end (i.e., remove the minerals that were added earliest).

Unfortunately, this conveyor belt has no knowledge base, so it cannot identify the class of any mineral on it. However, it has another useful feature: it displays *how many distinct classes of minerals are currently on the belt*. It turns out this is sufficient for the mine, as the dwarves do not need to know the class of each mineral: they just need to segregate the minerals into piles of the same class.

Can you help the dwarves solve this task using the conveyor belt?

Interaction Protocol

The classes of minerals **are fixed** at the beginning and do not depend on the queries made.

First, read a line containing one integer N, the number of minerals to classify. Then, you can perform operations of the following two types:

- `+` id — mineral with identifier id is added to the beginning of the conveyor belt,
- `-` — the mineral at the end of the conveyor belt is removed. If the belt is empty, nothing happens.

After each operation, the conveyor belt outputs one line containing the number of distinct classes currently on it.

To output the answer, first print a line containing `!` followed by k being the number of distinct classes in the mine. Then print k lines, each describing one class. Each line should contain a number c_i followed by c_i mineral identifiers of that class. Each of N minerals must appear in exactly one class. You may output the classes and minerals within each class in any order.

Remember to flush the output after each operation and after writing the answer. Remember to put spaces between numbers and the `+`, `-`, `!` symbols. You must read all data from the interactor.

Limits

$1 \le N \le 500$, you can perform at most 35 000 operations.

Sample interaction

The first sample test has $N = 3$ and the classes of minerals $1, 2, 3$ are $1, 2, 1$ respectively:

Input	Output	Status of the conveyor belt
3 1 2 2 2	 + 1 + 2 + 3 - ! 2 2 1 3 1 2	 [1] [1, 2] [1, 2, 1] [2, 1]

Local testing

In the *Files* section you can find **A.zip** containing sample tests and a grader `grader.cpp`. To test your solution, compile it, then pass the test name and your executable to the compiled `./grader`:

`./grader [test] [executable]`

For example: `./grader 0b.in ./abc`

The sample grader is **not guaranteed** to behave identically to the official one. However, neither is adaptive.