A. Bucket Sort

1 second, 256 megabytes

You are given an array with n integers. You are to apply *bucket sort* to sort this array. Given a pre-defined number of equally-sized buckets b that you need to use, your task is to output the contents of each bucket at the end of the process.

You are expected to divide the input range into b equal-sized buckets. It is guaranteed that this is possible, for this problem. For example if the input array is $[15, 13, 12, 10, 8, 8, 4]$ and $b = 4$, then the bucket ranges will be $[4, 6], [7, 9], [10, 12]$ and $[13, 15]$

Input
The first line contains two space-separated integers n $(1 \leq n \leq 10^5)$ — the size of the array, and b $(1 \leq b \leq 1000)$ — the number of equal-sized buckets you are to use. The second line contains n integers $a_1, a_2, \ldots, a_n$ $(-10^{12} \leq a_i \leq 10^{12})$ — the elements of the array.

Output
Print b lines — each one corresponding to the contents of the b buckets after sorting, respectively. Each of these b lines should contain space-separated integers, the first of which would be k — the number of integers placed in that bucket. k integers follow in the same line — the contents of the bucket, in sorted order.

input
7 4
15 13 12 10 8 8 4

output
1 4
2 8 8
2 10 12
2 13 15

input
10 1
17 5 0 -16 12 3 -1 19 12 20

output
10 -16 -1 0 3 5 12 12 17 19 20

input
27 20
15 25 14 -26 24 6 7 6 26 0 -9 18 0 -25 -25 -21 -11 -22 -27 -7 -1 21 20 13 0 24 32

output
4 -27 -26 -25 -25
1 -22
1 -21
0
0
1 -11
2 -9 -7
0
1 -1
3 0 0 0
0
3 6 6 7
0
2 13 14
1 15
2 18 20
1 21
4 24 24 25 26
0
1 32

Choose your data types wisely.

B. Nuts and Bolts

2 seconds, 256 megabytes

Steven has n nuts and n bolts. All nuts have different sizes from 1 to n, and all bolts have different sizes from 1 to n.

The operation Steven can perform is to try to compare one of the nuts with one of the bolts. As a result he will learn if the size of nut is greater, the size of bolt is greater or these nut and bolt match. He wants to find the matching nut for every bolt.

You must help Steven to do the plan, using no more than $5n \log_2 n$ operations.

Interaction
This is an interactive problem. Your program should communicate with the jury's program, using standard input and output for that.

At the beginning your program receives the integer n $(2 \leq n \leq 1000)$ — the number of nuts and bolts.

After that you can make no more than $5n \log_2 n$ queries. To do a query, output the character "?", and then numbers i and j ($1 \leq i \leq n, 1 \leq j \leq n$) — the number of nut and the number of bolt that Steven will try to compare.

As a result, you receive one of the three characters:

- "<", if the size of the i-th nut is less than the size of the j-th bolt,
- "=", if the i-th nut and the j-th bolt match,
- ">", if the size of the i-th nut is greater than the size of the j-th bolt.

As soon as you can say which nut matches which bolt, output the character "!", and then n distinct integers p_i $(1 \leq p_i \leq n)$, where p_i is the number of bolt which matches the nut with the number i. After that your program must terminate.

input
5
<
<
>
>
<
=
=
=
=
=

output

? 1 1

? 2 2

? 3 3

? 4 4

? 5 5

? 1 4

? 2 3

? 3 2

? 4 5

? 5 1

! 4 3 2 5 1

Please note that each your message must end with a line break. Also after outputting each message your program must flush the stream buffer, so that the outputted information could reach jury's program: for instance, this can be done by calling `fflush(stdout)` or `cout.flush()` in C++, `System.out.flush()` in Java, `Console.Out.Flush()` in C#, `flush(output)` in Pascal, `sys.stdout.flush()` in Python.

Emply lines in the sample are given only for convenience, to make it clear in which order the messages are written. When solving the problem you must not output empty lines and jury's program won't output empty lines too.