Twin Cookies

Problem name	**Twin Cookies**
Input file	standard input
Output file	standard output
Time limit	1 second
Memory limit	256 megabytes

This is an interactive problem. Your program will communicate with our grader by alternately writing messages to the standard output and reading messages from the standard input.

Sophie is preparing her twins' birthday party. The twins love cookies. For their birthday, they would like to try something new: cookies from the Unique Cookie Tastiness Company (UCTC).

Every cookie produced by UCTC has an integer tastiness value between 1 and 10^{16} inclusive. Since Sophie's twins get jealous of each other, each of them must receive cookies with the same sum of tastiness values.

UCTC only accepts orders of **exactly** n cookies. In each order the client specifies the tastiness of each of the n cookies they want.

Staying true to their name, the Unique Cookie Tastiness Company refuses to produce two cookies of the same tastiness for the same client. Sophie must make sure she never orders the same tastiness twice -- neither in the same order, nor in two different orders. Sophie has never purchased from UCTC before, so she can order each available tastiness once.

There is one more obstacle in Sophie's way: It is well-known that the delivery service of UCTC is awful. Whenever a client orders n cookies, only one of those n cookies actually reaches the client. The rest is eaten along the way by the employees of the delivery service. The client cannot influence which of the n ordered cookies will actually be delivered to her.

Since the birthday is approaching quickly, Sophie has the time to make at most 101 orders. Your task is to help her.

More specifically, you should do the following:

1. First, order the cookies. You can make at most 101 orders, each consisting of exactly n desired tastiness values. You make one order at a time. **Immediately after each order you are given the tastiness of the one cookie that you actually received.**

   Remember that you are not allowed to use the same tastiness value multiple times, even over multiple orders. (In particular, if you order a cookie with some tastiness t but it does not arrive, you **cannot** order a cookie with the same tastiness again.)

2. Then, divide the cookies. Once you have received enough cookies, you should distribute **some** of the received cookies between the twins. Both twins should receive at least one cookie, and each twin should receive cookies with the same total tastiness. **You do not have to use all the cookies you received!**

Output

Each time your program outputs one or more lines to the standard output, you must follow that action by **flushing the output stream**. This is necessary to make sure the data you output reaches the grader immediately.

Examples of how this can be done:

- In C++, there are multiple options.
  - `fflush(stdout);`
  - `std::cout << std::flush;`
  - `std::cout << std::endl;` (note that this also prints an extra newline)
  - reading from `std::cin` also flushes the output
- in Java, you can use `System.out.flush()`
- in Python, you can use `sys.stdout.flush()`

Interaction Protocol

Your program should perform the following sequence of actions:

1. Read the value n from the standard input.
2. At most 101 times:
   1. First, write one line describing an order of n cookies to the standard output.
   2. Then, read the tastiness of the cookie you received from the standard input. It is guaranteed that this value is among the n values that were in the current order.
3. Output three lines describing one valid way to give some of the cookies you received to the twins.

The grader will write each integer onto a separate line.

To order cookies, output a single line with `?` followed by n integers: the tastiness values of the cookies you want to order. Output a single space before each of the n integers.

Remember that you may make at most 101 orders and that you are not allowed to use the same tastiness value twice.

Once you have ordered enough cookies, output the final three lines that describe which cookies Sophie should give to the twins.

The first of these lines should have the form "`!` m k" with $m, k > 0$: the number of cookies the first and the second twin, respectively, should receive.

The second of these lines should contain m integers separated by single spaces: the tastiness values of the cookies the first twin should receive.

Similarly, the third line should contain k integers separated by single spaces: the tastiness values of the cookies the second twin should receive.

The output must satisfy the following conditions:

1. Each twin should receive at least one cookie.
2. Each twin should receive cookies with the same total tastiness.
3. Only cookies you actually received after your orders can be used.
4. Each of those cookies can only be given to at most one of the twins.

Any output that satisfies these conditions will be accepted. In particular, you can output the selected cookies in any order.

After you output the final three lines, flush the output stream one last time and then **terminate your program normally**.

Scoring

Subtask 1 (8 points): $n = 1$

Subtask 2 (9 points): $1 \leq n \leq 2$

Subtask 3 (18 points): $1 \leq n \leq 25$

Subtask 4 (16 points): $1 \leq n \leq 200$

Subtask 5 (13 points): $1 \leq n \leq 1000$

Subtask 6 (36 points): $1 \leq n \leq 5000$

Examples

standard input	standard output
1	? 13
13	? 7
7	? 31
31	? 12
12	? 5
5	? 3
3	! 2 3
	7 13
	12 5 3
2	? 3 7
7	? 2 8
2	? 1 5
5	! 2 1
	2 5
	7

Note

Examples of input and output should be read row by row. Your program alternately reads one value from standard input and writes one line (or three lines at the end) to standard output.

The grader chooses which cookie to return arbitrarily. This means that the grader might be adaptive to your queries in some tests, but it might also choose cookies at random in other tests. In particular, for $n = 2$, if you make the same sequence of orders as in the second example, you may receive a different set of cookies.