D. Finding lines

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

After some programming contest Roma decided to try himself in tourism. His home country Uzhlyandia is a Cartesian plane. He wants to walk along each of the Main Straight Lines in Uzhlyandia. It is known that each of these lines is a straight line parallel to one of the axes (i.e. it is described with the equation $x = a$ or $y = a$, where a is integer called the coordinate of this line).

Roma lost his own map, so he should find out the coordinates of all lines at first. Uncle Anton agreed to help him, using the following rules:

- Initially Roma doesn't know the number of vertical and horizontal lines and their coordinates;
- Roma can announce integer coordinates of some point in Uzhlandia, and Anton then will tell him the minimum among the distances from the chosen point to each of the lines. However, since the coordinates of the lines don't exceed 10^8 by absolute value, Roma can't choose a point with coordinates exceeding 10^8 by absolute value.

Uncle Anton is in a hurry to the UOI (Uzhlandian Olympiad in Informatics), so he can only answer no more than $3 \cdot 10^5$ questions.

The problem is that Roma doesn't know how to find out the coordinates of the lines. Write a program that plays Roma's role and finds the coordinates.

Input

There is no input initially. Your program should make queries to get information.

It is guaranteed that the number of horizontal and vertical lines is at least 1 and less than or equal to 10^4 for each type.

Interaction

To make a query, print a line "`0 x y`" ($-10^8 \le x, y \le 10^8$), where x and y are the coordinates of the point. After each query you need to print end-of-line, make "`flush`" operation, and then read the answer to the query — the minimum among the distances prom this point to the Main Straight Lines of Uzhlyandia.

You can do no more than $3 \cdot 10^5$ queries.

When you are ready to print the answer, print three lines:

1. In the first line print "`1 n m`", where n is the number of vertical lines (parallel to OY), and m is the number of horizontal lines (parallel to OX).
2. In the second line print n integers $x_1, x_2, ..., x_n$ — the coordinates of the vertical lines.
3. In the third line in the same format print m integers $y_1, y_2, ..., y_m$ — the coordinates of the horizontal lines.

You can print coordinates in arbitrary order.

To make "`flush`", you can use (just after printing a query/answer and end-of-line):

- `fflush(stdout)` in C++;
- `System.out.flush()` in Java;
- `stdout.flush()` in Python;
- `flush(output)` in Pascal;
- see the documentation for other languages.

You will get `Wrong Answer` if you make more queries than allowed or make an invalid query.

You can get `Idleness Limit Exceeded` if you don't print anything or if you forget to flush the output.

If at any moment your program reads `-1` as an answer, it should immediately exit normally (for example, by calling `exit(0)`). You will get `Wrong Answer` in this case, it means that you made more queries than allowed, or made an invalid query. If you ignore this, you can get other verdicts since your program will continue to read from a closed stream.

Making test for hacking

The first line should contain two integers n and m ($1 \le n, m \le 10^4$).

The second line should contain n distinct integers x_i ($-10^8 \le x_i \le 10^8$) — the coordinates of the vertical lines.

The third line should contain m distinct integers y_i ($-10^8 \le y_i \le 10^8$) — the coordinates of the horizontal lines.

You can write coordinates in arbitrary order.

You can see the example case in the notes.

Example

input
1 1 3 2

output
0 1 2 0 -2 -2 0 5 6 0 -2 2 1 1 2 2 0 -3

Note

The example test is

```
1 2
2
0 -3
```

The minimum distances are:

- from $(1, 2)$ to $x = 2$;
- from $(-2, -2)$ to $y = -3$;
- from $(5, 6)$ to $x = 2$;
- from $(-2, 2)$ to $y = 0$.