Problem H. Eager Sorting

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Petya is a sorting robot. In his memory, there is an array of length n (from 1 to 100), and its elements are pairwise distinct integers. The positions in the array are numbered left to right from 1 to n.

Nina is a robot operator. Nina wants to sort this array: make it so that, for each two distinct positions, the number on the left is less than the number on the right. The only available command for that is to compare elements at two positions, i and j. If the element on the left position (it can be position i or j) is greater than the element on the right position, Petya swaps them and displays number `1` on the screen. Otherwise, Petya does nothing with the array and just displays `0` on the screen.

Unfortunately, Petya's power system is damaged, so sometimes the robot shuts down. It looks as follows: when given a command, instead of executing it, Petya just displays `-1` on the screen, and then ignores any subsequent commands.

To make Petya work again, Nina disassembles him and then assembles anew. The array is not altered. Unfortunately, during repairs, Nina forgets which commands she already issued to the robot. After that, Petya works as long as his battery lasts, and then shuts down again.

The battery has enough power for Petya to execute 1500 commands. Petya shuts down exactly twice: after executing x-th command and after executing 1500-th command ($0 < x < 1500$, the value of x is not known to Nina). Knowing all the above, help Nina make it so that, after Petya shuts down for the second time, the array in his memory is sorted.

Interaction Protocol

In this problem, your solution will be run twice on each test. Your solution acts for Nina, and the jury program acts for Petya. Each line of input is terminated by an end-of-line character.

This is an interactive problem. Do not forget to flush the output immediately after printing each command!

Below we describe the interaction of your solution and the jury program: it is the same during both runs.

The first line of input contains an integer n, the size of the array ($1 \le n \le 100$). After that, your solution issues commands, and the jury program executes them and prints the answers.

To issue a command "compare elements at positions i and j", print a line of the form "i j" where $1 \le i, j \le n$. As a result, you will get a line with a single integer:

- `1` means that Petya swapped elements at positions i and j because the left one was greater than the right one;
- `0` means that Petya did not change anything because the left element was not greater than the right one (that is, either the left one was less than the right one, or $i = j$);
- `-1` means that Petya shut down instead of executing the command.

In the latter case, the solution must terminate. In other cases, another command can be issued.

If you want to terminate the interaction before Petya shuts down, instead of a command, print the line "`-1 -1`". After that, the solution must terminate.

In each test, the array before the first run is fixed in advance, and the array before the second run is in the state it was at the end of the first run. Additionally, in each test, the number of commands x after which Petya shuts down for the first time ($0 < x < 1500$) is also fixed in advance. During the second run, the robot will shut down after $1500 - x$ commands, even if the interaction during the first run ended before he shut down for the first time.

Example

Below we show two runs of a certain solution on the first test. Empty lines are added only for readers' convenience: there will be no empty lines during a real run.

standard input	*standard output*	*array*
5		4 2 5 1 3
	1 5	
1		3 2 5 1 4
	1 2	
1		2 3 5 1 4
	2 3	
0		2 3 5 1 4
	3 4	
1		2 3 1 5 4
	4 5	
1		2 3 1 4 5
	2 1	
0		2 3 1 4 5
	3 2	
1		2 1 3 4 5
	4 3	
0		2 1 3 4 5
	1 2	
-1		

standard input	*standard output*	*array*
5		2 1 3 4 5
	1 2	
1		1 2 3 4 5
	2 3	
0		1 2 3 4 5
	1 2	
0		1 2 3 4 5
	-1 -1	