COI '17 #4 Zagonetka

Time limit: 3.0s **Memory limit:** 1G

Mislav and Marin learned about permutations in their combinatorics class, and they invented an interesting game where the player must guess certain permutations that meet certain conditions. A *permutation* of order n is an array of numbers $p = (p_1, p_2, \ldots, p_n)$ where each number between 1 and n appears exactly once. A *condition* is a pair of distinct numbers (a, b), both between 1 and n, inclusive. A permutation p *meets* condition (a, b) if $p_a < p_b$.

The game is played as follows. Marin first chooses zero or more conditions and one permutation p of order n that meets all of them. In the beginning of the game, Marin texts Mislav only the chosen permutation p (the conditions remain secret). Mislav's goal is to determine the lexicographically smallest and lexicographically largest permutation that meets all of Marin's conditions. In each step of the game, Mislav chooses one permutation q of order n and texts it to Marin. Marin then reveals if that permutation q met all of his secret conditions.

This is an interactive task. Write a program that will play the game instead of Mislav. Your program must, for a given permutation p (of length at most 100) that meets the secret conditions, in at most $5\,000$ steps find the lexicographically smallest and the lexicographically largest permutation that meet all the conditions.

Constraints

Subtask	Points	Constraints
1	9	$2 \le n \le 6$
2	18	$30 \le n \le 70$, Marin chose exactly 1 condition.
3	22	$10 \le n \le 30$
4	51	$70 < n \le 100$

Interaction

Before the interaction, your program must read from the standard input the following: The first line of input contains the integer n — the size of all permutations in the game. The following line contains n distinct integers $p_1, p_2, \ldots, p_n$ $(1 \le p_j \le n)$ — the permutation p. You can assume that p meets all of Marin's conditions.

After this, your program can send Marin *queries* by writing to the standard output. Each query must be printed in its own line in the form of `query q1 q2 ... qn` where $q_1, q_2, \ldots, q_n$ are distinct integers between 1 and n, inclusively. After each printed query, your program must *flush* the output and read from the standard input Marin's *reply* — the number 1 if the permutation q from the query meets all the conditions, and 0 otherwise.

When your program has found a solution, it must output a line to the standard output containing the command `end`, then a line of the form of `a1 a2 ... an` containing the required lexicographically smallest permutation and a line of the form of `b1 b2 ... bn` containing the required lexicographically largest permutation. In the end, the program must *flush* the output again and terminate the execution.

Sample Interaction

In the following interaction sample, the first column contains the data that your program outputs to the standard output, and the second column contains the data that your program reads from the standard input. After three steps of the game, i.e. after three queries, the program determines the correct solution.

`>>>` denotes your output. Do not print this out.

The chosen secret conditions are $(2, 1)$ and $(3, 4)$.

```
4
3 2 1 4
>>> query 2 3 1 4
0
>>> query 3 2 4 1
0
>>> query 4 1 2 3
1
>>> end
>>> 2 1 3 4
>>> 4 3 1 2
```

Explanation

For the first query, condition $(2, 1)$ is not met.

For the second query, condition $(3, 4)$ is not met.

For the third and final query, both conditions are met.