

B. New Year Cards

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

As meticulous Gerald sets the table, Alexander finished another post on Codeforces and begins to respond to New Year greetings from friends. Alexander has n friends, and each of them sends to Alexander exactly one e-card. Let us number his friends by numbers from 1 to n in the order in which they send the cards. Let's introduce the same numbering for the cards, that is, according to the numbering the i -th friend sent to Alexander a card number i .

Alexander also sends cards to friends, but he doesn't look for the new cards on the Net. He simply uses the cards previously sent to him (sometimes, however, he does need to add some crucial details). Initially Alexander doesn't have any cards. Alexander always follows the two rules:

1. He will never send to a friend a card that this friend has sent to him.
2. Among the other cards available to him at the moment, Alexander always chooses one that Alexander himself likes most.

Alexander plans to send to each friend exactly one card. Of course, Alexander can send the same card multiple times.

Alexander and each his friend has the list of preferences, which is a permutation of integers from 1 to n . The first number in the list is the number of the favorite card, the second number shows the second favorite, and so on, the last number shows the least favorite card.

Your task is to find a schedule of sending cards for Alexander. Determine at which moments of time Alexander must send cards to his friends, to please each of them as much as possible. In other words, so that as a result of applying two Alexander's rules, each friend receives the card that is preferred for him as much as possible.

Note that Alexander doesn't choose freely what card to send, but he always strictly follows the two rules.

Input

The first line contains an integer n ($2 \leq n \leq 300$) — the number of Alexander's friends, equal to the number of cards. Next n lines contain his friends' preference lists. Each list consists of n different integers from 1 to n . The last line contains Alexander's preference list in the same format.

Output

Print n space-separated numbers: the i -th number should be the number of the friend, whose card Alexander receives right before he should send a card to the i -th friend. If there are several solutions, print any of them.

Examples

input
4 1 2 3 4 4 1 3 2 4 3 1 2 3 4 2 1 3 1 2 4
output
2 1 1 4

Note

In the sample, the algorithm of actions Alexander and his friends perform is as follows:

1. Alexander receives card 1 from the first friend.

2. Alexander sends the card he has received (at the moment he only has one card, and therefore it is the most preferable for him) to friends with the numbers 2 and 3.
3. Alexander receives card 2 from the second friend, now he has two cards — 1 and 2.
4. Alexander sends a card to the first friend. Despite the fact that Alexander likes card 1 more, he sends card 2 as he cannot send a friend the card sent by that very friend.
5. Alexander receives card 3 from the third friend.
6. Alexander receives card 4 from the fourth friend.
7. Among the cards Alexander has number 3 is his favorite and he sends it to the fourth friend.

Note that Alexander can send cards to multiple friends at a time (in this case the second and the third one). Alexander can send card 3 to the fourth friend after he receives the third card or after he receives the fourth card (both variants are correct).