

## Problem D. Lost in Transfer

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

Dima has a set of  $n$  numbers. Dima wants to transmit this set to Katya. He takes numbers from the set, one by one, in any order he pleases, and enters them into the transmitter.

Katya receives numbers from the receiver in the order in which Dima enters them. However, the transmission channel is not ideal, so one of the numbers could have been lost in transfer. Nevertheless, it is very important for Katya to exactly reconstruct the set that Dima wanted to transmit.

Help Dima and Katya agree in advance how they transmit numbers so that Katya can always reconstruct Dima's set, even if one of its elements was lost in transfer.

### Interaction Protocol

In this problem, your solution will be run twice on each test. Each test consists of separate test cases. Both in input and in output, adjacent numbers on a line are separated by spaces.

During the first run, the solution transmits sets as Dima. The first line contains the word **"transmit"**. The second line contains an integer  $t$ , the number of test cases ( $1 \leq t \leq 1000$ ). Each of the next  $t$  lines describes a single test case. Such line starts with an integer  $n$ , the number of elements in the set ( $20 \leq n \leq 100$ ). Then follow  $n$  pairwise distinct integers  $a_1, a_2, \dots, a_n$ , the elements of the set ( $1 \leq a_i \leq 500$ ).

Print  $t$  lines, one for each test case. On each line, print the respective integers  $a_1, a_2, \dots, a_n$ , each of them exactly once, in any order you like.

During the second run, the solution reconstructs sets as Katya. The first line contains the word **"recover"**. The second line contains an integer  $t$ , the number of test cases, same as during the first run ( $1 \leq t \leq 1000$ ). Each of the next  $t$  lines describes a single test case. Such line starts with an integer  $m$ , the number of integers received by Katya ( $19 \leq m \leq 100$ ). Then follow  $m$  pairwise distinct integers  $b_1, b_2, \dots, b_m$ , the integers received by Katya themselves. These are the integers sent by Dima during the first run, given in the order of transmission. However, one of the integers might be omitted (and then  $m$  is one less than the respective  $n$  during the first run).

Print  $t$  lines, one for each test case. On each line, print the integers  $a_1, a_2, \dots, a_n$  from the respective test case, each of them exactly once, in any order you like.

### Note

In this problem, the tests are generated using a pseudorandom number generator. In each test, the number of test cases  $t$  and the size of the set  $n$  in each test case are chosen in advance. After that, each set of size  $n$  is selected randomly with equal probability among all possible sets of size  $n$  consisting of integers from 1 to 500. The elements of the set are given in random order.

Additionally, in each test case, it is decided in advance which number will be lost in transfer. For a set of size  $n$ , a position  $p$  is selected randomly with equal probability among all possible integers from 1 to  $n + 1$ . If  $p \leq n$ , it means that  $p$ -th printed number will be lost. When  $p = n + 1$ , every number will be transmitted successfully.

## Example

On each test, the input during the second run depends on the solution's output during the first run. Two runs of some solution on the first test are shown below.

standard input
transmit 2 20 97 388 459 467 32 99 98 296 403 325 330 271 87 333 378 267 405 58 426 374 20 125 481 451 150 495 136 444 192 118 26 68 281 120 61 494 339 86 292 100 32
standard output
405 97 87 58 374 98 271 296 330 267 99 32 378 333 325 467 388 403 459 426 494 68 481 61 120 125 281 444 150 86 339 26 32 118 451 136 495 100 292 192

standard input
recover 2 19 97 87 58 374 98 271 296 330 267 99 32 378 333 325 467 388 403 459 426 20 494 68 481 61 120 125 281 444 150 86 339 26 32 118 451 136 495 100 292 192
standard output
97 87 58 374 98 271 296 330 267 99 32 378 333 325 467 388 403 459 426 405 494 68 481 61 120 125 281 444 150 86 339 26 32 118 451 136 495 100 292 192