

## Задача D. Утеряно при передаче

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

У Димы есть множество из  $n$  чисел. Дима хочет передать это множество Кате. Он берёт числа из множества — по одному, в любом удобном ему порядке — и вводит их в передатчик.

Катя получает числа из приёмника — в том порядке, в котором их ввёл Дима. Но канал передачи не идеален, и одно из чисел могло потеряться. Тем не менее, очень важно, чтобы Катя смогла построить ровно то множество, которое хотел передать Дима.

Помогите Диме и Кате заранее договориться, как передавать числа, чтобы Катя всегда могла восстановить Димино множество, даже если одно из чисел потерялось при передаче.

### Протокол взаимодействия

В этой задаче ваше решение будет запущено на каждом тесте два раза. Каждый тест состоит из отдельных тестовых случаев. При вводе и при выводе соседние числа в строке разделяются пробелами.

При первом запуске решение посылает множества за Диму. В первой строке записано слово «**transmit**». Вторая строка содержит целое число  $t$  — количество тестовых случаев ( $1 \leq t \leq 1000$ ). Каждая из следующих  $t$  строк описывает один тестовый случай. Такая строка начинается с целого числа  $n$  — количества чисел во множестве ( $20 \leq n \leq 100$ ). Далее следуют  $n$  попарно различных целых чисел  $a_1, a_2, \dots, a_n$  — элементы множества ( $1 \leq a_i \leq 500$ ).

Выведите  $t$  строк, по одной на каждый тестовый случай. В каждой строке выведите соответствующие числа  $a_1, a_2, \dots, a_n$ , каждое по одному разу, в любом желаемом порядке.

При втором запуске решение восстанавливает множества за Катю. В первой строке записано слово «**recover**». Вторая строка содержит целое число  $t$ , то же, что и при первом запуске — количество тестовых случаев ( $1 \leq t \leq 1000$ ). Каждая из следующих  $t$  строк описывает один тестовый случай. Такая строка начинается с целого числа  $m$  — количества чисел, полученных Катей ( $19 \leq m \leq 100$ ). Далее следуют  $m$  попарно различных целых чисел  $b_1, b_2, \dots, b_m$  — сами числа, полученные Катей. Это те числа, которые при первом запуске передал Дима, в порядке передачи — но, возможно, одно из чисел пропущено (и тогда число  $m$  на единицу меньше соответствующего числа  $n$  при первом запуске).

Выведите  $t$  строк, по одной на каждый тестовый случай. В каждой строке выведите исходные числа  $a_1, a_2, \dots, a_n$  из этого тестового случая, каждое по одному разу, в любом желаемом порядке.

### Замечание

Тесты в этой задаче сгенерированы генератором псевдослучайных чисел. В каждом тесте выбраны количество тестовых случаев  $t$  и размер множества  $n$  в каждом из тестовых случаев. После этого каждое множество размера  $n$  выбрано случайно и равномерно из всех возможных множеств размера  $n$ , состоящих из целых чисел от 1 до 500. Элементы множества даны в случайном порядке.

Кроме того, в каждом тестовом случае заранее зафиксировано, какое из передаваемых чисел потеряется при передаче. Для множества размера  $n$  позиция  $p$  выбрана случайно и равномерно из всех целых чисел от 1 до  $n + 1$ . Если  $p \leq n$ , это означает, что будет потеряно  $p$ -е по счёту из переданных чисел. При  $p = n + 1$  никакое число не теряется.

## Пример

На каждом тесте входные данные при втором запуске зависят от того, что вывело решение при первом запуске. Далее показаны два запуска какого-то решения на первом тесте.

стандартный ввод
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
стандартный вывод
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
стандартный ввод
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
стандартный вывод
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