

## Problem B. Even and Odd Combinations

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

Let a *k*-combination out of *n* be a *k*-element subset of the *n*-element set  $\{1, 2, \dots, n\}$ . To denote a combination, list its elements in ascending order. For example, 2-combinations out of 3 look as follows:  $\{1, 2\}$ ,  $\{1, 3\}$ ,  $\{2, 3\}$ .

Let a combination be *even* if the number of its elements is an even number, and *odd* otherwise. For a fixed  $n > 0$ , consider two sets:  $A_n$ , the set of all even combinations out of *n*, and  $B_n$ , the set of all odd combinations out of *n*. It can be shown that  $A_n$  and  $B_n$  contain the same number of combinations.

For each  $n = 1, 2, \dots, 50$ , your task is as follows. Construct any bijection (a one-to-one correspondence) between the sets  $A_n$  and  $B_n$ . After that, given an element of one of these sets, print the corresponding element of the other set.

### Interaction Protocol

To check that you indeed constructed a bijection, in this problem, your solution will be run twice on each test. Each line of input is terminated by an end-of-line character.

### First Run

During the first run, the first line contains an integer *t*, the number of test cases ( $1 \leq t \leq 1000$ ). Then follow their descriptions.

Each test case denotes a combination and is given on two input lines. The first of these lines contains two space-separated integers *n* and *k* ( $1 \leq n \leq 50$ ,  $0 \leq k \leq n$ ). The second one contains *k* space-separated integers  $a_1, a_2, \dots, a_k$ , the elements of the combination ( $1 \leq a_1 < a_2 < \dots < a_k \leq n$ ). If  $k = 0$ , the second line is empty.

For each test case, print the corresponding combination from the other set. Output format for combinations is exactly the same as input format. The printed combination must have the same *n* as the given one, and *k* must have different parity. There are no other restrictions on the correspondence.

### Second Run

During the second run, the input format is exactly the same as during the first run. However, in each test case, the given combination is not the initial one, but instead the one printed during the first run.

For each test case, as during the first run, print the corresponding combination from the other set. Output format for combinations is exactly the same as input format. As the correspondence must be a bijection, the combination printed during the second run must be the same as the one given during the first run. This is what the jury program will check.

### Example

For each test, the input during the second run depends on the solution's output during the first run.

Below we show two runs of a certain solution on the first test.

<i>standard input</i>	<i>standard output</i>
6	3 3
3 0	1 2 3
	2 2
2 1	1 2
1	3 0
3 3	
1 2 3	3 2
3 1	2 3
1	3 2
3 1	1 3
2	3 2
3 1	1 2
3	

<i>standard input</i>	<i>standard output</i>
6	3 0
3 3	
1 2 3	2 1
2 2	1
1 2	3 3
3 0	1 2 3
	3 1
3 2	1
2 3	3 1
3 2	2
1 3	3 1
3 2	3
1 2	