

## Problem A. Brackets

Input file:            `brackets.in`  
Output file:          `brackets.out`  
Time limit:           3 seconds  
Memory limit:        256 megabytes

Regular brackets sequence is the sequence of  $2n$  characters each of which is either “(” (opening bracket), or “)” (closing bracket), such that each prefix contains no more closing brackets than opening ones, and the whole sequence contains  $n$  opening and  $n$  closing brackets. For example, “”, “((()))”, “()()()()”, “(())()” are the regular brackets sequences, but “())(())” is not, because its prefix “())” contains more closing than opening brackets, neither is “((", because it contains more opening brackets than closing.

For each opening bracket you can find the *corresponding* closing bracket — the one following it, such that there is a regular brackets sequence between them.

The generalization of the regular brackets sequence is the regular brackets sequence with  $k$  types of brackets. Let each bracket have its *type* — an integer number from 1 to  $k$ . The sequence of opening and closing brackets with types is regular if it is a regular brackets sequence and the corresponding closing bracket for each opening bracket has the same type as the opening bracket itself. For example, “ $(_1(_2)_2)_1(_1)_1$ ” is the regular brackets sequence with two types of brackets, but “ $(_1(_2)_1)_2(_1)_1$ ” is not.

If you introduce some order on  $2k$  typed brackets, you can consider *lexicographical* order on all regular brackets sequences of length  $2n$ . You are given  $n, k$ , the order on typed brackets and the regular brackets sequence. Find the next regular brackets sequence in the lexicographical order.

### Input

The first line of the input file contains  $t$  — the number of tests ( $1 \leq t \leq 10000$ ). The description of  $t$  test cases follows.

The first line of each description contains two integer numbers:  $n$  and  $k$  ( $1 \leq n \leq 100\,000$ ,  $1 \leq k \leq 10\,000$ ). The second line contains  $2k$  integer numbers — the permutation of the set  $\{-k, -(k-1), \dots, -2, -1, 1, 2, \dots, k\}$ . It lists the brackets from the least to the greatest in the given order. Positive number  $i$  means the opening bracket of the  $i$ -th type, negative number  $-i$  means the closing bracket of the  $i$ -th type. The third line contains  $2n$  integer numbers from  $-k$  to  $k$  (except 0) and describes the regular brackets sequence.

The sum of  $n$  and the sum of  $k$  in all test cases don't exceed 100 000.

### Output

For each test case output  $2n$  integer numbers — the next regular brackets sequence after the one from the input file in lexicographical order. If the sequence in the input file is the last one, output the first one.

### Example

brackets.in	brackets.out
2	1 2 -2 -1 2 -2
3 2	1 2 -2 2 -2 -1
1 -1 2 -2	
1 2 -2 -1 1 -1	
3 2	
1 -1 2 -2	
1 2 -2 -1 2 -2	