

Problem G. Records and Cycles

Input file: `records-and-cycles.in`
Output file: `records-and-cycles.out`
Time limit: 2 seconds
Memory limit: 256 mebibytes

In this problem, you have to construct a bijection which maps every permutation of length n with k cycles to a permutation of length n with k records.

A *permutation* p of length n is a sequence of n integers $p(1), p(2), \dots, p(n)$ such that each of the numbers $1, 2, \dots, n$ occurs in that sequence exactly once.

A *cycle* in a permutation p is a sequence of integers c_1, c_2, \dots, c_m such that $p(c_1) = c_2, p(c_2) = c_3, \dots, p(c_m) = c_1$. A special case is a cycle of unit length: $p(i) = i$. It can be shown that each permutation can be represented as a collection of pairwise disjoint cycles.

A *record*, or record value, in a permutation p is its element p_j which is greater than all elements of the permutation preceding it: $p_j > p_i$ for each $i < j$. Note that p_1 is always a record by definition.

One can prove that for any given n and k , the number of permutations of length n with exactly k cycles is equal to the number of permutations of length n with exactly k records. Therefore, there exists a bijection which maps permutations of length n with k cycles to permutations of length n with k records and vice versa. Recall that a *bijection* is a function giving an exact pairing of the elements of two sets: every element of one set is paired with exactly one element from the other set, and every element of the other set is paired with exactly one element of the first set. Your task is to find and implement such a function.

Input

The first line of input contains an integer t , the number of permutations ($1 \leq t \leq 300\,000$).

Each of the next t lines contains description of a single permutation. A description starts with an integer n , the length of the permutation ($1 \leq n \leq 300\,000$), followed by a character which is either “**r**” or “**c**” and denotes the type of the permutation. Then follows a sequence of n integers which constitutes the permutation itself. Every integer from 1 to n occurs in this sequence exactly once.

Consecutive tokens on a line are separated by single spaces. The total length of all permutations in the input does not exceed 300 000.

Output

The output must be in the same format as the input.

The first line of output must contain the integer t , the number of permutations in the input.

After that, for each of the t given permutations, output the permutation corresponding to it on a separate line. If the type of the input permutation is “**r**”, find the number of records in it and output the corresponding permutation of type “**c**” with the same number of cycles. If the type of the input permutation is “**c**”, find the number of cycles in it and output the corresponding permutation of type “**r**” with the same number of records.

Each of the t lines must start with the integer n which is the length of the permutation, followed by the character denoting the type. After that must follow n integers which constitute the permutation itself. Separate consecutive tokens on a line by single spaces.

The correspondence must be consistent: if a permutation p of type “r” corresponds to a permutation q of type “c”, the inverse correspondence must also be true, no other permutation of type “c” can correspond to p , and no other permutation of type “r” can correspond to q . Note however that your bijection does not need to be consistent across different outputs.

Example

records-and-cycles.in	records-and-cycles.out	alternative output
6	6	6
3 r 1 2 3	3 c 1 2 3	3 c 1 2 3
3 c 1 2 3	3 r 1 2 3	3 r 1 2 3
3 r 2 1 3	3 c 2 1 3	3 c 3 2 1
3 r 2 3 1	3 c 3 2 1	3 c 1 3 2
3 c 2 1 3	3 r 2 1 3	3 r 1 3 2
2 c 2 1	2 r 2 1	2 r 2 1

Explanation

In this example, the first five permutations have length $n = 3$. The first permutation has type “r” and contains three records. The second permutation has type “c” and contains three cycles. These two permutations correspond to each other. The third and the fourth permutations have type “r” and contain two records each. The fifth permutation has type “c” and contains two cycles. It corresponds to the third permutation. The fourth permutation corresponds to the permutation 3, 2, 1 of type “c” which is another permutation with two cycles.

The last permutation has length $n = 2$, type “c” and contains one cycle. It corresponds to a permutation 2, 1 of type “r” with one record.

Note that this is not the only possible bijection: one alternative output is given above. Here, for the third permutation in the input (which is 2, 1, 3 of type “r”), we pick 3, 2, 1 of type “c” as the corresponding permutation. Then, the fourth permutation (2, 3, 1 of type “r”) must get a different corresponding permutation, too, and we take the permutation 1, 3, 2 of type “c”. After that, for the fifth permutation (2, 1, 3 of type “c”), the only possible option is to pick 1, 3, 2 of type “r” since the other two permutations with two records already correspond to different permutations with two cycles.