Problem K. Trijection

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

There are many types of combinatorial objects such that the number of distinct objects of a given size n is a Catalan number $C_n = \frac{(2n)!}{n!(n+1)!}$. Here are the first few Catalan numbers: $C_0 = 1$, $C_1 = 1$, $C_2 = 2$, $C_3 = 5$, $C_4 = 14$, $C_5 = 42$, ... Consider three such types of objects:

- Skew polyominoes with perimeter $2n+2$. These are collections of squares on a rectangular board of $h \times w$ squares where $h+w = n+1$. A polyomino must be side-connected. The bottom left square and the top right square are occupied by the polyomino. Other squares can be either free or occupied, but the following conditions must hold:
  - in each row and in each column, the occupied squares form a continuous segment;
  - each column segment starts higher or at the same level as the start of its left neighbor;
  - each column segment ends higher or at the same level as the end of its left neighbor;
  - each row segment starts to the right or at the same point as the start of its lower neighbor.
  - each row segment ends to the right or at the same point as the end of its lower neighbor.

  Here are all skew polyominoes of size $n = 4$:

```
poly    poly    poly    poly    poly    poly    poly
4 1     3 2     3 2     3 2     3 2     3 2     3 2
#       .#      .#      ##      ##      ##      .#
#       .#      ##      #.      ##      ##      ##
#       ##      #.      #.      #.      ##      ##
#

        poly    poly    poly    poly    poly    poly
poly    2 3     2 3     2 3     2 3     2 3     2 3
1 4     ..#     .##     ###     ###     ###     .##
####    ###     ##.     ##.     #..     ###     ###
```

- 321-avoiding permutations of length n. These are permutations $p_1, p_2, \ldots, p_n$ of elements $1, 2, \ldots, n$ which don't contain a triple of positions $i < j < k$ such that $p_i > p_j > p_k$.

  Here are all 321-avoiding permutations of size $n = 4$:

```
perm      perm      perm      perm      perm      perm      perm
1 2 3 4   1 3 2 4   1 4 2 3   2 1 4 3   2 3 4 1   3 1 2 4   3 4 1 2

perm      perm      perm      perm      perm      perm      perm
1 2 4 3   1 3 4 2   2 1 3 4   2 3 1 4   2 4 1 3   3 1 4 2   4 1 2 3
```

- Triangulations of a convex $(n+2)$-gon. Label the vertices of the polygon by integers from 1 to $n+2$ in the order of traversal. In each of the n triangles, arrange the vertex numbers in ascending order. Next, arrange the triangles themselves in ascending order as triples of integers.

  Here are all triangulations of size $n = 4$:

```
triang    triang    triang    triang    triang    triang    triang
1 2 5     1 2 3     1 2 3     1 2 6     1 2 4     1 2 3     1 2 4
1 5 6     1 3 4     1 3 6     2 3 4     1 4 5     1 3 6     1 4 6
2 3 5     1 4 6     3 4 5     2 4 5     1 5 6     3 4 6     2 3 4
3 4 5     4 5 6     3 5 6     2 5 6     2 3 4     4 5 6     4 5 6

triang    triang    triang    triang    triang    triang    triang
1 2 3     1 2 6     1 2 3     1 2 6     1 2 6     1 2 6     1 2 5
1 3 5     2 3 6     1 3 4     2 3 4     2 3 6     2 3 5     1 5 6
1 5 6     3 4 5     1 4 5     2 4 6     3 4 6     2 5 6     2 3 4
3 4 5     3 5 6     1 5 6     4 5 6     4 5 6     3 4 5     2 4 5
```

Let us fix the number n and consider three sets:

- A_n, the set of skew polyominoes of size n,
- B_n, the set of 321-avoiding permutations of size n,
- C_n, the set of triangulations of size n.

Construct a *trijection* between these sets. A trijection is like a bijection, but there are three sets instead of two. Formally, invent a function $f_n : A_n \cup B_n \cup C_n \to A_n \cup B_n \cup C_n$ such that its value for each object of any type is necessarily an object of a **different** type, and additionally, $f_n(f_n(x)) = x$ for every x (in other words, f_n is an inverse to itself).

After that, for given objects of the three types, print the results of trijection.

Additional constraint: the given number n **can not** have the form $2^k - 1$.

Interaction Protocol

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 two space-separated integers n and t: the size, which is the same for all objects, and the number of objects ($2 \le n \le 35$, $1 \le t \le 1000$, the number n can not have the form $2^k - 1$). Then t objects are given. Each object description starts with a line indicating its type, followed by one or more lines describing the object itself, which depend on the type. The detailed desription of types and the formatting rules for objects are shown in the statement above and in the example below.

On the first line, print n and t separated by a space (this line is part of the format for convenience, so that it is possible to use solution's output as input without changes). After that, print t objects: the results of the trijection for the t given objects. The output format for the objects is the same as the input format.

Second Run

During the second run, the input and output formats are the same as during the first run. However, instead of the t initial objects, the objects given in the input are the ones printed during the first run, **reordered randomly**.

The t initial objects are fixed in advance in each test. The random permutation applied between the first and the second run is also fixed in advance.

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. Note how the output of the second run is reordered input of the first run.

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>5 4
poly
4 2
.#
##
##
#.
perm
4 1 5 2 3
triang
1 2 4
1 4 5
1 5 7
2 3 4
5 6 7
perm
2 1 3 5 4</pre></td><td><pre>5 4
perm
3 1 4 2 5
poly
4 2
##
##
##
#.
poly
3 3
.##
###
##.
triang
1 2 3
1 3 7
3 4 7
4 5 7
5 6 7</pre></td></tr>
</table>

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>5 4
poly
4 2
##
##
##
#.
triang
1 2 3
1 3 7
3 4 7
4 5 7
5 6 7
poly
3 3
.##
###
##.
perm
3 1 4 2 5</pre></td><td><pre>5 4
perm
4 1 5 2 3
perm
2 1 3 5 4
triang
1 2 4
1 4 5
1 5 7
2 3 4
5 6 7
poly
4 2
.#
##
##
#.</pre></td></tr>
</table>