

Problem G. Good and Lucky Matrices

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 1024 mebibytes

This is a run-twice problem: your solution will be executed twice on each test. See the rest of the statement and the input format section for more details.

A square binary matrix is *good* if and only if it has an odd determinant.

A binary matrix of size $n \times n$ is *lucky* if and only if the greedy matching algorithm described below successfully finds a matching of size n . The algorithm reads the matrix line-by-line, from top to bottom. On each line, if there is a 1 in a column that was never chosen before, the algorithm greedily chooses the leftmost such column.

Your task is to compare numbers of good and lucky matrices of size $n \times n$. In order to do that, you have to produce a bijection between the smaller set and an equinumerous subset of the larger set.

Input

On each run, the first line contains an integer t : the number of test cases. The test cases follow.

The first line of each test case contains a string “good” or “lucky” denoting the type of matrix.

The next line contains a single integer n ($1 \leq n \leq 2000$): the size of the matrix.

Then follow n lines: i -th of them contains a binary string of length exactly n denoting the i -th row of the matrix.

The matrices given to your program on the second run will be the exactly the same as the matrices printed by your program on the first run, in the same order.

It is guaranteed that the sum of n over all test cases does not exceed 2000.

Output

On each run, you have to print t answers. Each answer should be either a single integer -1 printed on a separate line or a matrix of the opposite type formatted similarly to the input: a line with size n followed by n lines describing the rows.

The mapping produced by your program should be a bijection. In other words, all the matrices printed on the second run should correspond to preimage of the corresponding matrix on the first run. Additionally, the number of -1 should be not greater than the difference in the sizes of the sets.

Formally, if on the first run your answer for matrix A of some type was matrix B , and on the second run your answer for matrix B of the opposite type was matrix C , then $A = C$ must hold. Additionally, for each n and each matrix type, let U be the set of all $n \times n$ matrices of this type, and V the set of $n \times n$ matrices of the other type. Then you can answer -1 for at most $\max(|U| - |V|, 0)$ different matrices from U .

Examples

<i>standard input</i>	<i>standard output</i>
3 lucky 2 11 11 good 2 11 01 lucky 2 01 10	2 11 10 2 11 01 2 01 10
3 good 2 11 10 lucky 2 11 01 good 2 01 10	2 11 11 2 11 01 2 01 10