Problem G
Guarded Rook Combinations

Chess is a strategic board game that has been played for centuries. It involves two players taking turns moving their pieces on a square grid board. The objective of the game is to capture the opponent's king in a position where it cannot escape, known as "checkmate". One of the six different types of chess pieces is the rook. The rook is a powerful piece that can move horizontally or vertically any number of squares, as long as there are no obstacles in its path. It is usually represented by a structure resembling a tower and is considered one of the most valuable pieces in the game due to its ability to control multiple lines and contribute to various tactics and strategies.

As a passionate chess enthusiast, MofK always spends time to study chess pieces, positions, playing strategies, and then provides his own evaluations and comments to use in chess matches with Grandmasters in major tournaments. Today, MofK decides to dedicate an entire day to studying the properties of rooks. MofK has many chess boards and chess sets, so he decides to conduct the study with an $n \times n$ board instead of an 8×8 chesisboard. Each square on the chessboard can have at most one rook placed on it. Initially, MofK has placed some rooks on certain squares of his board. The more rooks he places, the more pairs of rooks that *guard* each other he sees in the chess position. To improve his predictive ability, MofK wants to add **exactly** k more rooks to the board, such that there are **exactly** p pairs of *guarding* rooks.

Two different rooks on the board are considered *guarding* each other if:

- the two rooks are in the same row or the same column on the board, and
- there is no other rook between the two rooks.

Given the initial board, the numbers k and p, help MofK find a way to add **exactly** k rooks to the chessboard such that there are **exactly** p pairs of *guarding* rooks, or indicate that there is no way to achieve that.

Input

The input contains multiple test cases. Each test case is presented as below:

- The first line contains three integers n, k and p ($1 \le n \le 10^3, 0 \le k \le 10^6, 0 \le p \le 10^9$) — the size of the board, the number of rooks to be added, and the number of pairs of *guarding* rooks that should appear on the board, respectively.
- The last n lines describe the initial board, each contains a string of length n. Each character of each string is either `"R"` (which represents a cell with a rook) or `"."` (which represents a cell without any rooks).

The input is terminated by a line containing three 0s which does not represent a test case. It is guaranteed that the sum of n^2 in all test cases does not exceed 2000^2.

Output

For each test case, if there is no way to add **exactly** k rooks to the board such that there are **exactly** p pairs of *guarding* rooks, print a single line `"NO"` (without quotes).

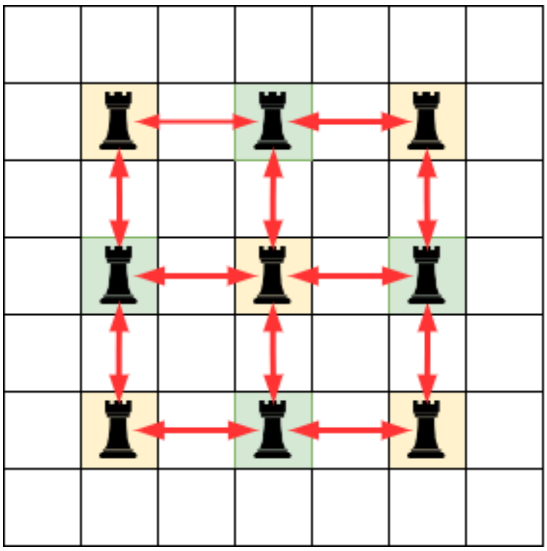
Otherwise, print "YES" (without quotes) on the first line. The next n lines describe the board after adding rooks in the same format as the input. Each line contains a string of length n where each character is either "R" or ".". "R" represents a cell with a rook, while "." represents a cell without any rooks.

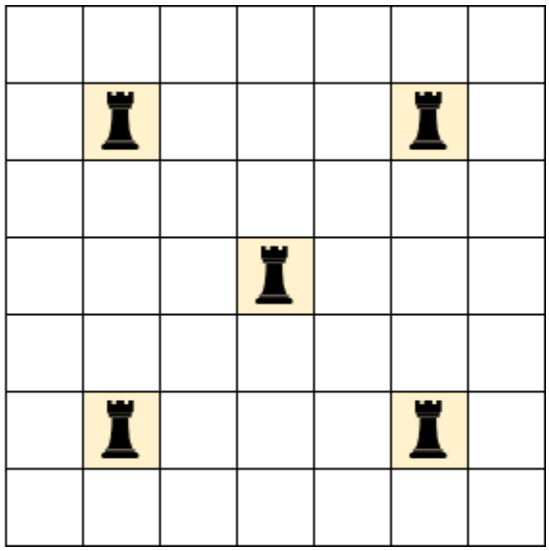
If there are multiple valid solutions, you can output any of them.

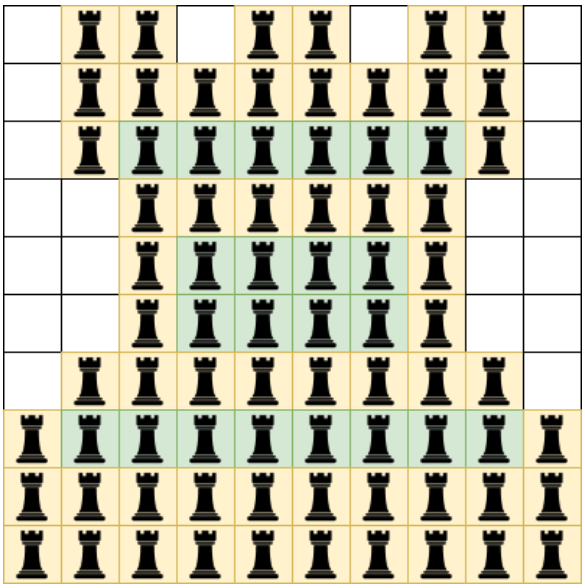
Sample explanation

These below figures demonstrate three test cases in the sample. The top-left one is the board after adding rook in the first test case, with each pair of *guarding* rooks is represented by a red line connecting the two rooks. The top-right one is the initial board in the second test case. The bottom one is the board after adding the rook in the third test case.

In the second test case, you can not add any rooks ($k = 0$), but initially there are only 4 pairs of *guarding* rooks. Hence it is not possible to have **exactly** $p = 12$ pairs.







Sample Input 1	Sample Output 1
7 4 12 .... .R...R. ....R... .... .R...R. .... 7 0 12 .... .R...R. ....R... .... .R...R. .... 10 22 136 .RR.RR.RR. .RRRRRRR. .R.....R. ..RRRRR.. ..R....R.. ..R....R.. .RRRRRRR. R.......R RRRRRRRRR RRRRRRRRR 0 0 0	YES .... .R.R.R. .... .R.R.R. .... .R.R.R. .... NO YES .RR.RR.RR. .RRRRRRR. .RRRRRRR. ..RRRRR.. ..RRRRR.. ..RRRRR.. .RRRRRRR. RRRRRRRRR RRRRRRRRR RRRRRRRRR