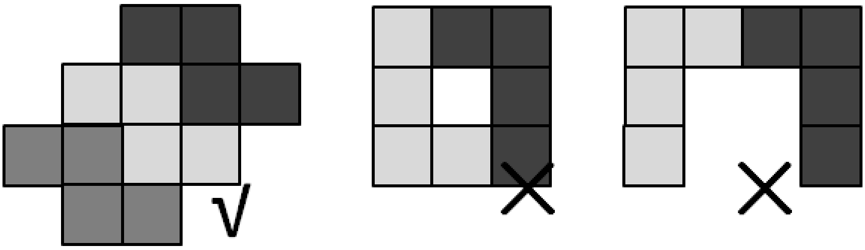
> A polyomino is a plane geometric figure formed by joining one or more equal squares edge to edge.
>
> – *Wikipedia*

Given a large polyomino, your task is to decompose it into at least two identical polyominoes (can't be flipped or rotated). The leftmost picture below is a correct way of decomposing, but the right two pictures are not. In the middle picture, one of the pieces is rotated. In the rightmost picture, one of the pieces is flipped. The number of pieces should be as small as possible. Note that there is at least one solution: decompose it into a lot of unit squares.



Input

There will be at most 30 test cases. Each test case begins with an integer n ($1 \leq n \leq 10$) in a single line. The next n lines describe the large polyomino. Each of these lines contains exactly n characters in '`*`','`.`'. A '`*`' indicates an existing square, and a '`.`' indicates an empty square. These characters are guaranteed to form a valid polyomino. There are at least one and at most twenty existing squares in the large polyomino. The input terminates with $n = 0$, which should not be processed.

Output

For each test case, output a line containing the decomposition of the large polyomino. Each existing square is replaced by a capital letter representing the label of the piece. Different pieces should use different labels. If there are multiple solutions, print the lexicographically smallest one. That is, if we regard the solution as a long string, by concatenating rows from top to bottom, your output should be lexicographically the smallest one among all possible solutions. Print an empty line after each case.

Sample Input

```
5
..**.
.****
****.
.**..
.....
2
**
**
0
```

Sample Output

```
..AA.
.AABB
AABB.
.BB..
.....

AA
BB
```