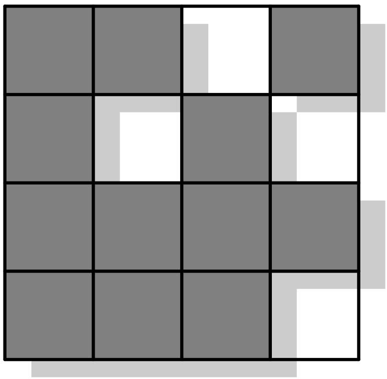
The Grille

`grille.c`, `grille.C`, `grille.java`

In the 16th century, there were no computers as we have today. On the other hand, even at that time it was necessary to protect messages from being read by inappropriate people. Old methods, like shaving a head of a slave, writing the message on his head, waiting until his hair grows back and then sending him through the area full of enemies, might work well but they took a long time. That is why new methods had to be invented. For example, an Italian mathematician Girolamo Cardano was the first one who described the *Grille* cipher.

For both enciphering and deciphering, you need a tool (basically, it is the cipher key) called the "grille". It is necessary that both parties (Alice and Bob) have the same grille. The grille looks like a rectangular grid of $N \times N$ unit squares with some of the squares being solid and some cut out to form "holes".



Let's have a grille with m holes. For enciphering, we write the first m letters of a message into the holes (starting in the first row from left to right, then continuing with other rows). Then we rotate the grille 90 degrees *clockwise* and write another m letters into the holes (again, from left to right and top to bottom). After that, we rotate the grille another 90 degrees and write another m letters. Then we repeat the same for the last time. At the end, we fill remaining places (if there are any) with random letters to make the ciphertext more secure. Please note that it is the *grille* that gets rotated, not the message!

For deciphering, we basically use the same algorithm, we just read the letters instead of writing them.

Input Specification

The input contains several test cases. Each test case contains description of a grille and a ciphertext. Your task is to decipher the message and write the plaintext to output.

Each test case starts with a line containing number N ($1 \le N \le 1000$), where N is the size of the grille. Then there are N lines containing the grille description. Each of those lines contains exactly N characters which are either the "hash" character "`#`" (solid/opaque material) or the uppercase letter "`O`" (hole).

Note: In praxis, the grille holes would be arranged in such a way that no position of the ciphertext is used more than once. In our problem, this is not guaranteed. Some grilles may contain holes that match the same position/letter of the ciphertext (after rotations). However, the deciphering algorithm is still the same.

After the grille description, there are another N lines with the enciphered message. Each of them contains exactly N characters — uppercase letters of alphabet.

The last test case is followed by a line containing one zero.

Output Specification

For each test case, output the deciphered message (plaintext) on one line with no spaces.

Sample Input

```
4
##O#
#O#O
####
###O
ARAO
PCEM
LEEN
TURC
3
O#O
###
O#O
ABC
DEF
GHI
0
```

Output for Sample Input

```
ACMCENTRALEUROPE
ACGIACGIACGIACGI
```