Bodyguards

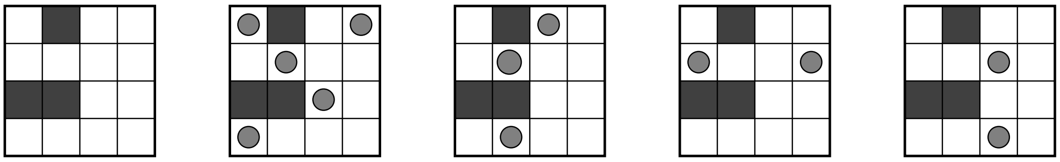
`guards.p`, `guards.c`, `guards.C`

A new directive of European Union specifies exact rules for positioning bodyguards of important persons, such as presidents, government members, etc. These rules are very restrictive and must be obeyed by member states.

For the purpose of the directive, an area guarded by bodyguards is considered to form a rectangular system. Each square field of the area is either empty or contains a wall. No two bodyguards can be placed onto the same horizontal row or vertical column unless there is at least one wall separating them. The goal is to place bodyguards to the area so that no two of them violate the above rule. Such position is then called *legal.*

Your task is to write a program that, given a description of an area, calculates the maximum number of bodyguards that can be placed there in a legal configuration. The program will be a prototype only, thus, it is enough to restrict our attention to small square areas (at most 4×4 fields).

The following image shows five pictures of the same area. The first picture is the empty area, the second and third pictures show legal configurations, and the fourth and fifth pictures show illegal configurations. For this area, the maximum number of bodyguards in a legal configuration is 5; the second picture shows one way to do it, but there are several other ways.



Input Specification

The input file contains one or more area descriptions, followed by a line containing the number 0 that signals the end of the file. Each area description begins with a line containing a positive integer n that is the size of the area; n will be at most 4. The next n lines each describe one row of the area, with a dot ("`.`") indicating an open space and an uppercase letter "`X`" indicating a wall. There are no spaces in the input file.

Output Specification

For each test case, output one line containing the maximum number of bodyguards that can be placed in the area in a legal configuration.

Sample Input

```
4
.X..
....
XX..
....
2
XX
.X
3
.X.
X.X
.X.
3
...
.XX
.XX
4
....
....
....
....
0
```

Output for Sample Input

```
5
1
5
2
4
```