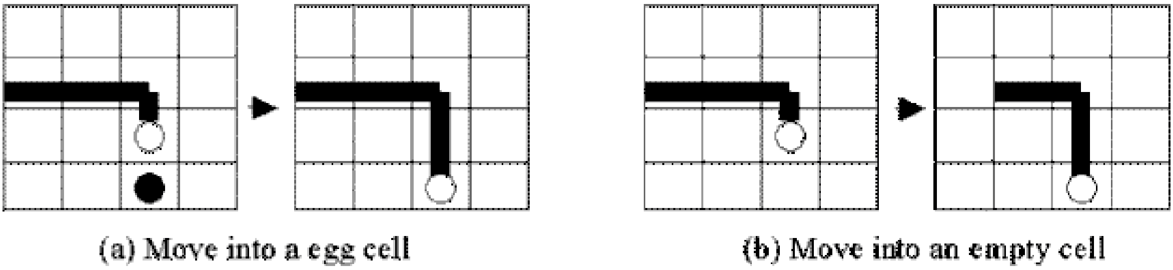
Consider a snake moving around a labyrinth in which there are obstacules and eggs. Starting from a given initial position, the goal of the snake is to eat all existing eggs without going into the obstacules or touching itself. Initially, the snake is small but its size increases by one unit each time it eats an egg (obviously, it never goes out of the labyrinth).

The snake can move in any direction (up, down, left and right), one unit per move. A tentative move that takes the head of the snake into one obstacule is not valid, hence it is not accomplished.

Write a program that given a labyrinth, a set of obstacules and a set of eggs, determines the smallest number of movements necessary for the snake to eat all eggs, if such a solution exists. If there is no solution then it outputs a no solution message.

Initially, the snake has size one (therefore occupies just one labyrinth cell) and is positioned in cell $(0, 0)$ (left upper corner of the labyrinth) where it is guaranteed that there is no egg or obstacule. Each time the snake eats an egg (i.e. moves for the first time to a position with an egg in it) its size increases by one unit and, as a result of the movement, its head occupies the cell where the egg was located. The following figure illustrates such movement (a) and the case in which the move is to an empty cell (b).



(a) Move into a egg cell

(b) Move into an empty cell

Figure 1: Snake movements

Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

The input includes the size of the labyrinth (max. 20), followed by a description of the labyrinth itself. The labyrinth is given as a set of strings, each representing a line of the labyrinth and composed by a sequence of the following characters: '-', 'x' and 'o'. The '-' indicates an empty cell, 'x' indicates an obstacule and 'o' indicates an egg.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

The output should be the smallest number of movements made by the snake to eat all eggs. If there is no solution, then the output should be 'NO'.

Sample Input

```
1

4
---o
-x--
-x--
ox--
```

Sample Output

```
11
```