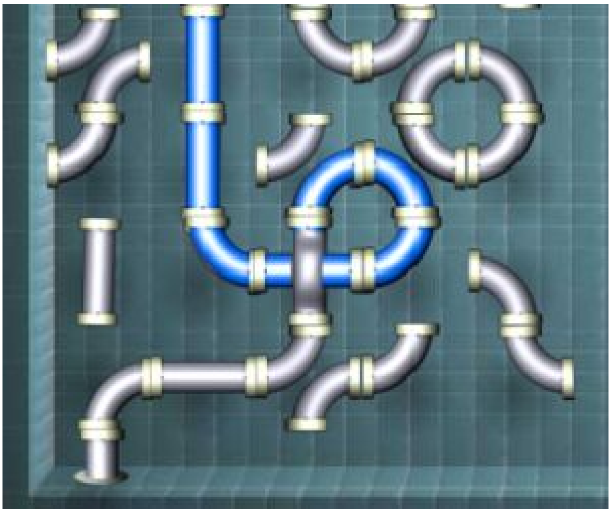

After writing a solver for the "moveable maze" game last week, you have grown tired of it. After all, you already know the optimal solution. To entertain yourself, you find another puzzle game called "Pipes", and play that for a while. On one puzzle, you have not been able to find a solution by hand, and you think that there is no solution. You decide to write a program to tell you whether this is the case.

The game is played on a grid with R rows and C columns. Each square of the grid contains a black dot in the centre and black lines in the direction of some, none, or all of its north, east, south, and west neighbouring squares, with the following restriction: if two opposite directions both have lines, then at least one of the other two directions has a line as well. In other words, it is forbidden for a square to consist of a straight line.

The objective of the game is to rotate each square, as many times as you like, such that for each square, if it has a line going in a compass direction (that is, north, east, south, or west), then it has a neighbour in that compass direction and that neighbour has a line going in the opposite compass direction. In other words, each edge in the grid should either have a line on both sides or neither side.

Your task is to determine whether a given grid has a solution.

Input

The input consists of several test cases.

The first line of each test case contains the two integers R and C, separated by spaces, with $1 \leq R, C \leq 12$.

The following R lines of input each contain one row of the grid, from north to south. Each of these lines contains exactly C strings of letters, separated by spaces, that correspond to squares of the grid, from west to east. Their format is as follows:

- If the string is the single character x, then the square does not contain a line to any of its neighbours.
- Otherwise, the string contains some of the characters 'N', 'E', 'S', 'W', which indicate that a black line extends from this square's centre in the direction of its north, east, south, or west neighbour, respectively. No character will appear in the string more than once.

Input is terminated by a line containing '0 0'. These zeros are not a test case and should not be processed.

Output

For each test case, output 'SOLVABLE' if there is a solution to the puzzle, and 'UNSOLVABLE' otherwise.

Sample Input

```
3 3
NW NW x
NES NESW W
E W x
2 2
ES x
x N
0 0
```

Sample Output

```
SOLVABLE
UNSOLVABLE
```