

C

Polyomino Composer

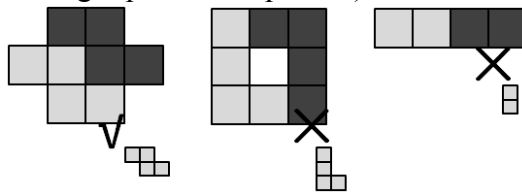
Input: Standard Input
Output: Standard Output



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

-- Wikipedia

Given a large polyomino and a small polyomino, your task is to determine whether you can compose the large one with two copies of the small one. The polyominoes can be translated, but not flipped or rotated. The two pieces should not overlap. The leftmost picture below is a correct way of composing the large polyomino, but the right two pictures are not. In the middle picture, one of the pieces was rotated. In the rightmost picture, both pieces are exactly identical, but they're both rotated from the original piece (shown in the lower-right part of the picture).



Input

There will be at most 20 test cases. Each test case begins with two integers n and m ($1 \leq m \leq n \leq 10$) in a single line. The next n lines describe the large polyomino. Each of these lines contains exactly n characters in $\{*, '\cdot'\}$. A $*$ indicates an existing square, and a \cdot indicates an empty square. The next m lines describe the small polyomino, in the same format. These characters are guaranteed to form valid polyominoes (note that a polyomino contains at least one existing square). The input terminates with $n=m=0$, which should not be processed.

Output

For each case, print 1 if the corresponding composing is possible, print 0 otherwise.

Sample Input

```

4 3
. * .
****
. * .
....
* .
. *
...
3 3
***
* . *
***
* .
* .
* .
**
4 2
*****
....
....
....
* .
* .
0 0

```

Output for Sample Input

```

1
0
0

```