Cleaning Robot

Here, we want to solve path planning for a mobile robot cleaning a rectangular room floor with furniture.

Consider the room floor paved with square tiles whose size fits the cleaning robot (1 × 1). There are 'clean tiles' and 'dirty tiles', and the robot can change a 'dirty tile' to a 'clean tile' by visiting the tile. Also there may be some obstacles (furniture) whose size fits a tile in the room. If there is an obstacle on a tile, the robot cannot visit it. The robot moves to an adjacent tile with one move. The tile onto which the robot moves must be one of four tiles (i.e., east, west, north or south) adjacent to the tile where the robot is present. The robot may visit a tile twice or more.

Your task is to write a program which computes the minimum number of moves for the robot to change all 'dirty tiles' to 'clean tiles', if ever possible.

Input

The input consists of multiple maps, each representing the size and arrangement of the room. A map is given in the following format.

w h
c11 c12 c13 ... c1w
c21 c22 c23 ... c2w
...
ch1 ch2 ch3 ... chw

The integers w and h are the lengths of the two sides of the floor of the room in terms of widths of floor tiles. w and h are less than or equal to 20. The character cyx represents what is initially on the tile with coordinates (x, y) as follows.

'.' : a clean tile
'*' : a dirty tile
'x' : a piece of furniture (obstacle)
'o' : the robot (initial position)

In the map the number of 'dirty tiles' does not exceed 10. There is only one 'robot'.

The end of the input is indicated by a line containing two zeros.

Output

For each map, your program should output a line containing the minimum number of moves. If the map includes 'dirty tiles' which the robot cannot reach, your program should output -1.

Example

Input:

```
7 5
.......
.o...*.
```

```
.......
.*...*.
.......
15 13
.......x.......
...o...x....*..
.......x.......
.......x.......
.......x.......
...............
xxxxx.....xxxxx
...............
.......x.......
.......x.......
.......x.......
..*....x....*..
.......x.......
10 10
..........
..o.......
..........
..........
..........
.....xxxxx
.....x....
.....x.*..
.....x....
.....x....
0 0
```

Output:

```
8
49
-1
```