Problem B: Fun House

Source file: `fun.{c, cpp, java}`
Input file: `fun.in`

American Carnival Makers Inc. (ACM) has a long history of designing rides and attractions. One of their more popular attractions is a fun house that includes a room of mirrors. Their trademark is to set up the room so that when looking forward from the entry door, the exit door appears to be directly ahead. However, the room has double-sided mirrors placed throughout at 45 degree angles. So, the exit door can be on any of the walls of the room. The set designer always places the entry and mirrors, but can never seem to be bothered to place the exit door. One of your jobs as part of the construction crew is to determine the placement of the exit door for the room given an original design.

The final diagram for a sample room is given below. The asterisk (`*`) marks the entry way, lower case x's mark the walls, the mirrors are given by the forward and backward slash characters (`/ and \`), open spaces with no visual obstructions are marked by periods (`.`), and the desired placement of the exit is marked with an ampersand (`&`). In the input diagram, there is an 'x' in place of the '&', since the exit has not yet been located. You need to alter the input diagram by replacing the proper 'x' with an '&' to identify the exit. Note that entrances and exits can appear on any of the walls (although never a corner), and that it is physically impossible for the exit to be the same as the entrance. (You don't need to understand why this is so, although it may be fun to think about.)

```
xxxxxxxxxxx
x../..\...x
x..../....x
*../......x
x.........x
xxxxxx&xxxx
```

Input: Each room will be preceded by two integers, W and L, where 5 ≤ W ≤ 20 is the width of the room including the border walls and 5 ≤ L ≤ 20 is the length of the room including the border walls. Following the specification of W and L are L additional lines containing the room diagram, with each line having W characters from the alphabet: { `*` , `x` , `.` , `/` , `\` }. The perimeter will always be comprised of walls, except for one asterisk (`*`) which marks the entrance; the exit is not (yet) marked. A line with two zeros indicates the end of input data.

Output: For each test case, the first line will contain the word, HOUSE, followed by a space and then an integer that identifies the given fun house sequentially. Following that should be a room diagram which includes the proper placement of the exit door, as marked by an ampersand (`&`).

Note: In both Java and C++ the backslash character (`\`) has special meaning as an escape character within character and string literals. You must use the combination `\\` to express a single backslash within a character or string literal within source code.

Example Input:	**Example Output:**

`11  6`
`xxxxxxxxxxx`
`x../..\...x`
`x..../....x`
`*../......x`
`x.........x`
`xxxxxxxxxxx`
`5 5`
`xxxxx`
	`HOUSE 1`
	`xxxxxxxxxxx`
	`x../..\...x`
	`x..../....x`
	`*../......x`
	`x.........x`
	`xxxxxx&xxxx`
	`HOUSE 2`
	`xxxxx`

```
*...x
x...x
x...x
xxxxx
5 5
xxxxx
x./\x
*./.x
x..\x
xxxxx
6 6
xxx*xx
x/...x
x....x
x/./.x
x\./.x
xxxxxx
10 10
xxxxxxxxxx
x.../\...x
x........x
x........x
x.../\..\x
*...\/../x
x........x
x........x
x...\/...x
xxxxxxxxxx
0 0
```

```
*...&
x...x
x...x
xxxxx
HOUSE 3
xxxxx
x./\x
*./.x
x..\&
xxxxx
HOUSE 4
xxx*xx
x/...x
x....x
x/./.&
x\./.x
xxxxxx
HOUSE 5
xxxxxxxxxx
x.../\...x
x........x
x........x
&.../\..\x
*...\/../x
x........x
x........x
x...\/...x
xxxxxxxxxx
```