In most puzzles we are given some pieces and we have to make a target pattern which can be built in only one possible way. But some puzzles are a bit different, we are given a target pattern and from that target pattern we have to find in how many ways the pieces can be placed. Such a puzzle is the puzzle of overlapping squares. To understand this puzzle, look at the pictures below:

In first figure we have placed a (2×2) filled square in a (4×4) grid. In the second figure we have placed another (2×2) filled square in the grid, which have of course deleted some part of the black lines of the previous square, in third picture we have placed a third square and in the fourth picture we have placed a fourth square. The picture can become even more complex if we place more (2×2) squares.

Write a program to determine if it's possible to form a target image using between 1 and 6 pieces (inclusive) of 2×2 squares.

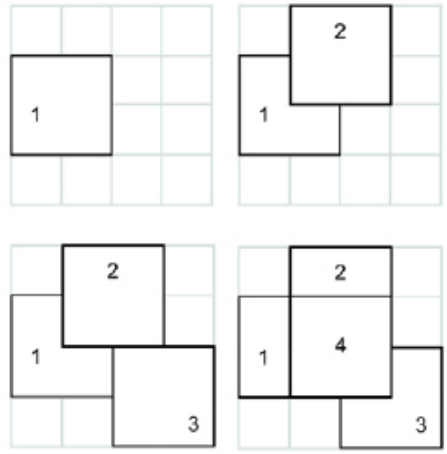

Fig 1. Placing four filled squares in an empty 4×4 grid.

Input

The input consists of several test cases. Each test case is contained in five lines and each line contains nine characters. If the horizontal border of a filled square is visible it is denoted with '_' (ASCII value 95) sign and if vertical border of a filled square is visible then it is denoted with '|' (ASCII value 124) character. The board contains no other character than '_', '|' and of course ' ' (ASCII Value 32). The border lines of the squares can only be along the grid lines. Each board lines end with a '#' (Hash character) which denotes the end of line. This character is not a part of the grid or square. The last test case is followed by a single zero, which should not be processed.

Output

For each test case, print the case number and '`Yes`' or '`No`', depending on whether it's possible to form the target.

Sample Input

```
         #
 _ _ _   #
|_|_ _|  #
|_|   |  #
  |_ _|  #
         #
   _ _   #
  |   |  #
  |_ _|  #
         #
 _ _ _ _ #
|_|_|_|_|#
|_|_|_|_|#
|_|_|_|_|#
|_|_|_|_|#
   _ _   #
 _|   |_ #
|_|_ _|_|#
|_|   |_|#
  |_ _|_|#
0
```

Sample Output

```
Case 1: Yes
Case 2: Yes
Case 3: No
Case 4: Yes
```