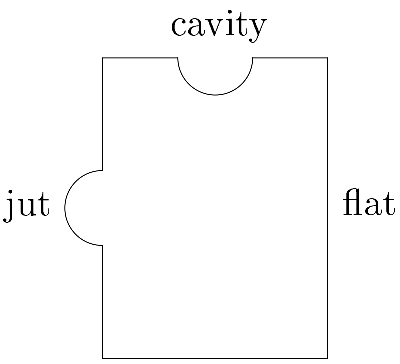
519 Puzzle (II)

Little Barborka has just started to learn how to solve a picture puzzle. She has started with a small one containing 15 pieces. Her daddy tries to solve the puzzle too. To make it a little bit harder for himself, he has turned all puzzle pieces upside down so that he cannot see pictures on the pieces. Now he is looking for a solution of the puzzle. Normally the solution should exist but he is not sure whether Barborka has not replaced some pieces of the puzzle by pieces of another similar puzzle. Help him and write a program which reads a description of a set of puzzle pieces and decides whether it is possible to assembly the pieces into a rectangle with given side lengths or not.

Input

The input file consists of blocks of lines. Each block except the last describes one puzzle problem. In the first line of the block there are integers n and m, $0 < n, m \leq 6$ separated by one space. The integers n, m indicate the number of rows and columns in the puzzle respectively. The description of individual puzzle pieces is in the following $n \times m$ lines of the block. Each piece is a rectangle 3 centimeters wide and 4 centimeters high with possible juts or cavities in the middle of its sides. For each side of a puzzle piece just one of the following possibilities is true (see picture):

- there is no jut or cavity on the side, i.e., the side is flat – such sides can be used only on edges of the final picture when assembling the puzzle,
- there is one jut in the middle of the side,
- there is one cavity in the middle of the side.



As is usual, two pieces can be placed side by side only if one has a jut and the other has a cavity on corresponding sides. We will denote the flat sides by `F`, the sides with juts by `O` and the sides with cavities by `I`. Each piece is described by four letters characterizing its top, right, bottom, and left side. To make the task easier the pieces can be used only as they are described i.e. they cannot be turned.

After each block there is an empty line. The last block consists of just one line containing 0 0, i.e. two zeros separated by one space.

Output

The output file contains the lines corresponding to the blocks in the input file. A line contains `YES` if the corresponding block in the input file describes a puzzle that can be correctly assembled. Otherwise it contains `NO`. There is no line in the output file corresponding to the last "null" block of the input file.

Sample Input

```
3 5
FOOF
FOOI
FOOI
FOOI
FFOI
IOOF
IOOI
IOOI
IOOI
IFOI
IOFF
IOFI
IOFI
IOFI
IFFI
0 0
```

Sample Output

```
YES
```