MWC '15 #3 P3: Bad News

Time limit: 0.6s **Memory limit:** 32M

Hypnova has been working on a crossword puzzle that he found in a newspaper. The crossword consists of an N by N grid, with each cell having one letter of the alphabet. He is given Q words to find in the crossword.

However, as the puzzle author was lazy, it is not guaranteed that each word can be found in the puzzle. For a word to exist on the board, each letter of the word must exist on the board and must be sequentially adjacent to each other, either horizontally, vertically or diagonally (similar to the rules of the board game Boggle). No letter can be used more than once in the making of a single word.

Wanting to save **Hypnova** some time, you decide to write a program to help him search for all the words!

Input Specification

On one line, space separated integers N $(1 \leq N \leq 25)$ and Q $(1 \leq Q \leq 50)$ are given, representing the grid size and the number of words that **Hypnova** must find.

The next N lines each contain N space separated characters, representing the characters found on the n^{th} row of the crossword puzzle.

The next Q lines will each contain one string, representing one of the words that Hypnova must find in the crossword.

Output Specification

For each word, output `good puzzle!` if it can be found in the crossword, or `bad puzzle!` if it cannot be found within the crossword (case insensitive).

Sample Input

```
4 4
p e a r
h a p s
t y i a
q s t r
pear
asfasfa
pair
pats
```

Sample Output

```
good puzzle!
bad puzzle!
good puzzle!
good puzzle!
```

Explanation for Sample Output

The below diagram shows how each word is made. Note that the second word cannot be found in the puzzle.

P	E	A	R
H	A	P	S
T	Y	I	A
Q	S	T	R