Problem Statement

You have a rectangular chessboard divided into unit squares. Each square is either empty or contains a rook. You are given the String[] **board** describing which squares of the board contain rooks. If **board**[i][j] equals 'R', there is a rook on the square in row i, column j. If **board**[i][j] equals '.', the corresponding square is empty.

We say that two rooks attack each other if and only if all the following conditions are satisfied:

- The rooks are of different colors.
- They are in the same row or in the same column.
- There are no other rooks between them.

You are going to color all the rooks. We say that a coloring of rooks is *friendly* if no two rooks attack each other. You want to produce a friendly coloring. What is the maximum number of distinct colors you may use?

Definition

Class:	FriendlyRooks
Method:	getMinFriendlyColoring
Parameters:	String[]
Returns:	int
Method signature:	int getMinFriendlyColoring(String[] board)
(be sure your method is public)	

Constraints

- **board** will contain between 1 and 20 elements, inclusive.
- Each element of **board** will contain between 1 and 20 characters, inclusive.
- All the elements of **board** will contain the same number of characters.
- Each character of each element of **board** will be 'R' or '.'.

Examples

0)

```
{".R.R",
 "R.R.",
 ".R.R"}
Returns: 2
```

The friendly coloring shown below uses two colors (denoted 1 and 2). We can show that there is no friendly coloring with more than two colors.

```
.1.1
2.2.
.1.1
```

1)

```
{"RRRRRRRRRRRRRR"}
Returns: 1
```

All rooks must share the same color.

2)

```
{"..............",
 "..............",
 "..............",
 "..............",
 "..............",
 ".............."}
```

```
Returns: 0
```

3)

```
{"....R.........",
 ".R..........R.",
 "....R.........",
 ".R........R....",
 "....R.........",
 "....R.....R...."}
```

```
Returns: 1
```

4)

```
{"R........R",
 ".R.......R.",
 "..R.....R..",
 "...R...R...",
 "....R.R....",
 ".....R.....",
 "....R.R....",
 "...R...R...",
 "..R.....R..",
 ".R.......R.",
 "R........R"}
```

```
Returns: 6
```