acm International Collegiate Programming Contest 2001

South Central USA Regional Programming Contest

Problem #5: Deeper Blue

Introduction

When trying to avoid conflict and maintain peace, a good strategy is to remove the elements that cause the most trouble. IBM is using its Deep Blue machine to try to study this strategy by modeling it with a game of chess. IBM needs a program to find the minimum number of chess pieces that must be removed from a chessboard in order for none of the pieces to be attacking each other.

All pieces will have the standard attack movements for that chess piece.

King - Can attack the adjacent space in any direction. Up, down, left, right and diagonally.
Queen - Can attack any number of spaces in any direction. Up, down, left, right and diagonally.
Bishop - Can attack any number of spaces diagonally.
Rook - Can attack any number of spaces up, down, left, or right but not diagonally.
Pawns - There won't be any pawns.
Knight - Can attack with their usual 'L' shaped movement. Up twice and right once, up once and left twice, etc... Here is a diagram, which demonstrates the movement of a knight.

```
 --------------
| | |*| |*| | |
---------------
| |*| | | |*| |
---------------
| | | |N| | | |
---------------
| |*| | | |*| |
---------------
| | |*| |*| | |
---------------
N = Knight
* = Square that the knight can attack
```

Input

Input to this problem will consist of a (non-empty) series of up to 100 data sets. Each data set will be formatted according to the following description, and there will be **no blank lines** separating data sets. The maximum dimensions of the board are 10 squares wide by 10 squares high. The maximum number of chess pieces that will start out on the board is 15.

A single data set has 5 components:

1. *Start line* - A single line, `"START"`
2. *Board Width (# of columns)* - A single line containing a positive integer, *w*, indicating the number of squares across the width of the board where 1 <= *w* <= 10.

3. *Board Height (# of rows)* - A single line containing a positive integer, *h*, indicating the number of squares that dictate the height of the board, where 1 <= *h* <= 10.
4. *Board Layout* - *h* lines, each corresponding to a row of the board. The first line corresponds to the first row, the second line to the second row, and so on. Each row consists of a space-separated list of single letters, each representing the contents of the corresponding square on the board according to the following list:
    - `K` - King
    - `Q` - Queen
    - `R` - Rook
    - `B` - Bishop
    - `N` - Knight
    - `E` - Empty Square
5. *End line* - A single line, "`END`"

Output

For each data set, there will be exactly one output set, and there will be **no blank lines** separating output sets.

A single output set consists of a single line, "`Minimum Number of Pieces to be removed: X`", where *X* is the minimum number of pieces that must be removed from the board such that none of the remaining pieces are attacking any other remaining piece.

Sample Input

```
START
3
3
K E K
E Q E
K E K
END
START
8
8
E E E E E E E E
E B E K E E N E
E E E E N E E E
E E E E E E E R
B E Q E E E E E
E E E E E Q E E
E E E E E B E E
E B E R E E E E
END
```

Sample Output

```
Minimum Number of Pieces to be removed: 1
Minimum Number of Pieces to be removed: 5
```