Problem C: Check The Check

The Problem

Your task is to write a program that reads a chess board configuration and answers if there's a king under attack (i.e. "in check"). A king is in check if it's in a square which is attacked by an oponnet's piece (i.e. it's in square which can be taken by an oponnet's piece in his next move).

White pieces will be represented by uppercase letters whereas black pieces will be represented by lowercase letters. White side will always be on the bottom of the board and black side will always be on the top of the board.

For those unfamiliar with chess, here are the movements of each piece:

Pawn (p or P): can only move straight ahead, one square at a time. But it takes pieces diagonally (and that's what concerns to you in this problem).
Knight (n or N): have a special movement and it's the only piece that can jump over other pieces. The knight movement can be viewed as an "L". See the example bellow.
Bishop (b or B): can move any number of squares diagonally (forward or backward).
Rook (r or R): can move any number of squares vertically or horizontally (forward or backward).
Queen (q or Q): can move any number of squares in any direction (diagonally, horizontally or vertically, forward or backward).
King (k or K): can move one square at a time, in any direction (diagonally, horizontally or vertically, forward or backward).

Movements examples ('*' indicates where the piece can take another pieces):

```
        Pawn        Rook        Bishop      Queen       King        Knight
        ........    ...*....    .......*    ...*...*    ........    ........
        ........    ...*....    *......*    *..*..*.    ........    ........
        ........    ...*....    .*...*..    .*.*.*..    ........    ..*.*...
        ........    ...*....    ..*.*...    ..***...    ..***...    .*...*..
        ...p....    ***r****    ...b....    ***q****    ..*k*...    ...n....
        ..*.*...    ...*....    ..*.*...    ..***...    ..***...    .*...*..
        ........    ...*....    .*...*..    .*.*.*..    ........    ..*.*...
        ........    ...*....    *......*    *..*..*.    ........    ........
```

Remember that the knight is the only piece that can jumper over other pieces. The pawn movement will depend on its side. If it's a black pawn, it can only move one square diagonally down the board. If it's a white pawn, it can only move one square diagonally up the board. The example above is a black pawn as it's a lowercase p (we say "move" meaning the squares where the pawn can move to when it takes another piece).

The Input

There will be an arbitrary number of board configurations on the input. Each board will consist of 8 lines of 8 characters each. A '.' character will represent an empty square. Upper and lower case letters (as defined above) will represent the pieces. There will be no invalid characters (i.e. pieces) and there won't be a configuration where both kings are in check. You must read until you find an empty board (i.e. a board that is formed only of '.' characters) which should not be processed. There will be an empty line

between each pair of board configurations. In all boards (except the last one which is empty) will appear both the white king and the black king (one, and only one of each).

The Output

For each board configuration read you must output one of the following answers:

```
Game #d: white king is in check.
Game #d: black king is in check.
Game #d: no king is in check.
```

Where d stands for the game number (starting from 1).

Sample Input

```
..k.....
ppp.pppp
........
.R...B..
........
........
PPPPPPPP
K.......

rnbqkbnr
pppppppp
........
........
........
........
PPPPPPPP
RNBQKBNR

rnbqk.nr
ppp..ppp
....p...
...p....
.bPP....
.....N..
PP..PPPP
RNBQKB.R

........
........
........
........
........
........
........
........
```

Sample Output

```
Game #1: black king is in check.
Game #2: no king is in check.
Game #3: white king is in check.
```

© 2001 Universidade do Brasil (UFRJ). Internal Contest 2001.