



# [CS 11] Prac 8j – Queen Threats

## Problem Statement

Consider an  $r \times c$  chessboard with some chess pieces in it. For simplicity, we ignore the actual rules of chess. For example, chess pieces don't have "color".

We say a square on the chessboard is **unsafe** if a piece can reach it in one move.

In one move, a queen can travel to any other square in its row or column, or in any of its 45-degree diagonals, as long as there are no other pieces on the way there.

Given the locations of queens, label which squares are unsafe.

[Submit solution](#)

[CS 11]

Practice 8

✓ Points: 160 (partial)

⌚ Time limit: 6.0s

💻 Memory limit: 1G

✉ Author:

kvatienza (Kevin Atienza)

➤ Problem type

▼ Allowed languages

NONE, py3

## Task Details

Your task is to implement a function called `queen_threats`. This function has a single parameter: a `tuple` of  $r$  `str`s, each of which is  $c$  characters long and represents a row. Each character represents a square and is:

- a `.` if it is free;
- a `#` if it contains a queen.

The function must return a `list` of `str`s representing the same chessboard, but with the unsafe squares labelled with `*`.

## Restrictions

(See 8a for more restrictions)

For this problem:

- Loops and lists are allowed.
- Up to 18 function definitions are allowed.
- Recursion is **disallowed**. (The recursion limit has been greatly reduced.)
- Sets and dictionaries are allowed.
- Generators and comprehensions are allowed.
- The source code limit is 6000.

## Example Calls

### Example 1 Function Call

```
queen_threats((  
    '.....',  
    '...#.',  
    '.....',  
    '.....',  
    '.....#',  
    '.....',  
    '.....',  
    '.....',  
)
```

[Copy](#)

### Example 1 Return Value

```
[  
    '.....*****',  
    '*****#***',  
    '.....****',  
    '.....*..***',  
    '*****#*****',  
    '.....*....***',  
    '.....*....**.',  
    '.....*....**.*',  
)
```

[Copy](#)

### Example 1 Explanation

**Hint:** You can print a grid of `str`s by doing:

```
for row in grid:  
    print(row)
```

[Copy](#)

or by doing

```
print(*grid, sep='\n')
```

[Copy](#)

## Constraints

- The function `queen_threats` will be called at most 1,000 times.
- The sum of the  $rc$  across all calls will be at most 500,000.
- $1 \leq r \leq 100$
- $1 \leq c \leq 5,000$

## Scoring

- You get 100 points if you solve all test cases where:
  - $r, c \leq 50$
  - the sum of the  $rc$  across all calls will be at most 10,000.
- You get 30 points if you solve all test cases where:
  - there are at most 5000 pieces.
- You get 30 points if you solve all test cases.

## ?

### Clarifications

[Report an issue](#)

No clarifications have been made at this time.