

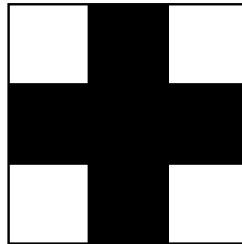
UCF Local Contest — September 5, 2015

Reach for the Stars

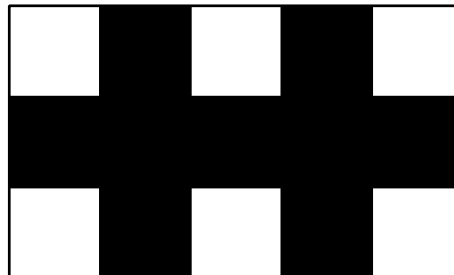
filename: stars
(*Difficulty Level:* Hard)

When you were a little kid, you would stamp papers randomly but your parents were still so impressed by your art. Now you are dealing with the UCF Programming Team Coaches and you need to be more creative to impress them. Good luck!

You are given a star shaped stamp like the one below. The black area is covered in ink and the white area is not. When the stamp hits the paper, it leaves a mark for each cell of ink that hits the paper.



For example, the image below can be made with two stampings. Notice the stamp must always remain axis-aligned when hitting the paper. We also require that the stamp be completely contained within the paper. Note a cell of paper stamped once with black ink is indistinguishable from a cell of paper stamped multiple times with black ink. Note also that cells and stamp line up properly, i.e., a cell is either covered completely by the stamp or not covered at all, i.e., the stamp will not cover part of a cell.



The Problem:

Given a black and white image, determine the minimum number of times, if possible, you would need to stamp the paper with the star stamp to end up with the design specified.

The Input:

The first input line contains a positive integer, n , indicating the number of images to evaluate. Each image starts with a line containing two integers, r and c , ($1 \leq r \leq 9$, $1 \leq c \leq 9$), representing the number of rows and columns, respectively. The next r input lines contains c characters each. The characters are either '.', representing a blank cell of the image and '#', representing a cell of the image covered in ink.

The Output:

For each image, output "Image # d : v " where v is the minimum number of stampings required to make the image. Replace v with "impossible" (without quotes) if it is not possible to form the image using the star shaped stamp. Leave a blank line after the output for each test case.

Sample Input:

```
5
1 1
.
1 1
#
3 3
.#.
###
.#.
3 5
.##.#.
#####
.##.#.
4 7
.##.#..
#####.
.#####
..#...#.
```

Sample Output:

```
Image #1: 0

Image #2: impossible

Image #3: 1

Image #4: 2

Image #5: 5
```