Problem H: Hangar Hurdles

Time limit: 8 s
Memory limit: 512 MiB

You are evaluating constructions plans for a giant new hangar that will house an airplane assembly line. The hangar floor can be represented as a rectangular grid consisting of n rows and n columns where every cell is either empty or blocked. The rows are numbered with integers 1 through n top to bottom, while the columns are numbered with integers 1 through n left to right.

It is important that large crates containing airplane parts can be freely moved between various locations inside the hangar. We can model a crate as a square aligned with grid cells and centered in one of the cells. Therefore, for an odd integer k, a *crate of size* k consists of cells in k consecutive rows and k consecutive columns. The crate can be moved up, down, left or right as long as it fits completely inside the grid and never contains a blocked cell.

You are given q pairs of cells A_k and B_k. For each pair, find the size of the largest crate that can be centered in A_k and then moved across the hangar floor until it's centered in B_k.

Input

The first line contains an integer n ($2 \leq n \leq 1\,000$) — the size of the hangar floor. Each of the following n lines contains a string of exactly n characters describing one row of the floor. The character "#" denotes a blocked cell while the character "." denotes an empty cell.

The following line contains an integer q ($1 \leq q \leq 300\,000$) — the number of queries. The k-th of the following q lines contains four integers $r_{A_k}, c_{A_k}, r_{B_k}, c_{B_k}$ ($1 \leq r_{A_k}, c_{A_k}, r_{B_k}, c_{B_k} \leq n$) — the row number and column number of cells A_k and B_k respectively. Cell A_k will be different than the cell B_k. Also, both cells will always be empty.

Output

Output q lines. The k-th line should contain a single integer s_k — the size of the largest crate that can be moved from A_k to B_k. If no crate can be moved from A_k to B_k then s_k should be 0.

Example

input	output
7 .....#. ...#.#. ....#.. ....### ....#.. #...... .... 5 2 5 5 2 2 5 3 6 2 2 6 3 2 2 6 6 1 1 7 7	1 0 3 1 1