

Castle on the Grid

You are given a square grid with some cells open (.) and some blocked (X). Your playing piece can move along any row or column until it reaches the edge of the grid or a blocked cell. Given a grid, a start and a goal, determine the minimum number of moves to get to the goal.

Example.

```
grid = ['...', '.X.', '...']
startX = 0
startY = 0
goalX = 1
goalY = 2
```

The grid is shown below:

```
...
.X.
...
```

The starting position ($startX, startY = (0, 0)$) so start in the top left corner. The goal is ($goalX, goalY = (1, 2)$). The path is $(0, 0) \rightarrow (0, 2) \rightarrow (1, 2)$. It takes **2** moves to reach the goal.

Function Description

Complete the *minimumMoves* function in the editor.

minimumMoves has the following parameter(s):

- *string grid[n]*: an array of strings that represent the rows of the grid
- *int startX*: starting X coordinate
- *int startY*: starting Y coordinate
- *int goalX*: ending X coordinate
- *int goalY*: ending Y coordinate

Returns

- *int*: the minimum moves to reach the goal

Input Format

The first line contains an integer n , the size of the array *grid*.

Each of the next n lines contains a string of length n .

The last line contains four space-separated integers, *startX*, *startY*, *goalX*, *goalY*

Constraints

- $1 \leq n \leq 100$

- $0 \leq \textit{startX}, \textit{startY}, \textit{goalX}, \textit{goalY} < n$

Sample Input

```
STDIN      FUNCTION
-----
3          grid[] size n = 3
.X.       grid = ['.X.', '.X.', '...']
.X.
...
0 0 0 2    startX = 0, startY = 0, goalX = 0, goalY = 2
```

Sample Output

```
3
```

Explanation

Here is a path that one could follow in order to reach the destination in **3** steps:

$(0,0) \rightarrow (2,0) \rightarrow (2,2) \rightarrow (0,2)$.