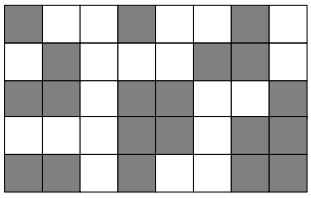
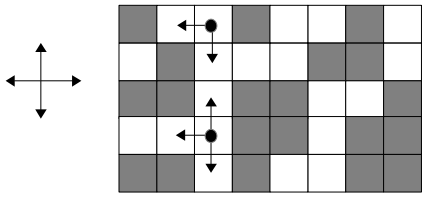
Problem B
Radioactive

Time Limit: 1 second.

There is an ancient facility recently found underground. For some unknown reason, the facility is entirely polluted with radioactive waste, so anyone planning to visit the facility must wear a protective suit. From the supersonic scanner, you have obtained its map, which clearly looks like a rectangular maze. The width of the facility is **M** units, and the length is **N** units. The facility map has **MN** cells (arranged into **M** rows and **N** columns). An example is shown below in Figure (a).



(a)



(b)

Each cell can be an empty cell (shown in an example above in white) or a brick cell (shown in black). From an empty cell, you can go to adjacent empty cell in 4 directions. Figure (b) shows 4 directions that you can go. It also shows possible moving directions from two of the empty cells. You cannot go out of the maze.

The facility has been designed very carefully so that from each empty cell, there is ***at most*** one way to reach another empty cell without visiting any cell more than once. Note that it might be impossible to reach some of the empty cell from some other empty cell.

For the sake of scientific revolution, you have to explore this facility. Your suit protection is good enough for **L** minutes. You will be teleported into the facility at some cell in the facility and you have to exit the facility using an exit teleporter located at some other cell. It takes you 1 minute to move from one cell to another adjacent cell.

Your goal is to visit ***as many*** cells as possible. To make sure you can publish your work while you are alive, you must reach the exit teleporter before your suit's protection power runs out.

Note that we are interested in ***the number of different cells you visit***, not the number of times you visit the cells. Also, note that when you are teleported to the facility, you arrived at some cell; clearly that cell has been visited.

Input
The first line contains an integer **T** denoting the number of test cases (**1 <= T <= 20**). Then, **T** test cases follow under the following format.

The first line contains 3 integers **M**, **N**, and **L**. (**2 <= M <= 30; 2 <= N <= 30; 1 <= L <= 900**)

The next **M** lines represent the map. Line **1+i** represents the **i**-th row of the map. Each line contains a string of length **N**. The j-th character in the string can be either '.' (representing an empty cell) or '#' (representing a brick cell).

The last line contains 4 integers **A**, **B**, **C**, and **D** denoting your starting location and the location of the exit teleporter. The cell you will be teleported to is in row **A**, column **B**. The exit teleporter is in a cell in row **C** column **D**. It is guaranteed that the exit teleporter is not in the cell you start.

Output

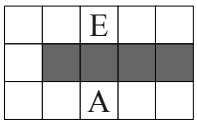
For each test case, your program should write one integer, the maximum number of cells you can visit so that you reach the exit teleporter within **L** minutes. If there is no way you can reach the teleporter within the limit, the program should write **-1**.

Example

Input	Output
3 5 8 9 #..#..#. .#...##. ##.##..# ...##.## ##.#..## 4 3 2 4 3 5 2 ..... .#### ..... 3 3 1 3 4 5 100 ..... .#### ...#. #.##. 3 3 1 3	7 -1 10

Remarks:

The map for case 2 is shown below. You start at the cell marked with A; the exit teleporter is in the cell marked with E.



The map for case 3 is shown below. Again, the starting cell is marked with A and the exit teleporter is at the cell marked with E.