
King of Maze

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Fish is extremely good at maze game and he becomes king of maze. This time he turns to build a tricky maze game and invite Ruins to solve.

The maze can be simply treated as a $N \times M$ grid, and some cells inside are walls where one can not move into. In this game, Ruins starts from one cell and he can move into any four-connected neighbor cells if empty. The length of a path in the maze is defined as the number of moves he have to do from the start to the end and his goal is to escape from the maze. As he is a smart guy, he always uses the shortest path strategy: he will always choose the shortest path from current cell to the exit, and moves to the next cell in this path. If there are multiple cells available, he will consider the cell in the top $(x - 1, y)$, underneath $(x + 1, y)$, in the left $(x, y - 1)$, in the right $(x, y + 1)$ in order and moves to the available one (in one of shortest path) if he is currently in (x, y) .

Fish's maze is a little different: besides the exit, walls and empty cells in all maze games, there is a different kind of cells: lift, which is controllable so that he can change cell of this kind into empty cell or into wall every time he wants. To make this game more interesting, before Ruins's next move, Fish will change some lift cells (or none) into walls and let the remainders become empty cells and wait for Ruins's decision. Of course, if Ruins is standing on a lift cell, Fish can not change this cell into wall.

Just help Fish to find a strategy to trap Ruins in the maze as long as possible!

Input

The first line of input contains an integer T , representing the number of test cases.

Then for each test case:

The first line contains three integers N, M, Q , representing the number of rows of the maze, the number of columns of the maze and the number of start cells you have to deal with.

Then N lines follow, the i -th line containing a string of length M representing a row in the maze, the j -th of which represents the cell of (i, j) . In the maze, "." represents empty cell, "#" represents wall, "?" represents lift cell and "E" represent the exit. It is guaranteed that there is exactly one "E" in a maze.

Then Q lines follow, each line containing two integers x, y which is the start cell of Ruins in one game for given maze, the start cell will never be a wall.

Output

For each test case, you should output **Case x :** first, where x indicates the case number starting from 1. Then Q lines follow, each line containing one integer representing the longest time you can trap Ruins in the maze for the given start cell. If you can trap him forever, output **-1**.

Example

standard input	standard output
2	Case 1:
4 5 4	2
..E..	5
.....	6
.?#?.	-1
.....	Case 2:
1 1	3
4 1	6
4 2	-1
4 3	-1
4 7 6	-1
...E...	-1
.....	
.??#??.	
.....	
1 1	
4 1	
4 2	
4 3	
4 4	
4 5	

Note

$$1 \leq T \leq 100$$

$$1 \leq N, M \leq 50$$

Denote the number of “?” in a maze as K , $0 \leq K \leq 10$

$$1 \leq Q \leq N \times M$$

$$1 \leq x \leq N$$

$$1 \leq y \leq M$$

For 90% test cases: $\max(N, M) \leq 10$