The task: You are to program a toy autonomous vehicle to drive from S to E on a map filled with obstacles:

The map	0	1	2	3	4	5	6	7	8	9
0	S	X				X	X	E		X
1		X	X			X	X		X	x
2		X	X				X			X
3			X			X	X		X	X
4					X	X				
5					X	X				
6	X									
7	X	X	X						X	X

You can make the car move forward, backward, left or right at any given position on the map, moving one cell with each move. Cells with obstacles cannot be traversed.

Input: a n x m numpy array representing a map, an example of which is shown above, where n < 20 and m < 20. The indices for the columns are the 'x' coordinates, and rows the 'y' coordinates. There are 4 possible characters that the array elements can take:

- 'X': obstacle
- 'S': start position
- 'E': end position
- '': drivable space

Output: a list of tuples, representing the (x, y) positions on the map, for any path that leads from position 'S' to 'E', in chronological order. i.e.

[(0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (1, 4), (2, 4), (3, 4), (3, 5) …]

There could be many solutions for a given map. Your program should return just one (any) that fulfills the criteria. If no path is feasible, return None