**DFS**

import copy

pacman\_x, pacman\_y = list(map(int, input().split()))

food\_x, food\_y = list(map(int, input().split()))

n, m = list(map(int, input().split()))

grid = []

node\_expanded = []

stack = []

answer\_routes = None

for i in range(0, n):

grid.append(list(map(str, input())))

directions = [[-1, 0], [0, -1], [0, 1], [1, 0]]

stack.append([pacman\_x, pacman\_y, []])

while len(stack) > 0:

x, y, r = stack.pop()

routes = copy.deepcopy(r)

routes.append([x, y])

node\_expanded.append([x, y])

if x == food\_x and y == food\_y:

if answer\_routes == None:

answer\_routes = routes

break

for direction in directions:

next\_x, next\_y = x + direction[0], y + direction[1]

if next\_x < 0 or next\_x >= n or next\_y < 0 and next\_y >= n:

continue

if grid[next\_x][next\_y] == "-" or grid[next\_x][next\_y] == ".":

grid[next\_x][next\_y] = '='

stack.append([next\_x, next\_y, routes])

print(str(len(node\_expanded)))

for point in node\_expanded:

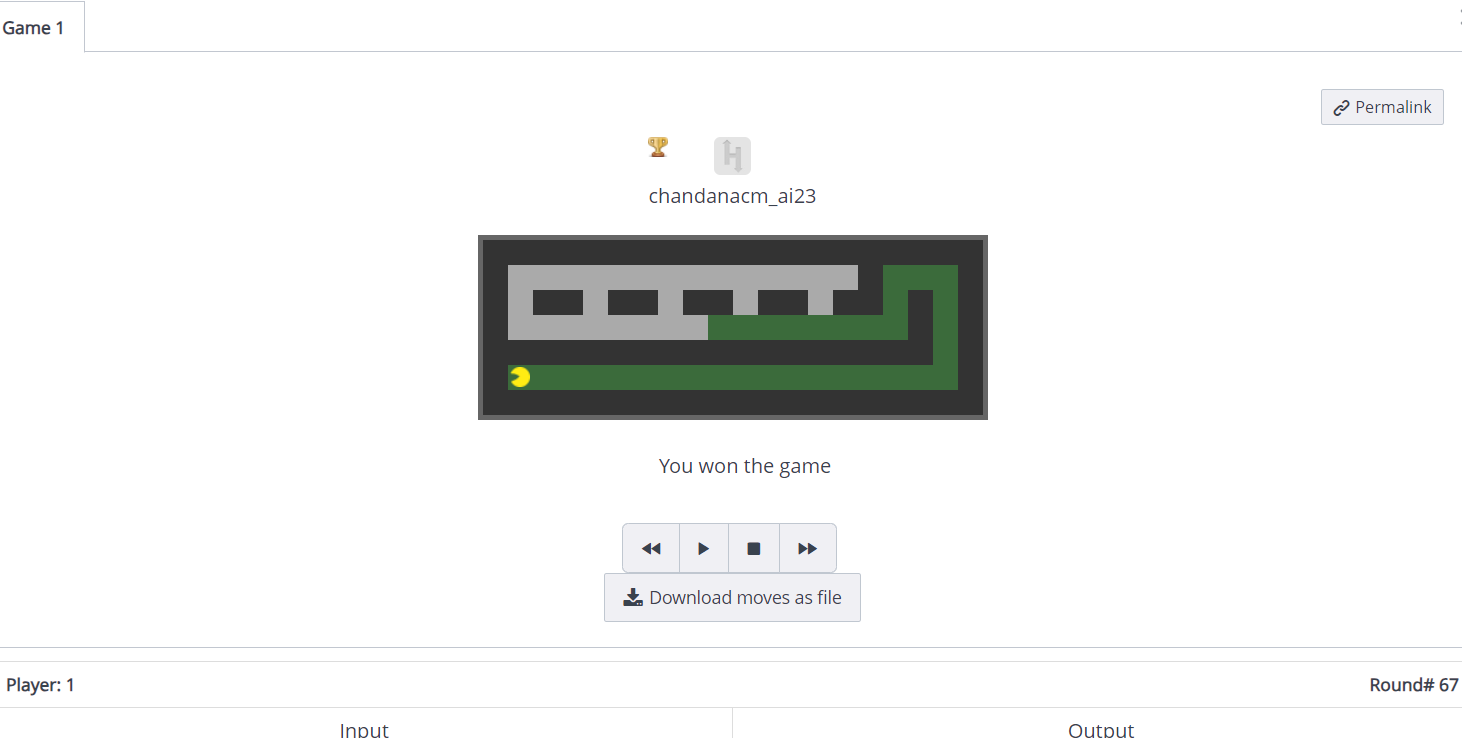
print(str(point[0]) + " " + str(point[1]))

print(str(len(answer\_routes) - 1))

for point in answer\_routes:

print(str(point[0]) + " " + str(point[1]))

**Output**

****