### Task-1:

"""

Input:

N N N Y Y N N

N Y N N Y Y N

Y Y N Y N N Y

N N N N N Y N

Y Y N N N N N

N N N Y N N N

"""

matrix = []

with open('input1.txt') as file:

for i in file:

lst = [x.strip() for x in i.split()]

matrix.append(lst)

print("Input - Task 1")

print(matrix)

rows = len(matrix)

columns = len(matrix[0])

dRow = [-1, 1, 0, 0, -1, -1, 1, 1]

dCol = [0, 0, -1, 1, 1, -1, 1, -1]

visited = []

def DFS\_util(row, column):

if (row, column) in visited:

return

visited.append((row, column))

for i in range(8):

if row + dRow[i] >=0 and row + dRow[i] < len(matrix) and column + dCol[i] >=0 and column + dCol[i] < len(matrix[0]) and matrix[row + dRow[i]][column + dCol[i]] == 'Y' and (row + dRow[i], column + dCol[i]) not in visited:

maximum[0] += 1

DFS\_util(row + dRow[i], column + dCol[i])

maximum1 = -1

maximum = [0]

for i in range(len(matrix)):

maximum[0] = 1

for j in range(len(matrix[0])):

if matrix[i][j] == 'Y' and (i, j) not in visited:

DFS\_util(i, j)

maximum1 = max(maximum1, maximum[0])

print("Output - Task 1")

print("The number of infected people in the biggest region: ", maximum1)

print()

"""

=========================================================================

"""

### Task-2:

"""

Input:

5

4

A H T H

H H T A

T T H H

A H T H

H T H H

"""

import itertools

matrix = []

with open('task2\_input1.txt') as file:

for i in file:

lst = [x.strip() for x in i.split()]

matrix.append(lst)

i, j = 0, 0

row = int(matrix[i][j])

column = int(matrix[i+1][j])

matrix.pop(0)

matrix.pop(0)

R = [0, 0, 1, -1]

C = [1, -1, 0, 0]

visited = [[False] \* column for i in range(row)]

x = []

y = []

mins = 0

for i in range(row):

for j in range(column):

if matrix[i][j] == 'H':

x.append((i, j))

if matrix[i][j] == 'A':

y.append([i, j, mins])

def BFS(matrix, x, y, visited):

while y:

temp = y.pop(0)

i = temp[0]

j = temp[1]

k = temp[2]

visited[i][j] = True

for (r, c) in itertools.zip\_longest(R, C):

if (i + r) >= 0 and (i + r) < row and (j + c) >= 0 and (j + c) < column and matrix[i + r][j + c] == 'H' and visited[i][j] == True :

matrix[i + r][j + c] = 'A'

visited[i + r][j + c] = True

y.append([i + r, j + c, (k + 1)])

x.remove((i + r, j + c))

return k

mins = BFS(matrix, x, y, visited)

print("Input - Task 2")

print(matrix)

print("Output - Task 2")

print("Time: ", mins, "minutes")

survivors = 0

for i in range(row):

for j in range(column):

if matrix[i][j] == 'H':

survivors += 1

if survivors == 0:

print("No one survived")

else:

print(survivors, "survived")