# -\*- coding: utf-8 -\*-

"""08\_18101501\_MDFahimIslam

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/11YqgT8SQCBzD7O4m-cxLY5Rzf2I1Rbj6

#Question 1

"""

inputFile = open('/content/drive/MyDrive/Fall2021/CSE422/Lab1 Inputs/Question1input1.txt','r')

# inputFile = open('/content/drive/MyDrive/Fall2021/CSE422/Lab1 Inputs/Question1input2.txt','r')

lines = inputFile.readlines()

print(len(lines))

adj\_matrix = []

row=len(lines)

column=0

count =0

for line in lines:

line = line.strip()

column = len(line.split(' '))

adj\_matrix.append(line.split(' '))

print(line.strip())

print(row,'X',column)

adj\_matrix

totalYInRegion = []

def dfs(startingR,startingC):

global count

global adj\_matrix

stack = []

stack.append([startingR,startingC])

count += 1

while len(stack) !=0:

# print(len(stack))

# print(stack)

r,c = stack[len(stack)-1][0],stack[len(stack)-1][1]

# print(r,c)

adj\_matrix[r][c] = 'N'

stack.pop()

if r>=0 and r<row and (c+1)>=0 and (c+1)<column:

if adj\_matrix[r][c+1] == 'Y':

stack.append([r,c+1])

adj\_matrix[r][c+1] = 'N'

count += 1

if r>=0 and r<row and (c-1)>=0 and (c-1)<column:

if adj\_matrix[r][c-1] == 'Y':

stack.append([r,c-1])

adj\_matrix[r][c-1] = 'N'

count += 1

if (r+1)>=0 and (r+1)<row and c>=0 and c<column:

if adj\_matrix[r+1][c] == 'Y':

stack.append([r+1,c])

adj\_matrix[r+1][c] = 'N'

count += 1

if (r-1)>=0 and (r-1)<row and c>=0 and c<column:

if adj\_matrix[r-1][c] == 'Y':

stack.append([r-1,c])

adj\_matrix[r-1][c] = 'N'

count += 1

if (r+1)>=0 and (r+1)<row and (c+1)>=0 and (c+1)<column:

if adj\_matrix[r+1][c+1] == 'Y':

stack.append([r+1,c+1])

adj\_matrix[r+1][c+1] = 'N'

count += 1

if (r+1)>=0 and (r+1)<row and (c-1)>=0 and (c-1)<column:

if adj\_matrix[r+1][c-1] == 'Y':

stack.append([r+1,c-1])

adj\_matrix[r+1][c-1] = 'N'

count += 1

# print(count)

totalYInRegion.append(count)

count=0

global adj\_matrix

startRow = 0

startColumn = 0

isFound = False

for i in range(0,row):

for j in range(0,column):

if adj\_matrix[i][j] == 'Y':

dfs(i,j)

"""###Output Of Question 1"""

max(totalYInRegion)

"""#Question 2 """

inputFile = open('/content/drive/MyDrive/Fall2021/CSE422/Lab1 Inputs/Question2 input1.txt','r')

# inputFile = open('/content/drive/MyDrive/Fall2021/CSE422/Lab1 Inputs/Question2 input2.txt','r')

row = int(inputFile.readline())

column = int(inputFile.readline())

lines = inputFile.readlines()

matrix = []

print(row,column)

for line in lines:

line = line.strip()

column = len(line.split(' '))

matrix.append(line.split(' '))

print(line.strip())

for m in matrix:

print(m)

queue = []

times = 0

def select\_initial\_Aliens():

global queue

for i in range(0,row):

for j in range(0,column):

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

if i>=0 and i<row and (j-1)>=0 and (j-1)<column and matrix[i][j-1] == 'H':

queue.append([i,j])

elif i>=0 and i<row and (j+1)>=0 and (j+1)<column and matrix[i][j+1] == 'H':

queue.append([i,j])

elif (i+1)>=0 and (i+1)<row and j>=0 and j<column and matrix[i+1][j] == 'H':

queue.append([i,j])

elif (i-1)>=0 and (i-1)<row and j>=0 and j<column and matrix[i-1][j] == 'H':

queue.append([i,j])

print(queue)

select\_initial\_Aliens()

def runBFS():

global times

global queue

while len(queue) != 0:

leafNodeList = []

while len(queue) != 0:

v = queue.pop(0)

if v[0]>=0 and v[0]<row and (v[1]-1)>=0 and (v[1]-1)<column and matrix[v[0]][v[1]-1] == 'H':

matrix[v[0]][v[1]-1] = 'A'

i = v[0]

j = v[1]-1

if i>=0 and i<row and (j-1)>=0 and (j-1)<column and matrix[i][j-1]== 'H':

leafNodeList.append([i,j-1])

if i>=0 and i<row and (j+1)>=0 and (j+1)<column and matrix[i][j+1]== 'H':

leafNodeList.append([i,j+1])

if i+1>=0 and i+1<row and j>=0 and j<column and matrix[i+1][j]== 'H':

leafNodeList.append([i+1,j])

if i-1>=0 and i-1<row and j>=0 and j<column and matrix[i-1][j]== 'H':

leafNodeList.append([i-1,j])

if v[0]>=0 and v[0]<row and (v[1]+1)>=0 and (v[1]+1)<column and matrix[v[0]][v[1]+1] == 'H':

matrix[v[0]][v[1]+1] = 'A'

i = v[0]

j = v[1]+1

if i>=0 and i<row and (j-1)>=0 and (j-1)<column and matrix[i][j-1]== 'H':

leafNodeList.append([i,j])

if i>=0 and i<row and (j+1)>=0 and (j+1)<column and matrix[i][j+1]== 'H':

leafNodeList.append([i,j])

if i+1>=0 and i+1<row and j>=0 and j<column and matrix[i+1][j]== 'H':

leafNodeList.append([i,j])

if i-1>=0 and i-1<row and j>=0 and j<column and matrix[i-1][j]== 'H':

leafNodeList.append([i,j])

if (v[0]+1)>=0 and (v[0]+1)<row and v[1]>=0 and v[1]<column and matrix[v[0]+1][v[1]] == 'H':

matrix[v[0]+1][v[1]] = 'A'

i = v[0]+1

j = v[1]

if i>=0 and i<row and (j-1)>=0 and (j-1)<column and matrix[i][j-1]== 'H':

leafNodeList.append([i,j])

if i>=0 and i<row and (j+1)>=0 and (j+1)<column and matrix[i][j+1]== 'H':

leafNodeList.append([i,j])

if i+1>=0 and i+1<row and j>=0 and j<column and matrix[i+1][j]== 'H':

leafNodeList.append([i,j])

if i-1>=0 and i-1<row and j>=0 and j<column and matrix[i-1][j]== 'H':

leafNodeList.append([i,j])

if (v[0]-1)>=0 and (v[0]-1)<row and v[1]>=0 and v[1]<column and matrix[v[0]-1][v[1]] == 'H':

matrix[v[0]-1][v[1]] = 'A'

i = v[0]-1

j = v[1]

if i>=0 and i<row and (j-1)>=0 and (j-1)<column and matrix[i][j-1]== 'H':

leafNodeList.append([i,j])

if i>=0 and i<row and (j+1)>=0 and (j+1)<column and matrix[i][j+1]== 'H':

leafNodeList.append([i,j])

if i+1>=0 and i+1<row and j>=0 and j<column and matrix[i+1][j]== 'H':

leafNodeList.append([i,j])

if i-1>=0 and i-1<row and j>=0 and j<column and matrix[i-1][j]== 'H':

leafNodeList.append([i,j])

for m in matrix:

print(m)

#leafNodeList.append([v[0]-1,v[1]])

times += 1

for leaf in leafNodeList:

queue.append(leaf)

runBFS()

for m in matrix:

print(m)

"""###Output Of Question 2"""

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

count=0

for m in matrix:

for h in m:

if h=='H':

count+=1

if count == 0:

print("No one survived")

else:

print(count," survived ")