class Solution:

"""

@param grid: Given a 2D grid, each cell is either 'W', 'E' or '0'

@return: an integer, the maximum enemies you can kill using one bomb

"""

def maxKilledEnemies(self, grid):

# write your code here

if not grid:

return 0

if not grid[0]:

return 0

def upf(grid):

m,n = len(grid),len(grid[0])

up = [[0 for i in range(n)] for i in range(m)]

for i in range(n):

if grid[0][i] == 'E':

up[0][i] = 1

for i in range(1,m):

for j in range(n):

a=0

if grid[i][j] == 'E':

a = up[i-1][j]+1

elif grid[i][j] =='W':

a = 0

else:

a = up[i-1][j]

up[i][j] = a

return up

def downf(grid):

m,n = len(grid),len(grid[0])

down = [[0 for i in range(n)] for i in range(m)]

for i in range(n):

if grid[-1][i] == 'E':

down[-1][i] = 1

for i in range(m-2,-1,-1):

for j in range(n):

a=0

if grid[i][j] == 'E':

a = down[i+1][j]+1

elif grid[i][j] =='W':

a = 0

else:

a = down[i+1][j]

down[i][j] = a

return down

def leftf(grid):

m,n = len(grid),len(grid[0])

left = [[0 for i in range(n)] for i in range(m)]

for i in range(m):

if grid[i][0] == 'E':

left[i][0] = 1

for i in range(m):

for j in range(1,n):

a=0

if grid[i][j] == 'E':

a = left[i][j-1]+1

elif grid[i][j] =='W':

a = 0

else:

a = left[i][j-1]

left[i][j] = a

return left

def rightf(grid):

m,n = len(grid),len(grid[0])

right = [[0 for i in range(n)] for i in range(m)]

for i in range(m):

if grid[i][-1] == 'E':

right[i][-1] = 1

for i in range(m):

for j in range(n-2,-1,-1):

a=0

if grid[i][j] == 'E':

a = right[i][j+1]+1

elif grid[i][j] =='W':

a = 0

else:

a = right[i][j+1]

right[i][j] = a

return right

m,n = len(grid),len(grid[0])

up = upf(grid)

down = downf(grid)

left = leftf(grid)

right = rightf(grid)

dp = [[0 for i in range(n)] for i in range(m)]

for i in range(m):

for j in range(n):

if grid[i][j]=='0':

dp[i][j] = up[i][j]+down[i][j]+left[i][j]+right[i][j]

#print up

#print down

#print left

#print right

maximum = float("-inf")

for i in range(m):

for j in range(n):

maximum = max(maximum,dp[i][j])

return maximum