1、

def solution(n):

if(n < 1):

return 0

if(n == 1):

return 1

if(n == 2):

return 2

if(n == 5):

return 9

if(n == 10):

return 129

return solution(n-1) + solution(n-2) + solution(n-5) + solution(n-10)

n = int(input())

print(solution(n))

2、

import numpy as np

dx = [0,0,1,-1]

dy = [1,-1,0,0]

def dfs(x,y,m,n,result,flag,maze):

flag[x][y] = 1

if(maze[x][y] == 'T'):

return True

for i in range(4):

nx = x+dx[i]

ny = y+dy[i]

if nx>0 and ny>0 and nx<m and ny<n and flag[nx][ny] == 0 or maze[nx][ny] == '0' or maze[nx][ny] == 'T':

result.append([x,y])

return dfs(nx,ny,m,n,result,flag,maze)

flag[nx][ny] = 0

return False

if \_\_name\_\_ == "\_\_main\_\_":

m = 3

n = 3

maze = [['0', '0', '0'],['\*', '\*', '0'], ['\*', '0', '0']]

maze[m-1][n-1] = 'T'

flag = np.zeros((m,n))

result = []

dfs(0,0,m,n,result,flag,maze)

print(result)

3、