一開始使用矩陣存取圖案，再丟到map裡面，使用access()函數，對細胞進行操作，根據遊戲規則讓細胞進行轉化，再放入newmap裡面，最後再print矩陣

規則: 相鄰細胞只有2~3個才會活，有3個會生出細胞

2) 跑其後續的變化: 我設計的金和大，跑了很多次都沒有消失(以下是實作



經過了 1000 代，很明顯的看到，細胞在穩定狀態，細胞以經不會動作了



而金，則不會達到穩定狀態，但是經過了好幾千代，他還存活!!

3) 生命細胞遊戲，細胞最後會消失? 為甚麼?

因為根據規則，細胞旁邊要有2~3個才會活，而細胞又會生成細胞，細胞過多的時候，會擁擠死，死到一定程度就會孤單死，只有在穩定狀態且沒有外在細胞干擾的情況，細胞在會存活

以下是我的程式碼

# 生命細胞遊戲實作  
# File Name: lifeGame.py  
# Version 4.0 (Updated on May 8, 2021)  
import random  
  
MAXROW = 10  
MAXCOL = 25  
DEAD = 0  
ALIVE = 1  
map = [[DEAD for col in range(MAXCOL)] for row in range(MAXROW)]  
newmap = [[ALIVE for col in range(MAXCOL)] for row in range(MAXROW)]  
generation = 0  
  
big\_cell = [  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
]  
chin\_cell = [  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0],  
]  
king\_cell = [  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
]  
  
  
def rand\_cell(m):  
 for x in range(len(m)):  
 for y in range(len(m[x])):  
 a = random.randrange(1, 10, 1)  
 if a == 1:  
 m[x][y] = ALIVE  
  
  
def set\_cell():  
 global map  
 print('\n\n 生命細胞遊戲: ')  
 print('相鄰細胞只有2~3個才會活，有3個會生出細胞')  
 print('==============================')  
 print('\*\*\*\*\* 選擇下面選項輸入: \*\*\*\*\*\*\*')  
 print(' <1> 金 細胞 ')  
 print(' <2> 大 細胞 ')  
 print(' <3> 慨影細胞 ')  
 print(' <4> 隨機生成細胞 ')  
 print(' <5> 自定義細胞 ')  
 print('==============================')  
 try:  
 option = int(input(' Choice : '))  
 except ValueError:  
 print('Not a correct number.')  
 print('Try again\n')  
  
 print()  
 if option == 1:  
 map = chin\_cell  
 elif option == 2:  
 map = big\_cell  
 elif option == 3:  
 map = king\_cell  
 elif option == 4:  
 rand\_cell(map)  
 elif option == 5:  
 init()  
 else:  
 print('不正確的選項')  
  
  
def init():  
 global map  
  
 row = 0  
 col = 0  
 print('\n\nx,y位置: 0 <= x <= %d, 0 <= y <= %d' % (MAXROW - 1, MAXCOL - 1))  
 print('輸入-1就跳出輸入')  
  
 # 輸入活細胞之位置，以(-1, -1)結束輸入  
 while True:  
 row = int(input('x-->'))  
 if row == -1: break  
 col = int(input('y-->'))  
 if col == -1: break  
 if (0 <= row and row < MAXROW and 0 <= col and col < MAXCOL):  
 map[row][col] = ALIVE  
 elif row == -1 and col == -1:  
 print('Input is terminated')  
 else:  
 print('(x, y) exceeds map range!')  
  
  
def neighbors(row, col):  
 global map  
  
 count = 0  
 # 計算每一個cell的鄰居個數  
 # 因為cell本身亦被當做鄰居計算  
 # 故最後還要調整  
  
 for r in range(row - 1, row + 2):  
 for c in range(col - 1, col + 2):  
 # if r < 0 or r >= MAXROW or c < 0 or c >= MAXCOL:  
 # continue  
 if r < 0: r = MAXROW - 1  
 if r >= MAXROW: r = 0  
 if c < 0: c = MAXCOL - 1  
 if c >= MAXCOL: c = 0  
 if map[r][c] == ALIVE:  
 count += 1  
  
 # 調整鄰居個數  
 if map[row][col] == ALIVE:  
 count -= 1  
 return count  
  
  
# 顯示目前細胞狀態  
def output\_map():  
 global generation  
  
 space = ' '  
 print(space, '\nGame of life cell status')  
 generation += 1  
 print('------Generation %d------' % (generation))  
 for row in range(MAXROW):  
 print()  
 print(space)  
 for col in range(MAXCOL):  
 if map[row][col] == ALIVE:  
 print('@', end='')  
 else:  
 print('-', end='')  
  
  
def access():  
 global newmap  
  
 ans = 'y'  
 while ans == 'y' or ans == 'f':  
 # 計算每一個(row, col)之cell的鄰居個數  
 # 依此個數決定其下一代是生是死。  
 # 將下一代的map\_暫存在newmap以防overwrite map\_。  
 for row in range(MAXROW):  
 for col in range(MAXCOL):  
 if neighbors(row, col) == 0 \  
 or neighbors(row, col) == 1 \  
 or neighbors(row, col) == 4 \  
 or neighbors(row, col) == 5 \  
 or neighbors(row, col) == 6 \  
 or neighbors(row, col) == 7 \  
 or neighbors(row, col) == 8:  
 newmap[row][col] = DEAD  
 elif neighbors(row, col) == 2:  
 newmap[row][col] = map[row][col]  
 elif neighbors(row, col) == 3:  
 newmap[row][col] = ALIVE  
  
 copymap() # 將newmap copy to map\_  
  
 while True:  
 ans = input('\n\n next Generation (y) \n next 999 Generation (f) \n back to menu (m) \n leave game (n): ')  
 if ans == 'y' or ans == 'n' or ans == 'm' or ans == 'f':  
 break  
  
 if ans == 'y':  
 output\_map()  
 if ans == 'm':  
 set\_cell()  
 output\_map()  
 access()  
 if ans == 'f':  
 i = 0  
 while i < 999:  
 i += 1  
 output\_map()  
  
  
# 將newmap copy至map\_中  
def copymap():  
 global map\_  
  
 for row in range(MAXROW):  
 for col in range(MAXCOL):  
 map[row][col] = newmap[row][col]  
  
  
def main(): # 主函數  
 set\_cell() # 起始map  
 output\_map()  
 access()  
  
  
main()