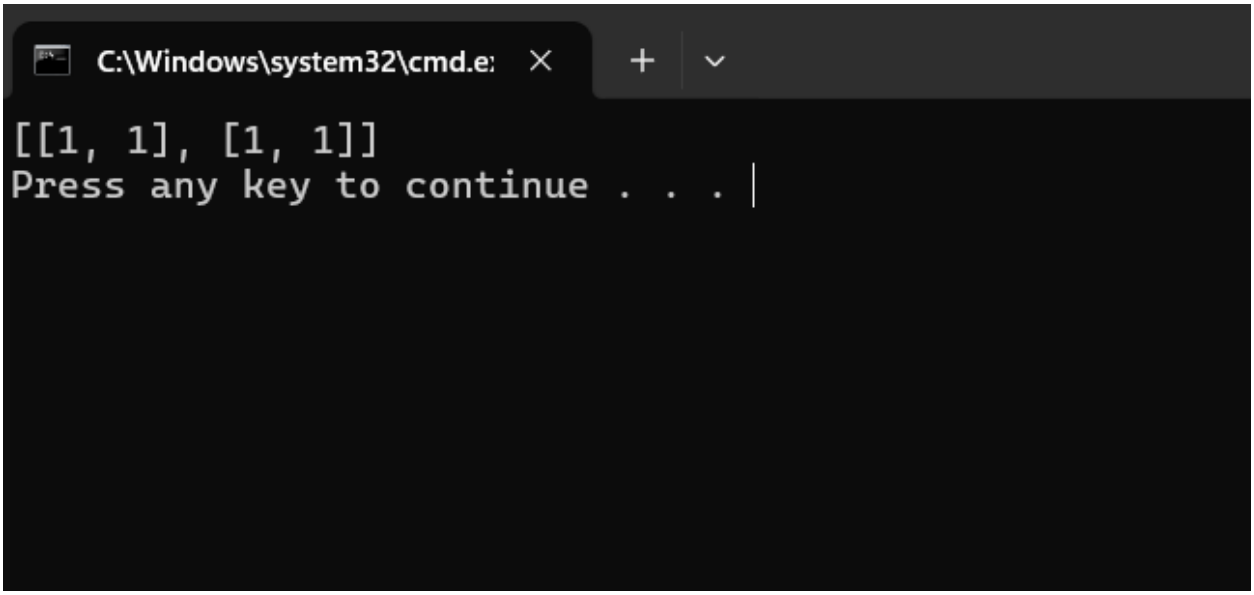
129)Game of life

CODE:

```
def game_of_life(board):
    m, n = len(board), len(board[0])
    next_board = [[0] * n for _ in range(m)]

    def count_live_neighbors(row, col):
        count = 0
        for i in range(-1, 2):
            for j in range(-1, 2):
                if i == 0 and j == 0:
                    continue
                if 0 <= row + i < m and 0 <= col + j < n and board[row + i][col + j]
== 1:
                    count += 1
        return count

    for i in range(m):
        for j in range(n):
            live_neighbors = count_live_neighbors(i, j)
            if board[i][j] == 1:
                if live_neighbors < 2 or live_neighbors > 3:
                    next_board[i][j] = 0
                else:
                    next_board[i][j] = 1
            else:
                if live_neighbors == 3:
                    next_board[i][j] = 1

    return next_board

# Example Usage
board = [[1,1],[1,0]]
next_state = game_of_life(board)
print(next_state)
```

OUTPUT:



TIME COMPLEXITY : O(m*n)