2025-10-19/p2Matrix.py

```python
# matrix with m rows and n columns
from random import random

m = 10
n = 15
matrix = [[int(random() * 100) for _ in range(n)] for _ in range(m)]  #init. with random numbers

# a) display the elements of the matrix row by row
for i in range(m):
    print("row {}: ".format(i), end = '')
    for j in range(n):
        print("{:2d} ".format(matrix[i][j]), end = '')
    print()
print()

# b) display the elements of the matrix column by column
for j in range(n):
    print("col {}: ".format(j), end = '')
    for i in range(m):
        print("{:2d} ".format(matrix[i][j]), end = '')
    print()
print()

# c) display the elements of the diagonal matrix and the parallel lines
maxim = max(m, n)   # create a super sqaure matrix
for j in range(maxim - 1, -maxim, -1):  # parse it in reverse
    for i in range(0, maxim):
        if 0 <= i < m and 0 <= i + j < n:
            print("{:2d} ".format(matrix[i][i + j]), end = '')
    print()

# d) display the elements in a spiral
k = m * n # number of total elements to display
step_i = 0      # step for i
step_j = 1      # step for j
min_i = 0       # minimum for i
max_i = m - 1   # maximum for i
min_j = 0       # minimum for j
max_j = n - 1   # maximum for j
i = 0           # starting position
j = 0           # starting position

while k > 0:
    # 1: print position
    print("{:2d} ".format(matrix[i][j]), end = '')

    # 2: count another printed position:
    k -= 1
```

```
49
50      # 3: move to the next position:
51      i = i + step_i
52      j = j + step_j
53
54      # 4: check the limits (to see if we're out of the matrix):
55      if j > max_j:   #checking the right limit
56          # 4.1: move back to the matrix:
57          j = max_j
58          i += 1
59          # 4.2: change the direction:
60          step_i = 1
61          step_j = 0
62          # 4.3: mark the completed row and change the limit:
63          min_i += 1
64
65      if i > max_i:   #checking the bottom limit
66          i = max_i
67          j -= 1
68          step_i = 0
69          step_j = -1
70          max_j -= 1
71
72      if j < min_j:   #checking the left limit
73          j = min_j
74          i -= 1
75          step_i = -1
76          step_j = 0
77          max_i -= 1
78
79      if i < min_i:   #checking the upper limit
80          i = min_i
81          j += 1
82          step_i = 0
83          step_j = 1
84          min_j += 1
85  print()
```