import random

def generar\_array(n):

# Genera una matriz cuadrada de tamaño NxN con números aleatorios entre 0 y 9.

array = []

for i in range(n):

fila = [random.randint(0, 9) for i in range(n)]

array.append(fila)

return array

def imprimir\_array(array):

# Imprime la matriz sin corchetes.

for fila in array:

print(\*fila)

def calcular\_suma\_filas(array):

# Calcula la suma de los elementos de cada fila y los almacena en una lista.

return [sum(fila) for fila in array]

def calcular\_suma\_columnas(array):

# Calcula la suma de los elementos de cada columna y los almacena en una lista.

n = len(array)

return [sum(array[i][j] for i in range(n)) for j in range(n)]

def imprimir\_sumas(sumas\_filas, sumas\_columnas):

# Calcula y muestra las sumas de cada fila y columna antes de las dos listas finales.

print("\nSuma de cada fila:")

for i, suma in enumerate(sumas\_filas):

print(f"Fila {i + 1}: {suma}")

print("\nSuma de cada columna:")

for i, suma in enumerate(sumas\_columnas):

print(f"Columna {i + 1}: {suma}")

print("\nLista de suma de cada fila:")

print("sumas\_filas =", sumas\_filas)

print("\nLista de suma de cada columna:")

print("sumas\_columnas =", sumas\_columnas)

def main():

# Incluya manejo de excepciones en caso de que el usuario ingrese un valor no válido para N

try:

n = int(input("Ingrese el tamaño de la matriz cuadrada (N): "))

except ValueError:

print("Error: Ingrese un número entero válido para N.")

return

if n <= 0:

print("Error: N debe ser un número entero positivo.")

return

array = generar\_array(n)

print("\nArray generada:")

imprimir\_array(array)

sumas\_filas = calcular\_suma\_filas(array)

sumas\_columnas = calcular\_suma\_columnas(array)

imprimir\_sumas(sumas\_filas, sumas\_columnas)

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

main()