import numpy as np

def input\_matrix():

rows = int(input("Enter the number of rows: "))

cols = int(input("Enter the number of columns: "))

matrix = []

print("Enter the elements row by row:")

for i in range(rows):

row = list(map(float, input(f"Enter row {i + 1}: ").split()))

matrix.append(row)

return np.array(matrix)

def display\_matrix(matrix):

print("\nMatrix:")

print(matrix)

def matrix\_addition(A, B):

return A + B

def matrix\_subtraction(A, B):

return A - B

def matrix\_multiplication(A, B):

return np.dot(A, B)

def matrix\_transpose(A):

return np.transpose(A)

def matrix\_determinant(A):

if A.shape[0] == A.shape[1]: # Check if it's a square matrix

return np.linalg.det(A)

else:

return "Determinant is not defined for non-square matrices."

def main():

print("Welcome to the Matrix Operations Tool!")

# Input matrices

print("\nInput Matrix A:")

A = input\_matrix()

print("\nInput Matrix B:")

B = input\_matrix()

# Display matrices

display\_matrix(A)

display\_matrix(B)

# Matrix Addition

print("\nMatrix A + Matrix B:")

try:

print(matrix\_addition(A, B))

except Exception as e:

print(f"Error in addition: {e}")

# Matrix Subtraction

print("\nMatrix A - Matrix B:")

try:

print(matrix\_subtraction(A, B))

except Exception as e:

print(f"Error in subtraction: {e}")

# Matrix Multiplication

print("\nMatrix A \* Matrix B:")

try:

print(matrix\_multiplication(A, B))

except Exception as e:

print(f"Error in multiplication: {e}")

# Matrix Transpose

print("\nTranspose of Matrix A:")

display\_matrix(matrix\_transpose(A))

print("\nTranspose of Matrix B:")

display\_matrix(matrix\_transpose(B))

# Matrix Determinant

print("\nDeterminant of Matrix A:")

print(matrix\_determinant(A))

print("\nDeterminant of Matrix B:")

print(matrix\_determinant(B))

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

main()