Q.6) Consider the 2 relations R & S where R->(5x6) S->(6x3), establish

a)

i)max_min,

ii)min_max,

iii)max_product,

iv)max_avg on composite relationship of R & s, considering the values of R & S randomly between 0-1.

b)Plot the graph of derived 4 principles.

```
import numpy as np

import matplotlib.pyplot as plt


# Generate random matrices R (5x6) and S (6x3) with values between 0 and 1

np.random.seed(42) #Setting seed for reproducibility

R = np.random.rand(5, 6)  #R (5x6)

S = np.random.rand(6, 3)  #S (6x3)


#a)

#i)

# Function for max-min composition

def max_min_composition(R, S):

   result = np.zeros((R.shape[0], S.shape[1]))  #result=resultant mat ;  R.shape = (5, 6) S.shape = (6, 3) => (R.shape[0], S.shape[1]) = (5, 3)  ; np.zeros function in NumPy is used to create a new array filled with zeros.

   for i in range(R.shape[0]):

      for j in range(S.shape[1]):

         result[i, j] = np.max(np.minimum(R[i, :], S[:, j]))  #R-> ith row, all cols(:)  ; S-> jth col,all rows(:)

   return result


#ii)
```

```
# Function for min-max composition
def min_max_composition(R, S):
    result = np.zeros((R.shape[0], S.shape[1]))
    for i in range(R.shape[0]):
        for j in range(S.shape[1]):
            result[i, j] = np.min(np.maximum(R[i, :], S[:, j]))
    return result

#iii)
# Function for max-product composition
def max_product_composition(R, S):
    result = np.zeros((R.shape[0], S.shape[1]))
    for i in range(R.shape[0]):
        for j in range(S.shape[1]):
            result[i, j] = np.max(R[i, :] * S[:, j])
    return result

#iv)
# Function for max-avg composition
def max_avg_composition(R, S):
    result = np.zeros((R.shape[0], S.shape[1]))
    for i in range(R.shape[0]):
        for j in range(S.shape[1]):
            result[i, j] = np.max((R[i, :] + S[:, j]) / 2)
    return result

# Calculate the compositions
max_min_result = max_min_composition(R, S)
min_max_result = min_max_composition(R, S)
max_product_result = max_product_composition(R, S)
max_avg_result = max_avg_composition(R, S)
```

```
# Print the results
print("R:\n", R)
print("\nS:\n", S)
print("\nMax-Min Composition:\n", max_min_result)
print("\nMin-Max Composition:\n", min_max_result)
print("\nMax-Product Composition:\n", max_product_result)
print("\nMax-Avg Composition:\n", max_avg_result)



#b)
# Plot the results separately
def plot_matrix(matrix, title):   #matrix: The matrix (a 2D NumPy array) that you want to visualize, title: A string that will be used as the title of the plot.
    plt.figure(figsize=(6, 5))
    plt.imshow(matrix, cmap='viridis', aspect='auto') #plt.imshow() is used to display the matrix as an image, cmap='viridis' Specifies the colormap to use. viridis is a popular color map , aspect='auto': Automatically adjusts the aspect ratio of the plot to fit the figure.
    plt.title(title)
    plt.colorbar()  #Adds a colorbar to the side of the plot
    plt.show()


plot_matrix(max_min_result, "Max-Min Composition")
plot_matrix(min_max_result, "Min-Max Composition")
plot_matrix(max_product_result, "Max-Product Composition")
plot_matrix(max_avg_result, "Max-Avg Composition")
```

```
R:
 [[0.37454012 0.95071431 0.73199394 0.59865848 0.15601864 0.15599452]
 [0.05808361 0.86617615 0.60111501 0.70807258 0.02058449 0.96990985]
 [0.83244264 0.21233911 0.18182497 0.18340451 0.30424224 0.52475643]
 [0.43194502 0.29122914 0.61185289 0.13949386 0.29214465 0.36636184]
 [0.45606998 0.78517596 0.19967378 0.51423444 0.59241457 0.04645041]]

S:
 [[0.60754485 0.17052412 0.06505159]
 [0.94888554 0.96563203 0.80839735]
 [0.30461377 0.09767211 0.68423303]
 [0.44015249 0.12203823 0.49517691]
 [0.03438852 0.9093204  0.25877998]
 [0.66252228 0.31171108 0.52006802]]
```

```
Max-Min Composition:
 [[0.94888554 0.95071431 0.80839735]
 [0.86617615 0.86617615 0.80839735]
 [0.60754485 0.31171108 0.52006802]
 [0.43194502 0.31171108 0.61185289]
 [0.78517596 0.78517596 0.78517596]]

Min-Max Composition:
 [[0.15601864 0.31171108 0.25877998]
 [0.03438852 0.17052412 0.06505159]
 [0.30424224 0.18182497 0.30424224]
 [0.29214465 0.13949386 0.29214465]
 [0.30461377 0.19967378 0.45606998]]
```

```
Max-Product Composition:
 [[0.90211906 0.91804019 0.76855492]
 [0.82190202 0.83640743 0.7002145 ]
 [0.50574624 0.27665368 0.27290904]
 [0.27634312 0.28122019 0.41864996]
 [0.74504211 0.75819106 0.63473416]]

Max-Avg Composition:
 [[0.94979992 0.95817317 0.87955583]
 [0.90753084 0.91590409 0.83728675]
 [0.71999375 0.60678132 0.52241223]
 [0.62005734 0.62843059 0.64804296]
 [0.86703075 0.875404   0.79678665]]
```