**1. Write a Python Program to add two Matrices?**

I/P:

**def** addMatrices(a,b):

print(f'Inputs: {a},{b}')

**if** len(a) **==** len(b):

out\_matrix **=** []

**for** ele **in** range(len(a)):

**if** len(a[ele]) **==** len(b[ele]):

out\_matrix**.**append([])

**for** sub\_ele **in** range(len(a[ele])):

out\_matrix[ele]**.**append(a[ele][sub\_ele]**+**b[ele][sub\_ele])

**else**:

print('Both Matrices must contains same no of rows and columns')

**else**:

print('Both Matrices must contains same no of rows and columns')

print(f'Output: {out\_matrix}')

addMatrices([[1,2,3],[4,5,6],[7,8,9]],[[9,8,7],[6,5,4],[3,2,1]])

addMatrices([[2,3,5],[1,1,1],[2,2,2]],[[4,3,5],[1,2,3],[3,2,1]])

Inputs: [[1, 2, 3], [4, 5, 6], [7, 8, 9]],[[9, 8, 7], [6, 5, 4], [3, 2, 1]]

Output: [[10, 10, 10], [10, 10, 10], [10, 10, 10]]

Inputs: [[2, 3, 5], [1, 1, 1], [2, 2, 2]],[[4, 3, 5], [1, 2, 3], [3, 2, 1]]

Output: [[6, 6, 10], [2, 3, 4], [5, 4, 3]]

**2. Write a Python Program to multiply two Matrices?**

I/P:

a **=** [[1,2,3],[4,5,6],[7,8,9]]

b **=** [[1,4,7],[2,5,8],[3,6,9]]

**def** multiply\_matrice(a,b):

output **=** []

**if** len(a[0]) **==** len(b):

**for** ele **in** range(len(a[0])):

output**.**append([0 **for** ele **in** range(len(b[0]))])

**for** i **in** range(len(a)):

**for** j **in** range(len(b[0])):

**for** k **in** range(len(b)):

output[i][j] **+=** a[i][k]**\***b[k][j]

print(output)

**else**:

print('Matrix Multiplication is Not Possible')

multiply\_matrice(a,b)

O/P:

[[14, 32, 50], [32, 77, 122], [50, 122, 194]]

**3. Write a Python Program to transpose a Matrix?**

I/P:

a **=** [[1,2,3],[4,5,6],[7,8,9]]

b **=** [[1,2],[4,5],[7,8]]

c **=** [[1,2,3],[4,5,6]]

**def** generate\_transpose(in\_matrix):

out\_matrix **=** []

**for** ele **in** range(len(in\_matrix[0])):

out\_matrix**.**append([0 **for** i **in** range(len(in\_matrix))])

**for** i **in** range(len(in\_matrix)):

**for** j **in** range(len(in\_matrix[i])):

out\_matrix[j][i] **=** in\_matrix[i][j]

print(f'{in\_matrix} -> {out\_matrix}')

generate\_transpose(a)

generate\_transpose(b)

generate\_transpose(c)

O/P:

[[1, 2, 3], [4, 5, 6], [7, 8, 9]] -> [[1, 4, 7], [2, 5, 8], [3, 6, 9]]

[[1, 2], [4, 5], [7, 8]] -> [[1, 4, 7], [2, 5, 8]]

[[1, 2, 3], [4, 5, 6]] -> [[1, 4], [2, 5], [3, 6]]

**4. Write a Python Program to sort Words in an Alphabetical Order?**

I/P:

**def** sortString():

in\_string **=** input("Enter a String: ")**.**title()

sorted\_list **=** sorted(in\_string**.**split(' '))

print(' '**.**join(sorted\_list))

sortString()

O/P:

Enter a String: Full Stack Data Science

Data Full Science Stack

**5. Write a Python Program to remove Punctuations from a String?**

I/P:

**def** removePunctuatuions():

punctuations **=** '''!()-[]{};:'"\,<>./?@#$%^&\*\_~'''

in\_string **=** input('Enter a String: ')

out\_string **=** ''

**for** ele **in** in\_string:

**if** ele **not** **in** punctuations:

out\_string **+=** ele

print(out\_string)

removePunctuatuions()

O/P:

Enter a String: "Full Stacks DS" @ Ineuron

Full Stacks DS Ineuron