Python_basic_programming_8

1. Write a Python Program to Add two Matrices ?

In [2]:
```
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]]
Both Matrices must contains same no of rows and columns
Output:[[10, 10, 10]]
Both Matrices must contains same no of rows and columns
Output:[[10, 10, 10], [10, 10, 10]]
Both Matrices must contains same no of rows and columns
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]]
Both Matrices must contains same no of rows and columns
Output:[[6, 6, 10]]
Both Matrices must contains same no of rows and columns
Output:[[6, 6, 10], [2, 3, 4]]
Both Matrices must contains same no of rows and columns
Output:[[6, 6, 10], [2, 3, 4], [5, 4, 3]]
```

2. Write a Python Program to Multiply two Matrices ?

In [3]:
```
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)
```

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

3. Write a Python Program to transpose a Matrix ?

In [4]:
```
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)
```

```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]->[[1, 0, 0], [2, 0, 0], [3, 0, 0]]
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]->[[1, 4, 0], [2, 5, 0], [3, 6, 0]]
[[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, 0, 0], [2, 0, 0]]
[[1, 2], [4, 5], [7, 8]]->[[1, 4, 0], [2, 5, 0]]
[[1, 2], [4, 5], [7, 8]]->[[1, 4, 7], [2, 5, 8]]
[[1, 2, 3], [4, 5, 6]]->[[1, 0], [2, 0], [3, 0]]
[[1, 2, 3], [4, 5, 6]]->[[1, 4], [2, 5], [3, 6]]
```

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

In [5]:
```
def sortString():
    in_string = input("Enter a String: ").title()
    sorted_list = sorted(in_string.split(' '))
    print(' '.join(sorted_list))
sortString()
```

```
Enter a String: Ineuron Full Stack Data Science
Data Full Ineuron Science Stack
```

5. Write a Python Program to remove Punctuations From a String ?

In [6]:
```
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()
```

```
Enter a String: "Full Stack DS" @ Ineuron

F
Fu
Ful
Full
Full
Full S
Full St
Full Sta
Full Stac
Full Stack
Full Stack
Full Stack D
Full Stack DS
Full Stack DS
Full Stack DS
Full Stack DS
Full Stack DS
Full Stack DS  I
Full Stack DS  In
Full Stack DS  Ine
Full Stack DS  Ineu
Full Stack DS  Ineur
Full Stack DS  Ineuro
Full Stack DS  Ineuron
```

In [ ]: