**Aim:** Demonstrate the use of Numpy for matrix operations

**Source Code:**

#Addition of matrix

import numpy as np

a = np.array([[2, 4], [5, -6]])

b = np.array([[9, -3], [3, 6]])

c = a + b

print(c)

#Multiplication of matrix

a = np.array([[3, 6, 7], [5, -3, 0]])

b = np.array([[1, 1], [2, 1], [3, -3]])

c = a.dot(b)

print(c)

#Transpose of matrix

a = np.array([[1, 1], [2, 1], [3, -3]])

print(a.transpose())

#Accessing matrix elements

a = np.array([2, 4, 6, 8, 10])

print("a[0] =", a[0])

print("a[2] =", a[2])

print("a[-1] =", a[-1])

a = np.array([[1, 4, 5, 12],

[-5, 8, 9, 0],

[-6, 7, 11, 19]])

print("a[0][0] =", a[0][0])

print("a[1][2] =", a[1][2])

print("a[-1][-1] =", a[-1][-1])

#Accessing rows of the matrix

a = np.array([[1, 4, 5, 12],

[-5, 8, 9, 0],

[-6, 7, 11, 19]])

print("a[0] =", a[0])

print("a[2] =", a[2])

print("a[-1] =", a[-1])

#Accessing columns of the matrix

a = np.array([[1, 4, 5, 12],

[-5, 8, 9, 0],

[-6, 7, 11, 19]])

print("a[:,0] =",a[:,0])

print("a[:,3] =", a[:,3])

print("a[:,-1] =", a[:,-1])

#silicing matrix

a = np.array([1, 3, 5, 7, 9, 7, 5])

print(a[2:5])

print(a[:-5])

print(a[5:])

print(a[:])

print(a[::-1])

**Output:**

[[11 1]

[ 8 0]]

[[ 36 -12]

[ -1 2]]

[[ 1 2 3]

[ 1 1 -3]]

a[0] = 2

a[2] = 6

a[-1] = 10

a[0][0] = 1

a[1][2] = 9

a[-1][-1] = 19

a[0] = [ 1 4 5 12]

a[2] = [-6 7 11 19]

a[-1] = [-6 7 11 19]

a[:,0] = [ 1 -5 -6]

a[:,3] = [12 0 19]

a[:,-1] = [12 0 19]

[5 7 9]

[1 3]

[7 5]

[1 3 5 7 9 7 5]

[5 7 9 7 5 3 1]