

# question - 5

## sum demonstration

```
In [7]: import numpy as np

r = int(input("number of rows : "))
c = int(input("number of columns : "))

l = []

for i in range(r):
    l_ = eval(input("enter a row :"))
    l.append(l_)
m1 = np.matrix(l)
print("first matrix : \n",m1)

r1 = int(input("number of rows : "))
c1 = int(input("number of columns : "))

l1 = []

for i in range(r):
    l_1 = eval(input("enter a row :"))
    l1.append(l_1)
m2 = np.matrix(l1)
print("second matrix : \n",m2)

co = int(input("Enter a choice (1-sum,2-diff,3-product,4-det,5-transpose) : "))

if co ==1:
    if (r,c) == (r1,c1):
        print("sum = \n")
        print(m1+m2)
    else:
        print("sum not possible")
elif co ==2:
    if (r,c) == (r1,c1):
        print("difference = \n")
        print(m1-m2)
    else:
        print("difference not possible")
elif co == 3:
    if c == r1:
        print("dot product = \n",m1.dot(m2))
    else:
        print("not possible")

elif co ==4:
    print("the determinants are :")
    if r==c:
        print(np.linalg.det(m1))
        if r1 == c1:
            print(np.linalg.det(m2))
    else:
        print("the transpose are :")
        print(np.transpose(m1))
        print(np.transpose(m2))
```

```

number of rows : 2
number of columns : 2
enter a row :1,2
enter a row :2,3
first matrix :
[[1 2]
 [2 3]]
number of rows : 2
number of columns : 2
enter a row :4,5
enter a row :5,6
second matrix :
[[4 5]
 [5 6]]
Enter a choice (1-sum,2-diff,3-product,4-det,5-transpose) : 1
sum =

[[5 7]
 [7 9]]

```

## transpose demonstration

```

In [9]: import numpy as np

r = int(input("number of rows : "))
c = int(input("number of columns : "))

l = []

for i in range(r):
    l_ = eval(input("enter a row :"))
    l.append(l_)
m1 = np.matrix(l)
print("first matrix : \n",m1)

r1 = int(input("number of rows : "))
c1 = int(input("number of columns : "))

l1 = []

for i in range(r):
    l_1 = eval(input("enter a row :"))
    l1.append(l_1)
m2 = np.matrix(l1)
print("second matrix : \n",m2)

co = int(input("Enter a choice (1-sum,2-diff,3-product,4-det,5-transpose) : "))

if co ==1:
    if (r,c) == (r1,c1):
        print("sum = \n")
        print(m1+m2)
    else:
        print("sum not possible")
elif co ==2:
    if (r,c) == (r1,c1):
        print("difference = \n")
        print(m1-m2)
    else:
        print("difference not possible")
elif co == 3:
    if c == r1:
        print("dot product = \n",m1.dot(m2))
    else:
        print("not possible")

elif co ==4:

```

```

print("the determinants are :")
if r==c:

    print(np.linalg.det(m1))
if r1 == c1:
    print(np.linalg.det(m2))
else:
    print("the transpose are :")
    print(np.transpose(m1))
    print(np.transpose(m2))

```

```

number of rows : 2
number of columns : 2
enter a row :1,2
enter a row :3,4
first matrix :
[[1 2]
 [3 4]]
number of rows : 2
number of columns : 2
enter a row :4,5
enter a row :5,6
second matrix :
[[4 5]
 [5 6]]
Enter a choice (1-sum,2-diff,3-product,4-det,5-transpose) : 5
the transpose are :
[[1 3]
 [2 4]]
[[4 5]
 [5 6]]

```

## question 6

```

In [14]: from sympy import *

x, y = symbols('x,y')
eq1 = Eq(x**2+y**2,5)
eq2 = Eq(y,x*3 -5)
sol = solve([eq1,eq2],[x,y])
print("intersection points in (x,y) form are : ")
print(sol)

```

```

intersection points in (x,y) form are :
[(1, -2), (2, 1)]

```