

## Assignment 3 : 2020-21 Test 4

Class: SE IV Roll No: 21430

Batch: FG

DOP:

DOS: 25/10/2020

### Problem Statement:

Write a python program to compute the following computation on matrix.

- Addition of two matrices
- Subtraction of two matrices
- Multiplication of two matrices
- Transpose of a matrix.

### Learning objective:

To implement concept of matrix multiplication subtraction, addition and transpose in python, using numpy module's data structure, matrix and array.

### Learning Outcomes:

Will be aware of implementation of matrix multiplication, addition, subtraction and transpose using data structure matrix & array in Numpy module of python.

### Software requirements

Python IDE community version 2020

O.S - Windows 10

### Hardware requirements

Windows 10 64 bit.

8 GB ram.

## \* Theory

### i) Abstract data type (ADT)

ADT is a type for objects whose behaviour is defined by a set of values or operations. Its definition only comprises of what operations are to be performed not implementation. It doesn't specify memory organization and algorithm implements. This process is called abstraction.

#### Syntax ::

ADT\_name():

Data-1, Data-2

Fun-1()

Fun-2()

#### Eg.

```
class matrix:
```

```
    mat = []
```

```
    addition()
```

```
    subtraction
```

### ii) class

A class is a building block that leads to object oriented programming. It is a user defined data type which holds the data member and member functions. It acts as a blueprint for objects.

#### Syntax:

```
class class_name:
```

```
    [statement 1]
```

```
    statement 2
```



E.g.

```
class matrix:
    def __init__(self):
        self.matrix = []
    def addition(self, m2):
        return (self.matrix + m2.matrix)
```

iii) object:-

An object is simply a collection of data and variables that act on those data. An object is also called an instance of a class and the process of creating this object is called instantiation.

syntax:

obj-name = class-name()

Eg. mat = matrix()

iv) Matrix:

matrix is an arrangement of numbers into rows and columns. Number of operations can be performed using matrix, addition, subtraction, multiplication, transpose etc.

Addition:

Matrix addition is possible if both matrices have same no of rows and columns. Each corresponding elements in the matrices are added.

Algorithm:

matrix  $A[r][c]$ ,  $B[r][c]$ ,  $C[r][c]$



for i from 0 to r-1

for j from 0 to c-1

$$C[i][j] = A[i][j] + B[i][j]$$

return C

\* Subtraction:-

matrix subtraction is possible when number of rows and columns are same in both matrices.

The subtraction matrix is obtained by subtracting the each element from one matrix from another matrix.

Algorithm:-

matrix A[r][c], B[r][c], C[r][c]

for i from 0 to r-1

for j from 0 to c-1

$$C[i][j] = A[i][j] - B[i][j]$$

1. Multiplication:-

Matrix multiplication is possible when the number of rows of second matrix is equal to no. of column of 1<sup>st</sup> matrix. The resulting matrix has no. of array as of 1<sup>st</sup> matrix and no. of columns as of 2<sup>nd</sup> matrix.

Algorithm:-

matrix A[r<sub>1</sub>][c<sub>1</sub>], B[r<sub>2</sub>][c<sub>2</sub>], C[r<sub>1</sub>][c<sub>2</sub>]

for i from 0 to r<sub>1</sub>-1

for j from 0 to c<sub>2</sub>-1

$$C[i][j] = 0$$

for i from 0 to r<sub>1</sub>-1

for j from 0 to c<sub>2</sub>-1

for k from 0 to c<sub>1</sub>-1

$$C[i][j] = C[i][j] + A[i][k] * B[k][j]$$

end for

end for



\* ADT for matrix class:

class Matrix:

matrix = [] #empty matrix.

addition() #returns add of 2 matrix.

subtraction() #returns matrix subtraction of 2 matrix.

multiplication() #returns multiplication of 2 matrix.

transpose() #returns transpose of matrix.

getmatrix() #takes i/p value of matrix.

Show() #display matrix.

\* Psuedocode of main program.

i. mat1 = Matrix()

ii. mat2 = Matrix()

iii. mat1.read()

iv. mat2.read()

v. read\_option()

vi. if (option == 1)

vii. mat1.addition(mat2)

viii. else if (option == 2)

mat1.subtraction(mat2)

ix. else if (option == 3)

mat1.multiplication(mat2)

x. else

option.read()

if (option == 1)

mat1.transpose()

else

mat2.transpose()

END

\* Pseudocode for addition

- i. start
- ii.  $c = []$  #  $n \times n$  matrix
- iii. for  $i$  from 0 to  $n-1$ 
  - 3.1 for  $j$  from 0 to  $n-1$ 
    - 3.2  $c[i, j] = A[i, j] + B[i, j]$
    - 3.3 END for
  - 3.4 END for
- iv. return  $c$
- v. END

\* Pseudocode for subtraction

- i. start
- ii.  $c = []$  #  $n \times n$  matrix.
- iii. for  $i$  from 0 to  $n-1$ 
  - for  $j$  from 0 to  $n-1$ 
    - $c[i, j] = A[i, j] - B[i, j]$
  - END for
- iv. END for
- v. return  $c$
- vi. END

\* Pseudocode for multiplication

- i. start
- ii.  $c = []$  #  $n \times n$  matrix.
- for  $i$  from 0 to  $n-1$ 
  - for  $j$  from 0 to  $n-1$ 
    - $c[i, j] = A[i, j] * B[i, j]$
  - END for
- iii. END for
- iv. return  $c$
- v. END



\* Psuedocode for transpose

- i. Start
- ii.  $B = []$  #  $n \times n$  matrix
- iii. for  $i$  from 0 to  $n-1$   
     for  $j$  from 0 to  $n-1$   
          $B[i, j] = A[j, i]$   
     END for
- iv. END for
- v. return  $B$
- vi. END

\* Test cases:

NO.	Description	Input	Expected o/p	Actual o/p	Result.
1.	1. Addition 2. subtraction 3. multiplication 4. Transpose	$mat1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$add = \begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$	$add = \begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$	Pass
2.	1. Addition 2. subtraction 3. multiplication 4. Transpose	$mat1 = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$ $mat2 = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$	$add = \begin{bmatrix} 4 & 6 \\ 8 & 10 \end{bmatrix}$	$add = \begin{bmatrix} 4 & 6 \\ 8 & 10 \end{bmatrix}$	Pass

Conclusion:-

We learnt to implement matrix in python and its operations of addition, subtraction, multiplication and transpose

```

24 for i in range(b):
25     row = []
26     print("Enter element in ", i + 1, " Row")
27     for j in range(a):
28         l = int(input())
29         row.append(l)
30     m2.append(row)
31 print("Given matrix is :- ")
32 for i in range(a):
33     for j in range(a):
34         print(m2[i][j], end=" ")
35     print()
36 y = array(m2)
37 m2 = matrix(y)
38 if(m==n==a==b):
39     print("choose any of the following choice ")
40     print("Enter 1 for matrix addition ")
41     print("Enter 2 for Matrix subtraction ")
42     print("Enter 3 for marix multiplication ")
43     print("Enter 4 for matrix transpose ")
44     print("Enter 5 for exit")
45     choice=int(input("Enter your choice "))
46     m1=matrix('1 2 3;2 3 4;3 4 5')

```

while (1) > for i in range(n) > for j in range(n)

Looks like you're using NumPy  
Would you like to turn scientific mode on?  
[Use scientific mode](#) [Keep current layout...](#)

Python Console x main x FDS ASSIGNMENT3 x FDS ASSIGNMENT3 (1) x

Problems TODO Terminal Python Console

Event Log

16:27 CRLF UTF-8 4 spaces Python 3.8 (pythonProject)



```
1 from numpy import *
2 m1=[]
3 m2=[]
4 while(1):
5     n = int(input("Enter Number of column "))
6     m = int(input("Enter Number of rows "))
7     for i in range(m):
8         row = []
9         print("Enter element in ", i + 1, " Row")
10        for j in range(n):
11            l = int(input())
12            row.append(l)
13            m1.append(row)
14        print("Given matrix is :- ")
15        for i in range(n):
16            for j in range(n):
17                print(m1[i][j], end=" ")
18            print()
19        x=array(m1)
20        m1=matrix(x)
21
22        a = int(input("Enter Number of column "))
23        b = int(input("Enter Number of rows "))
```

while (1) > for i in range(m)

Looks like you're using NumPy  
Would you like to turn scientific mode on?  
[Use scientific mode](#) [Keep current layout...](#)

Python Console x main x FDS ASSIGNMENT3 x FDS ASSIGNMENT3 (1) x

Problems TODO Terminal Python Console

Event Log

9:50 CRLF UTF-8 4 spaces Python 3.8 (pythonProject)

```
FDS ASSIGNMENT3.py x 1\main.py x main.py x ganesh.py x
41 print("Enter 1 for matrix addition ")
42 print("Enter 3 for matrix multiplication ")
43 print("Enter 4 for matrix transpose ")
44 print("Enter 5 for exit")
45 choice=int(input("Enter your choice "))
46 m1=matrix('1 2 3;2 3 4;3 4 5')
47 m2 = matrix('1 2 3;2 3 4;3 4 5')
48 if(choice==1):
49     m3=matrix(m1+m2)
50     print("Addition matrix is ",m3)
51 elif(choice==2):
52     m3 = matrix(m1 - m2)
53     print("subtraction matrix is ", m3)
54 elif choice == 3):
55     m3 = matrix(m1 * m2)
56     print("multiplication matrix is ", m3)
57 elif choice == 4):
58     m3 = matrix(transpose(m1))
59     print("transposematrix is ", m3)
60     m3 = matrix(transpose(m2))
61     print("transposematrix is ", m3)
62 elif(choice==5):
63     exit()
```

```
while (1) > for i in range(n) > for j in range(n)
```

Python Console x main x FDS ASSIGNMENT3 x FDS ASSIGNMENT3 (1) x