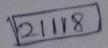
Assignment No: 03 Matrix operations



Problem Statement.

Write a python program to compute following computation ou matrix.

or addition of two matrices

by Substraction of two matrices

of multiplication of two matrices

of Transpose of a matrix

objectives:

> Understand concept & operations of making

2) Understand use of 20 list to represent motion & peeform various operations

Outromes:

is to implement string operations using list data structure

in python

2) To write meny driven, modular program in python 2) To implement user defined functions in the python

Hardware Requirements:

nambanturer: Aca

model Stoff St 314-559

Processos: Intel(R) Gre(TM) 15-82650 epu@ 1.604H2

1.8 GH

Installed memory (RAM): 8.00 4B (7.85 4B Available)

System Type: 64-bib 05, x 64-based orchitecture.

Soffware Requirements:

as: coindons 10, Home Engle language (Version: 1903)

Python Version: 3.8.5

Vs Code (bent editor) : Version: 1.49.1 (Aug. 2020 vernion)

Classmate Page

Theory:

Concept:
Mateix: 20 array of elements having nounceted.

solumns number of columns.

· Matrix operations:

Apaddition: (ow major)

is addition of two matrices is possible only if two
matrices have came sows of same columns.

is the addition matrix is a one whose element at
location (i'ij) is the sum of element at location

(ij) for the two matrices.

they A+B = [3 5] = Addition matrix

Substraction (said major)

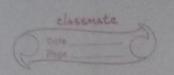
Psubstraction of two materices is possible only if they have some was some of same columns

1) the substraction materix is a one whose element at location (iii) is the substraction of element at location (iii) for the two materices

 $A-13 = \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix}$ = Substraction matrix

of Mustiplication (Fow major):

only if columns of A 4 saws of B are equal.



| A = | Tau | 912 | 913 | | · 9m] | B= | b,1 | b12. | . bik | |
|-----|------|-----|------|------|-------|------------------------|-----|-------|-------|------|
| | 91 | | | | | | bal | | | |
| - | 931 | - | | | | | ba) | -1.5. | . ? | |
| | - | - | .6 4 | 205. | Num | xu- | L | | buk - | nxk. |
| | Lami | | , | 180 | amnı | NAME OF TAXABLE PARTY. | DNI | | | |

the product matery M is given as.

M = [9, b, +9,2b2) ... + 9m, bn, ...

amibit ... + amo boi ami bikt ... + ambox mock

eq.
$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$
 $B > \begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix}$

product mater M= 5 87

d> Transpose: The transpose of a given materix is a materix obtained by interchanging zows & columns.

the transpose of A, AT = [1 9 3 6]

· class: class is a blueprint of object It provides way to implement various oop concepts.

eg data hiding, abstraction.

object som instance of class for same class there can be multiple objects.

· oop concepts:

data hiding work through function, don't play with actual data.

Abstraction: shaving imp details & hiding uncerenary
things

Also basic knowledge of python language, makage lub data type is required.

ADT:

ADT matrix is

pata object: A set of sows & columns & c = {0, ... x} x belongs to tre sureges.

Represented as array of two dimensions with indices of

M= 40, ..., UE-15 N= 40...; UQ-15

for each mater M, mf n are sows & columns 42, 4l are copper bounds of sows & columns.

Create Makers (42, 42): Makers

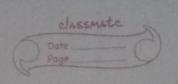
Display Matrix (42,48): Matrix

Marquete matrix of 42x48.

Add Matrices (A, UZA, UZA, B, UZB, UQB): Matrix MAdds matrices A RB If possible.

SubMatrices (A, UZA, ULP, B, UZB, UEB): Matrix
(1) Substrait matrix B From A III possible.

17 Multiply two matrices A&B if possible



Transpose (A, UR, , UR): Matrix 11 finds transpose of matern A.

class Declaration:

· class Matrix is declased with number of gows, columns &

corresponding to 20 11st.

· Committee used for class is parameterzed. If value for But & column are provided it creates pero matern of that particular Size.

else creates empty list.

· Other methods in class are as below:

class Materix U:

def .inib = - (self, 80ws =0, columns =0).

self-gows = gows

Self- colums = colums.

Selfymymatrix = [[0] * seff.columns for _ in range (self-tow)]

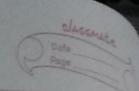
def readmatrix (self): # reads the mater.

def add Matrices (self, matrix B): addition Matrix Hadds maters with maters 13

det Substrait Matrices (self, matro. B): Rubitraitien Matro # Substract material from material (10. mymaters)

det Multiply Matrices (self, matrix B): products matrix with matrix. B.

def Transpose (self) ancidasets): Transpose Matrix # returns branspose of matrix (re. mymatrix)



Algorithm for each operation

Algorithm add Matrices (A, B)

1. Check If your & (dumn are of A 3B are of ul 60 not, if not equal tetuer from method.

2. create materix c. of order some as A

3. for each element at index (1/j) of matrix c arrigh this clement with sum of elements at Same index in A & B.

4 return matric c.

Algorithm Sub Matrices (AIB)

1. Check If zows & columns of A & B are equal. @ not if not equal return from method.

2. create maters c of order same as A.

3. for each element at index (1) of matrix 4 ashigh this element with difference of elements at same index in A & B

4. return matric c

Algorithm Mult. Matrices (A, B):

1. Check if columns of A are equals to East of B. If equal proceed to step 2 die zetury from

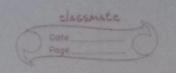
2. beate materix c of order some as low of AX Colymnof

3. multiply its 9000, with jts column of B & arrigh the sesult to the elever index (i,i) in matrix c.

4. teturu matrix c.

Algorithm Transpose (A):

1. create matrix AT of order column of A X row of A



2. for each dement at index ((ij) in A, asingn it to the index (j,i) in matrix AT.

Analysis of Algorithms:

| | 0 | | |
|-----|-------------------------------------------|-----------------------|----------------------------|
| | Algorithm | Time Complexity | Spare Couplersity |
| 7 | add Matrices () | They: | spare is required |
| | P. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. | O(mxn) where | to store the elements |
| | 9696 | m, n are goust | of addition matrix. |
| | | columns respectively. | |
| | | / 0 | space complexity is |
| | | | where min are |
| | 1 8 3 3 | TE + E - 5 | East resolumns respe. |
| | | | |
| 27 | Submatrices () | O(MXH) where MIH | space complexity |
| | | are tows & columns | space complexity is O(MXN) |
| | | respectively. | where min are Eaws |
| | UQ O I IIA | 0 | & column respectively |
| | | 1500 | , , |
| 37 | Mulf Matrices() | leb Amxu. | space complexity |
| | | 4BAX.10 | is O(mxb) |
| | A 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | then time complexity. | where A mxu & |
| | 4 60 | will be o (m/o/s) | Bnx P. |
| | | 0645 | |
| 10> | Transpose() | O(mxn) where | O(mxn) where min |
| 7. | Transpose() | M, u are Eows & | are zows & columns |
| | 28 | Columns of matrix | of maters. |
| | | | 12 2 42 |
| | | | |
| | | | water la duni et la |
| | | | |

| Test Cases: | The and the sale of | Jan Landon |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|--------------------------------------------------------------|
| A 1B are Ir | ient matrices, c is or | TRUO MATRIX. |
| Test case Input given | exparted output | Actual output |
|). A= 00 | c= [0 0] | (=[00] |
| B=[0 07 | 700 | |
| for addition | 3 | |
| 2 A = [1 23] 4 56 | Addition not | Addition not |
| B= [9 237 4 56] | possible. | possible |
| for addition 3 $A = \begin{bmatrix} -1 & 2 & 3 \\ -4 & 5 & -6 \end{bmatrix}$ | $c = \begin{bmatrix} 3 & 7 & -3 \\ -3 & 7 & -9 \end{bmatrix}$ | c=[37-3] -37-9 |
| $B = \begin{bmatrix} 4 & 5 & -6 \\ 1 & 2 & -3 \end{bmatrix}$ $+61 \text{addition}$ | tamele de cos ca | Chandala vo |
| 4) - for transpose A = [0 0 0] 0 0 0 | AT = [000] | AT = [000] |
| 5) for transpore. A = [1 0 0] 0 2 0 0 0 3 | AT = \[1 \ 0 \ 0 \ 0 \ 0 \ 3 \] | AT=[000 003] |
| for transpose $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ | $AT = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$ | $AT = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$ |
| 7) for multiplication | | |

| | A = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | K:= [0 00] | e= [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|-------------------------------------------|
| 8> | $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}_{2\times 2}$ $B = \begin{bmatrix} 1 & 2 \\ 3 & 9 \\ 5 & 6 \end{bmatrix}_{3\times 2}$ | Multiplication not passible | Multiplication nob possible |
| *> | $A = \begin{bmatrix} 1 & 2 & 3 \\ 9 & 5 & 6 \end{bmatrix}_{2 \times 3}$ $B = \begin{bmatrix} 1 & 2 \\ 3 & 9 \\ 3 & 6 \end{bmatrix}_{3 \times 2}$ | C= [22 28] 49 64] 2×2 | c= [22 28] 49 64] |
| | | | |

Applications

Applications of matrices are found in most scientific fields.

En every branch of physics, including classical mechanics optics, electromagnetism, quantum mechanics of quantum electrodynamics, they are und to study physical phenomenous such as the motion of rigid bodies.

conclusion:

At the end of this assignment I'm able to perform vazious matros operations using the computer with help of python program. Got better undoestanding of matrox operations of developed programming logic.