Modern Education Society's

College of Engineering, Pune

NAME OF STUDENT: Sandesh Santosh Pabitwar CLASS: Comp A	
SEMESTER/YEAR: III sem	**ROLL NO: F20111040**
DATE OF PERFORMANCE:	**DATE OF SUBMISSION:19/11/2021**
EXAMINED BY:Prof.Amol Dhawale	**EXPERIMENT NO: DSL A-09**

TITLE : PERFORM VARIOUS OPERATIONS ON MATRICES

PROBLEM STATEMENT: Write a **Python** program to compute following computation on matrix:

a) Addition of two matrices B) Subtraction of two matrices

c) Multiplication of two matrices d) Transpose of a matrix

OBJECTIVES:

1. To understand structure of 2DArray.
2. To understand how to Create, Display and perform various operations on 2D array.

OUTCOMES:

1. To analyze the problems to apply suitable algorithm and data structure.
2. To understand concept of multi-dimensional array.

PRE-REQUISITES:

1. Knowledge of python programming
2. Knowledge of 2D array and matrix operations.

APPARATUS:

QUESTIONS:

1. What is sparse matrix? Explain with example.

2. Write algorithm to perform fast transpose on sparse matrix.

```python
# all operations are for 3*3 matrix
print('enter elements of matrix one')
a = list(map(int, input().split()))
b = list(map(int, input().split()))
c = list(map(int, input().split()))
print('enter elements of second matrix')
d = list(map(int, input().split()))
e = list(map(int, input().split()))
f = list(map(int, input().split()))
m1 = [a, b, c]
m2 = [d, e, f]
m3 = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]  # empty matrix to hold output


def addition(m1, m2):
    for i in range(3):
        for j in range(3):
            m3[i][j] = m1[i][j] + m2[i][j]

    print('addition matrix')
    print(m3)


addition(m1, m2)


def substraction(m1, m2):
    for i in range(3):
        for j in range(3):
            m3[i][j] = m1[i][j] - m2[i][j]

    print(m3)
substraction(m1, m2)


def transpose(m1):
    for i in range(3):
        for j in range(3):
            m3[i][j] = m1[j][i]

    print('transpose of m1')
    print(m3)


transpose(m1)


def multiplication(m1, m2):
    # iterating by row of m1
    for i in range(3):

        # iterating by column by m2
        for j in range(3):

            # iterating by rows of m2
            for k in range(3):
                m3[i][j] += m1[i][k] * m2[k][j]
    print('multiplication of two matrix is:')
    print(m3)

multiplication(m1, m2)
```

Output:

```
C:\Users\sspab\PycharmProjects\new\venv\Scripts\python.exe
enter elements of matrix one
1 2 3
4 5 6
7 8 9
enter elements of second matrix
2 4 6
6 7 8
5 6 7
addition matrix
[[3, 6, 9], [10, 12, 14], [12, 14, 16]]
subtraction matrix
[[-1, -2, -3], [-2, -2, -2], [2, 2, 2]]
transpose of m1
[[1, 4, 7], [2, 5, 8], [3, 6, 9]]
multiplication of two matrix is:
[[30, 40, 50], [70, 92, 114], [110, 144, 178]]

Process finished with exit code 0
```

DSL A09

Question

1. What is sparse matrix? Explain with example.

Ans: Sparse matrix:

A matrix can be defined as a two-dimensional array having 'm' columns and 'n' rows representing m x n matrix. Sparse matrices are those matrix that have the majority of their elements equal to zero. In other words, the sparse matrix can be defined as the matrix that has a greater number of zero elements than the non-zero elements.

e.g.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 9 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

Let's consider above 5 X 6 matrix containing 6 numbers of non-zero values. This matrix can be represented as shown below.

Row	columns	values.
5	6	6
0	4	9
1	1	8
2	0	4
2	3	2

Row	Column	values
3	5	5
4	2	2

In above example matrix, there are only 6 non zero elements. and matrix size is 5X6. We represent this matrix as shown in above table. Here the first row in the table is filled with values 5, 6 & 6 which indicates that it is a sparse matrix with 5 rows, 6 columns & 6 non zero values. The second row filled with 0, 4 & 9 which indicates the non-zero value 9 is at 0th row, 4th Column in the Sparse matrix. In the same way, the remaining non-zero values also follow similar pattern.

2. Write algorithm to perform fast transpose on sparse matrix.

Algorithm:

Step 1: obtain the triplet from sparse matrix.
Step 2: Create 2 one dimensional array total_array & inden_array.
Step 3: size of total array is the the original matrix.
Step 4: Ate every inden of total, put the number of times respective inden appears in second column of sparce matrix.
Step 5: Size of inden array: size of total array + 1
Step 6: inden[0] = 1 & inden[i] = inden[i-1] + total[i-1]
Step 7: Traverse the sparse matrix from second row & consider column elements.
Step 8: Location = inden[col-No]
Step 9: Locationth inden of transpose matrix = swapped row from original matrix.
Step 10: Increase inden[col-No] by 1.
Step 11: Repeate step 7 to 10 for remaining triplets of original matrix.