# Experiment No-03

**Title:-** Python program to compute various operations on matrix. **Objectives:-** To understand the use of matrix operations **Problem Statement:-**

Write a Python program to compute following computation on matrix:

1. Addition of two matrices
2. Subtraction of two matrices
3. Multiplication of two matrices
4. Transpose of a matrix

# Theory-

**Operations on Matrices**

Addition, subtraction and multiplication are the basic operations on the matrix.

To add or subtract matrices, these must be of identical order and for multiplication, the number of columns in the first matrix equals the number of rows in the second matrix.

* Addition of Matrices
* Subtraction of Matrices
* Multiplication of Matrices
* Transpose of Matrices

# Addition of Matrices

If A[aij]mxn and B[bij]mxn are two matrices of the same order then their sum A + B is a matrix, and each element of that matrix is the sum of the corresponding elements. i.e. A + B

= [aij + bij]mxn

Consider the two matrices A & B of order 2 x 2. Then the sum is given by:



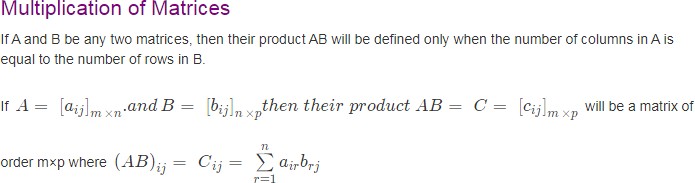
# Subtraction of Matrices

If A and B are two matrices of the same order, then we define A-B=A+\left( -B

\right).*A*−*B*=*A*+(−*B*).

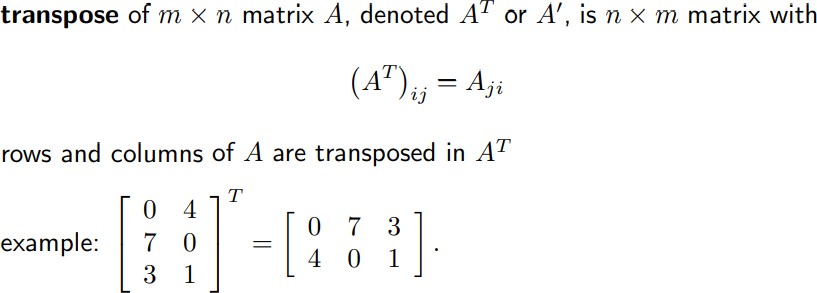
Consider the two matrices A & B of order 2 x 2. Then the difference is given by:

We can subtract the matrices by subtracting each element of one matrix from the corresponding element of the second matrix. i.e. A – B = [aij – bij]mxn



# Transpose of a Matrix

The matrix whose row will become the column of the new matrix and column will be the row of the new matrix.



**Pseudocode for matrix addition**

**matrixAdd(matrix1, matrix 2):**

1. If the number of rows and number of columns of matrix 1 and matrix 2 is equal,
2. for i=1 to rows[matrix 1]
3. for j=1 to columns [matrix 1]
4. Input matrix 1 [i,j]
5. Input matrix 2 [i,j]
6. matrix 3 [i,j]= matrix 1 [i,j]+ matrix 2 [i,j];
7. Display matrix 3 [i,j];

**Pseudocode for matrix subtraction**

**matrixsubtraction(matrix1, matrix 2):**

1. If the number of rows and number of columns of matrix 1 and matrix 2 is equal,
2. for i=1 to rows[matrix 1]
3. for j=1 to columns [matrix 1]
4. Input matrix 1 [i,j]
5. Input matrix 2 [i,j]
6. matrix 3 [i,j]= matrix 1 [i,j]- matrix 2 [i,j];
7. Display matrix 3 [i,j];

**Pseudocode for matrix multiplication**

**matrixMultiply(A, B):**

# Assume dimension of A is (m x n), dimension of B is (p x q)

1.If n and p are equal then perform steps 2 to 3

2.if n is not same as p, then exit

   otherwise define C matrix as (m x q)

  3.for i in range 0 to m - 1, do

     i) for j in range 0 to q – 1, do

       ii) for k in range 0 to p, do

            C[i, j] = C[i, j] + (A[i, k] \* A[k, j])

         done

      done

   done

End

**Algorithm for finding transpose of matrix**

1. Declare and initialize a 2-D array p[a][b] of order axb.
2. Read the matrix p[a][b] from the user.
3. Declare another 2-dimensional array **t** to store the transpose of the matrix. This array will have the reversed dimensions as of the original matrix.
4. The next step is to loop through the original array and to convert its rows to the columns of matrix t.
   * Declare 2 variables i and j.
   * Set both i,j=0
   * Repeat until i<b
     + Repeat until j<a
     + t[i][j] = p[j][i]
     + j=j+1\*\*
   * i=i+1
5. The last step is to display the elements of the transposed matrix t.

**Conclusion:**

Hence we studied matrix representation using 2 D array and how to perform matrix addition,subtraction, multiplication and transpose.