In [1]:

```
import numpy as np    #for loadtxt()
```

In [2]:

```
#loading the matrices

A = np.loadtxt('matrix_a.csv', delimiter = ',')      #2D ARRAY
B = np.loadtxt('matrix_b.csv', delimiter = ',')      #2D ARRAY
C_1 = np.loadtxt('matrix_c.csv', delimiter = ',')    #1D ARRAY which will be interpreted as row matrix
D_1 = np.loadtxt('matrix_d.csv', delimiter = ',')    #1D ARRAY which will be interpreted as column matrix

C = []         #ROW MATRIX
C.append(C_1)

D = []         #COLUMN MATRIX
for i in D_1:
    temp = []
    temp.append(i)
    D.append(temp)
```

In [3]:

```
print("Matrix A is: ", "\n", A)    #SQUARE MATRIX
print("Matrix B is: ", "\n", B)    #SQUARE MATRIX
print("Matrix C is: ", "\n", C)    #ROW MATRIX
print("Matrix D is: ", "\n", D)    #COLUMN MATRIX
```

```
Matrix A is:
 [[1. 2. 0.]
 [5. 3. 6.]
 [7. 4. 9.]]
Matrix B is:
 [[ 5.  1.  6.]
 [ 3.  8.  3.]
 [ 5.  9. 12.]]
Matrix C is:
 [array([7., 4., 9.])]
Matrix D is:
 [[2.0], [6.0], [1.0]]
```

In [4]:

```
def mat_mult(X, Y):                              #DEFINING THE MATRIX MULTIPLICATION FUNCTION
    len_row_X = len(X)
    len_column_X = len(X[0])
    len_row_Y = len(Y)
    len_column_Y = len(Y[0])

    if len_column_X == len_row_Y:                #CHECKING IF MATRIX MULTIPLICATION IS POSSIBLE
        Z = []                                   #THIS WILL BE FINAL MULTIPLIED MATRIX
        for i in range(len(X)):
            Z_row_temp = []                      #THIS WILL BE THE TEMPORARY ROWS APPENDED TO Z AFTER EACH ITERATION
            for j in range(len(Y[0])):
                temp = 0                         #THIS WILL BE THE TEMPORARY ELEMENTS OF EACH ROW
                for k in range(len(Y)):
                    temp += X[i][k]*Y[k][j]      #THE MAIN ITERATION
                Z_row_temp.append(temp)          #APPENDING THE ELEMENT TO THE ROW
            Z.append(Z_row_temp)                 #APPENDING THE ROW TO Z
        return Z
    else:
        print("Invalid Argument")
```

In [5]:

```
def print_matrix(Z):                             #DEFINING THE MATRIX PRINTING
    if Z is None:                                #FOR WHEN THE MULTIPLICATION IS INVALID
        print("\n")
    else:
        for row in Z:
            print(row)
```

In [6]:

```
print matrix(mat mult(C, D))                     #ROW X COLUMN
```

```
[47.0]
```

In [7]:

```
print matrix(mat mult(A, B))                     #SQUARE X SQUARE
```

```
[11.0, 17.0, 12.0]
[64.0, 83.0, 111.0]
[92.0, 120.0, 162.0]
```

In [8]:

```
print matrix(mat mult(C, A))                     #ROW X SQUARE
```

```
[90.0, 62.0, 105.0]
```

In [9]:

```
print matrix(mat mult(B, D))                     #SQUARE X COLUMN
```

```
print matrix(mat mult(B, B))                    #SQUARE X COLUMN
```

```
[22.0]
[57.0]
[76.0]
```