Applied Linear Structures for Computing
Mini-project 1, Total Grade= 100 (+5) points
There is a bonus question with 5 extra points
Instructor: M. Hadi Amini
TA: Leila Zahedi

For questions regarding homework please reach out to the TA (lzahe001@fiu.edu)

Name: Garcia Milord

Panther ID: 6168616

[1] **PART I**

For the following three problem, you will first compute solutions by hand (show as many steps as possible, including transpositions). Then, use programming to compute the same solution. Submit your code as a single PDF document along with your hand written work (write comments to describe what your code is doing). Feel free to also use popular computational packages (i.e. Numpy for Python).

Question 1: (20 points)
Compute the norm of $\langle 3, 1, 2 \rangle$.

```
def norm():
   x=3
   y=1
   z=2
   norm = math.sqrt(pow(x,2)+pow(y,2)
+pow(z,2))
   return print('Question 1: Part 1: Compute the
norm of (3,1,2): \n'+ str(norm))
#print solution 3.7416573867739413
```

Question 2:(20 points)
Compute the product $\mathbf{A}^T\mathbf{x}$ given that:

$$A = \begin{bmatrix} 4 & -2 & 1 \\ 5 & 7 & -1 \end{bmatrix}, x = \begin{bmatrix} 2 \\ 3 \end{bmatrix} =$$

```
def transposeProduct():
   Amatrix = np.array([[4, -2, 1], [5, 7, -1]])
   Tmatrix =Amatrix.T
   Xvector = np.array([[2],[3]])
   Product = Tmatrix @ Xvector
   return print('Question 2: Part 1: Compute
the product AT x given that: \n'+
str(Product))
#print solution
 [[23]
 [17]
 [-1]]
```

Question 3: (25 points)
Compute the determinant

$$\begin{vmatrix} 3 & 4 & 3 \\ -1 & 9 & -1 \\ 7 & 2 & 7 \end{vmatrix}$$

```
def determinant():
   matrix = np.array([[3,4,3],[-1,9,-1],
[7,2,7]])
   det = np.linalg.det(matrix)
   return print('Question 3: Part 1:
Compute determinant: \n'+str(det))
#print solution 0.0
```

PART II

The next questions are programming related. Python is the recommended language.

Question 1:(15+5 points)
Write down a matrix multiplication code for multiplying two arbitrary matrices. Part 1: Mandatory (15 points): The first function should use popular packages (i.e. if you are using python, use the numpy package) to achieve the same result.
Part 2: (5 extra points) The second function should be programmed by yourself based on the definition of matrix multiplication.

Question 2:(20 points)
Solve the following system of equations

$$2x_1 + 2x_2 + 6x_3 = 24$$

$$2x_1 - 2x_2 - 2x_3 = 0$$

$$4x_1 + 2x_2 - 4x_3 = 6$$

```
def matrixMultipcation():
    a = np.array([[-1.5,3,2],[1,-1,0]])
    b = np.array([[-1,-1],[0,2],[1,0]])
    c = a @ b
    return print("Question 1: Part 2: numpy matrix multiplication: \n"+str(c))
Question 1: Part 2: numpy matrix multiplication:
[[ 3.5  7.5]
 [-1.  -3. ]]

def myMatrixMultipcation():
    #2x4
    Am = [[-1.5,3,2],[1,-1,0]]
    #3x2
    Bm = [[-1,-1],[0,2],[1,0]]
    print("Question 1: Part 2: my matrix multiplication: ")
    result = [[sum(a * b for a, b in zip(A_row, B_col))
               for B_col in zip(*Bm)]
                  for A_row in Am]
    for r in result:
        print(r)
Question 1: Part 2: my matrix multiplication:
[3.5, 7.5]
[-1, -3]

def systemofequation():
    # 2* x1 + 2* x2 + 6* x3 =24
    # 2* x1 - 2* x2 - 2* x3 =0
    # 4* x1 + 2* x2 - 4* x3 =6
    A = np.array([[2,2,6],[2,-2,-2],[4,2,-4]])
    B = np.array([24,0,6])
    x = np.linalg.solve(A,B )
    return print("Question 2: Solve the following system of equations: \n"+str(x))

Question 2: Solve the following system of equations:
[3.5 1.  2.5]
```