1.Create a row vector and column vector in numpy, and show the shape of the vectors.

2. Transpose the row vector you defined above into a column vector and calculate the L1, L2, and L∞ norm of it. Verify that the L∞ norm of a vector is equivalent to the maximum value of the elements in the vector. (Do all the above verifications using python,numpy,sympy)

3.Compute the angle between the vectors v=[10,9,3] and w=[2,5,12].(Do this manually using the definition of dot product. No code required)

4.Given the row vectors v=[0,3,2], w=[4,1,1], and u=[0,−2,0], write the vector x=[−8,−1,4] as a linear combination of v, w, and u.(Write python code for the same)

5.Determine by inspection whether the following set of vectors is linearly independent: v=[1,1,0], w=[1,0,0], u=[0,0,1]. (do this manually)

6.Let the Python matrices P=[[1,7],[2,3],[5,0]] and Q=[[2,6,3,1],[1,2,3,4]]. Compute the matrix product of P and Q. Show that the product of Q and P will produce an error. Find and explain the reason for the error.

7.Use Python to find the determinant of the matrix M=[[0,2,1,3],[3,2,8,1],[1,0,0,3],[0,3,2,1]]. Use the np.eye function to produce a 4×4 identity matrix, I. Multiply M by I to show that the result is M.

8.
<u>Linear Transformation</u>
For vectors x and y, and scalars a and b, it is sufficient to say that a function, F, is a linear transformation if
F(ax+by)=aF(x)+bF(y).

It can be shown that multiplying an m×n matrix, A, and an n×1 vector, v, of compatible size is a linear transformation of v. Therefore from this point forward, a matrix will be synonymous with a linear transformation function.

Let x be a vector and let F(x) be defined by F(x)=Ax where A is a rectangular matrix of appropriate size. Show that F(x) is a linear transformation.
(do this manually)

9.Use Gauss Elimination to solve the following equations. (do this manually)

$$\begin{aligned} 4x_1 + 3x_2 - 5x_3 &= 2 \\ -2x_1 - 4x_2 + 5x_3 &= 5 \\ 8x_1 + 8x_2 &= -3 \end{aligned}$$

10.Use Gauss-Jordan Elimination to solve the above equations.(do this manually)

11. Write a python program to solve the above system of linear equations.

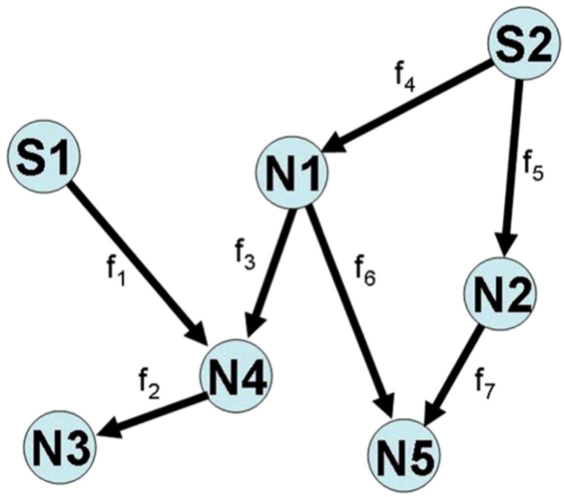
12. Show that matrix multiplication distributes over matrix addition: show A(B+C)=AB+AC assuming that A,B, and C are matrices of compatible size.

13. Write a function my_is_orthogonal(v1,v2, tol), where v1 and v2 are column vectors of the same size and *tol* is a scalar value strictly larger than 0. The output should be 1 if the angle between v1 and v2 is within *tol* of $\pi/2$; that is, $|\pi/2-\theta|<$*tol*, and 0 otherwise. You may assume that v1 and v2 are column vectors of the same size, and that tol is a positive scalar.

14.Write a function my_make_lin_ind(A), where A and B are matrices. Let the rank(A)=n. Then B should be a matrix containing the first n columns of A that are all linearly independent. Note that this implies that B is full rank.

15. Consider the following network consisting of two power supply stations denoted by S1 and S2 and five power recipient nodes denoted by N1 to N5. The nodes are connected by power lines, which are denoted by arrows, and power can flow between nodes along these lines in both directions.

Let di be a positive scalar denoting the power demands for node i, and assume that this demand must be met exactly. The capacity of the power supply stations is denoted by S. Power supply stations must run at their capacity. For each arrow, let fj be the power flow along that arrow. Negative flow implies that power is running in the opposite direction of the arrow.



Write a function my_flow_calculator(S, d), where S is a 1×2 vector representing the capacity of each power supply station, and d is a 1×5 row vector representing the demands at each node (i.e., d[0] is the demand at node 1). The output argument, f, should be a 1×7 row vector denoting the flows in the network (i.e., f[0]=f1 in the diagram). The flows contained in f should satisfy all constraints of the system, like power generation and demands. Note that there may be more than one solution to the system of equations.

The total flow into a node must equal the total flow out of the node plus the demand; that is, for each node i, $f_{inflow}=f_{outflow}+di$. You may assume that $\Sigma S_j=\Sigma d_i$.