MATHEMATICS

LAB ASSIGNMENTS

Set 1

Question 1:

Define a vector

```
import numpy as np

print("Enter the values of vector : ")
list = [int(input()) for i in range(3)]

vector1 = np.array(list)

print("The vector defined is : ",vector1)
```

Output :

Enter the values of vector :

1

2

3

The vector defined is : [1 2 3]

Question 2:

Add two vectors using NumPy Arrays

```
import numpy as np

print("Enter the  values of first vector : ")
list1 = [int(input()) for i in range(3)]

print("Enter the values of second vector : ")
list2 = [int(input()) for i in range(3)]

vector1 = np.array(list1)
vector2 = np.array(list2)

print("The resultant vector is : ", vector1 + vector2)
```

Output :

Enter the values of first vector :

1

2

3

Enter the values of second vector :

4

5

6

The resultant vector is : [5 7 9]

Question 3

Subtract two vectors using NumPy Arrays

```
import numpy as np

print("Enter the  values of first vector : ")
list1 = [int(input()) for i in range(3)]

print("Enter the values of second vector : ")
list2 = [int(input()) for i in range(3)]

vector1 = np.array(list1)
vector2 = np.array(list2)

print("The resultant vector is : ", vector1 - vector2)
```

Output :

Enter the values of first vector :

1

2

3

Enter the values of second vector :

1

4

5

The resultant vector is : [0 -2 -2]

Question 4

Multiply two vectors using NumPy Arrays

```
import numpy as np

print("Enter the  values of first vector : ")
list1 = [int(input()) for i in range(3)]

print("Enter the values of second vector : ")
list2 = [int(input()) for i in range(3)]

vector1 = np.array(list1)
vector2 = np.array(list2)

print("The resultant vector is : ", vector1 * vector2)
```

Output :

Enter the values of first vector :

1

2

3

Enter the values of second vector :

4

5

6

The resultant vector is : [4 10 18]

Question 5

Divide two vectors using NumPy Arrays

```
import numpy as np
print("Enter the  values of first vector : ")
```

```
list1 = [int(input()) for i in range(3)]

print("Enter the values of second vector : ")
list2 = [int(input()) for i in range(3)]

vector1 = np.array(list1)
vector2 = np.array(list2)

print("The resultant vector is : ", vector1 / vector2)
```

Output :

Enter the values of first vector :

1

2

3

Enter the values of second vector :

4

5

6

The resultant vector is : [0.25 0.4   0.5]

Question 6

Find dot product of two vectors

```
import numpy as np

print("Enter the  values of first vector : ")
list1 = [int(input()) for i in range(3)]

print("Enter the values of second vector : ")
list2 = [int(input()) for i in range(3)]

vector1 = np.array(list1)
vector2 = np.array(list2)

resultVector = np.dot(vector1,vector2)

print("The resultant vector is : ", resultVector)
```

Output :

Enter the values of first vector :

1

2

3

Enter the values of second vector :

4

5

6

The resultant vector is : 32

Question 7

Perform vector Scalar Multiplication

```
import numpy as np

print("Enter the  values of first vector : ")
list1 = [int(input()) for i in range(3)]

scalar = int(input("Enter the scalar value"))
vector1 = np.array(list1)

resultVector = vector1 * scalar

print("The resultant vector is : ", resultVector)
```

Output :

Enter the values of first vector :

1

2

3

Enter the scalar value : 4

The resultant vector is : [4 8 12]

Question 8

Calculate L1, L2, Max Norms of a vector.

```
import numpy as np
from numpy.linalg import norm
from math import inf

print("Enter the values of  vector : ")
list1 = [int(input()) for i in range(3)]

vector1 = np.array(list1)

resultVector1 = norm(list1, 1)
resultVector2 = norm(list1, 2)
resultVector3 = norm(list1, inf)

print("The L1 norm of vector is : ", resultVector1)
print("The L2 norm of vector is : ", resultVector2)
print("The Max norm of vector is : ", resultVector3)
```

Output :

Enter the values of vector :

1

2

3

The L1 norm of vector is : 6.0

The L2 norm of vector is : 3.7416573867739413

The Max norm of vector is : 3.0

Set-2

Question 1

Define Matrix

```
rows = int(input("enter the number of rows in Matrix "))
columns = int(input("enter the number of columns in Matrix "))

print(f"Enter the elements of the {rows}x{columns} matrix")
a = [[int(input("=>")) for j in range(columns)] for i in range(rows)]

#To print matrix in conventional order of matrix
for row in a :
  for element in row:
    print(element,end=" ")
  print('')
```

Output :

Enter the elements of the 3x3 matrix

=>1

=>2

=>3

=>4

=>5

=>6

=>7

=>8

=>9

1 2 3

4 5 6

7 8 9

Question 2

Add two matrices

```
import numpy as np
rows = int(input("Enter the number of rows in  Matrices : "))
```

```
columns = int(input("Enter the number of columns in Matrices : "))

print(f"Enter the elements of the first {rows}x{columns} matrix")
matrix_1 = [[int(input("=>")) for j in range(columns)] for i in range(rows)]

print(f"Enter the elements of the second {rows}x{columns} matrix")
matrix_2 = [[int(input("=>")) for j in range(columns)] for i in range(rows)]

matrix_1 = np.array(matrix_1)
matrix_2 = np.array(matrix_2)

print("first Matrix \n",matrix_1)
print("Second Matrix \n", matrix_2)
print("Sum of Matrices\n",matrix_1 + matrix_2)
```

Output :

Enter the elements of the first 2x2 matrix

=>1

=>2

=>3

=>4

Enter the elements of the second 2x2 matrix

=>5

=>6

=>7

=>8

first Matrix

[[1 2]

[3 4]]

Second Matrix

[[5 6]

[7 8]]

Sum of Matrices

[[6 8]

[10 12]]

Question 3

Subtract two matrices

```
rows = int(input("Enter the number of rows in  Matrices : "))
columns = int(input("Enter the number of columns in Matrices : "))

print(f"Enter the elements of the first {rows}x{columns} matrix")
matrix_1 = [[int(input("=>")) for j in range(columns)] for i in range(rows)]

print(f"Enter the elements of the second {rows}x{columns} matrix")
matrix_2 = [[int(input("=>")) for j in range(columns)] for i in range(rows)]

matrix_1 = np.array(matrix_1)
matrix_2 = np.array(matrix_2)

print("first Matrix \n",matrix_1)
print("Second Matrix \n", matrix_2)
print("Difference of Matrices\n",matrix_1 - matrix_2)
```

Output :

Enter the number of rows in Matrices : 2

Enter the number of columns in Matrices : 2

Enter the elements of the first 2x2 matrix

=>5

=>6

=>7

=>8

Enter the elements of the second 2x2 matrix

=>1

=>2

=>3

=>4

first Matrix

[[5 6]

[7 8]]

Second Matrix

[[1 2]

[3 4]]

Difference of Matrices

[[4 4]

[4 4]]

Question 4

Find the hadamard product of two matrices

```
import numpy as  np

rows = int(input("Enter the number of rows in  Matrices : "))
columns = int(input("Enter the number of columns in Matrices : "))

print(f"Enter the elements of the first {rows}x{columns} matrix")
matrix_1 = [[int(input("=>")) for j in range(columns)] for i in range(rows)]

print(f"Enter the elements of the second {rows}x{columns} matrix")
matrix_2 = [[int(input("=>")) for j in range(columns)] for i in range(rows)]

matrix_1 = np.array(matrix_1)
matrix_2 = np.array(matrix_2)

print("first Matrix \n",matrix_1)
print("Second Matrix \n", matrix_2)
print("Hadamard product of two matrices are:\n", matrix_1 * matrix_2)
```

Output :

Enter the number of rows in Matrices : 2

Enter the number of columns in Matrices : 2

Enter the elements of the first 2x2 matrix

=>1

=>2

=>3

=>4

Enter the elements of the second 2x2 matrix

=>5

=>6

=>7

=>8

first Matrix

[[1 2]

[3 4]]

Second Matrix

[[5 6]

[7 8]]

Hadamard product of two matrices are:

[[5 12]

[21 32]]

Question 5

Divide two matrices

```
import numpy as  np


rows = int(input("Enter the number of rows in  Matrices : "))
columns = int(input("Enter the number of columns in Matrices : "))

print(f"Enter the elements of the first {rows}x{columns} matrix")
matrix_1 = [[int(input("=>")) for j in range(columns)] for i in range(rows)]

print(f"Enter the elements of the second {rows}x{columns} matrix")
matrix_2 = [[int(input("=>")) for j in range(columns)] for i in range(rows)]

matrix_1 = np.array(matrix_1)
matrix_2 = np.array(matrix_2)

print("first Matrix \n",matrix_1)
print("Second Matrix \n", matrix_2)

print("Resultant matrix after division is:\n", matrix_1 / matrix_2)
```

Output :

Enter the number of rows in Matrices : 2

Enter the number of columns in Matrices : 2

Enter the elements of the first 2x2 matrix

=>9

=>8

=>7

=>6

Enter the elements of the second 2x2 matrix

=>5

=>4

=>3

=>2

first Matrix

[[9 8]

[7 6]]

Second Matrix

[[5 4]

[3 2]]

Resultant matrix after division is:

[[1.8 2.]

[2.33333333 3.]]

Question 6

Find the product of two matrices

```
import numpy as np

rows = int(input("Enter the number of rows in first Matrix : "))
columns = int(input("Enter the number of columns in first Matrix : "))

column_2 = int(input("Enter the number of columns in Second Matrix : "))

print(f"Enter the elements of the first {rows} x {columns} matrix")
matrix_1 = [[int(input("=>")) for j in range(columns)] for i in range(rows)]

print(f"Enter the elements of the second {columns} x {column_2} matrix")
matrix_2 = [[int(input("=>")) for j in range(column_2)] for i in range(columns)]

matrix_1 = np.array(matrix_1)
matrix_2 = np.array(matrix_2)

print("first Matrix \n",matrix_1)
print("Second Matrix \n", matrix_2)

result = []
```

```
for i in range(rows):
  for j in range(column_2):
    sum = 0
    for k in range(columns):
        sum += matrix_1[i][k] * matrix_2[k][j]
    result.append(sum)
result = np.array(result)
result = result.reshape(rows, column_2)

print("Resultant matrix after multiplication is:\n", result)
```

Output :

Enter the number of rows in first Matrix : 2

Enter the number of columns in first Matrix : 2

Enter the number of columns in Second Matrix : 2

Enter the elements of the first 2 x 2 matrix

=>1

=>2

=>3

=>4

Enter the elements of the second 2 x 2 matrix

=>5

=>6

=>7

=>8

first Matrix

[[1 2]

[3 4]]

Second Matrix

[[5 6]

[7 8]]

Resultant matrix after multiplication is:

[[19 22]

[43 50]]

Question 7

Find vector matrix multiplication

```
import numpy as np
from time import sleep

def MatrixMaker(rows, columns):
  a = [int(input("=>")) for i in range(rows * columns)]
  a = np.array(a)
  a = a.reshape(rows, columns)
  return a

rows = int(input("Enter the number of rows in Matrix : "))
columns = int(input("Enter the number of columns in Matrix : "))

if rows > 3 or columns > 3 :
  sleep(2)
  print("Vectors can only have maximum elements of 3")
  sleep(2)
  print("Please rerun the program")
  exit()

column_2 = 1 #defining the columns in a vector space

print(f"Enter the elements of the {rows} x {columns} matrix: ")
matrix_1 = MatrixMaker(rows, columns)

print(f"Enter the elements of the Vector: ")
matrix_2 = MatrixMaker(columns, column_2)

print("Matrix \n",matrix_1)
print("Vector \n", matrix_2)

result = []

for i in range(rows):
    sum = 0
    column_2 -= 1 # for adjusting into the index of matrix
    for k in range(columns):
        sum += matrix_1[i][k] * matrix_2[k][column_2]
    result.append(sum)
result = np.array(result)
result = result.reshape(rows, column_2)

print("Resultant matrix after multiplication is:\n", result)
```

Output :

Enter the number of rows in Matrix : 2

Enter the number of columns in Matrix : 2

Enter the elements of the 2 x 2 matrix:

=>1

=>2

=>3

=>4

Enter the elements of the Vector:

=>5

=>6

Matrix

[[1 2]

[3 4]]

Vector

[[5]

[6]]

Resultant matrix after multiplication is:

[[17]

[39]]

Question 8

Perform scalar - matrix multiplication

```
import numpy as np

def MatrixMaker(rows, columns):
  a = [int(input("=>")) for i in range(rows * columns)]
  a = np.array(a)
  a = a.reshape(rows, columns)
  return a

rows = int(input("Enter the number of rows in  Matrices : "))
columns = int(input("Enter the number of columns in Matrices : "))

print(f"Enter the elements of the first {rows}x{columns} matrix")
matrix_1 = MatrixMaker(rows, columns)
```

```
scalar_value = int(input("Enter any scalar value : "))

print("first matrix \n", matrix_1)
print("Scalar value \n", scalar_value)

print("Resultant matrix is \n",matrix_1 * scalar_value)
```

Output :

Enter the number of rows in Matrices : 2
Enter the number of columns in Matrices : 2
Enter the elements of the first 2x2 matrix
=>1
=>2
=>3
=>4
Enter any scalar value : 7
first matrix
 [[1 2]
 [3 4]]
Scalar value
 7
Resultant matrix is
 [[7 14]
 [21 28]]

Question 9

Define a 3 x 3 square matrix and Calculate lower and upper triangular matrix from it.

```
import numpy as np

def MatrixMaker(rows, columns):
  a = [int(input("=>")) for i in range(rows * columns)]
  a = np.array(a)
  a = a.reshape(rows, columns)
  return a

rows = 3
```

```
columns = 3

print(f"Enter the elements of the first {rows}x{columns} matrix")
matrix_1 = MatrixMaker(rows, columns)

#upper triangular matrix
upperTriangle = []
upperTriangle = matrix_1.copy()
for i in range(rows):
  for j in range(columns):
    if i > j :
      upperTriangle[i][j] = 0

#lower triangular matrix
lowerTriangle = []
lowerTriangle = matrix_1.copy()
for i in range(rows):
  for j in range(columns):
    if i < j:
      lowerTriangle[i][j] = 0

print("Matrix \n", matrix_1)
print("Upper triangular matrix \n",upperTriangle)
print("Lower triangular matrix \n",lowerTriangle)
```

Output :

Enter the elements of the first 3x3 matrix

=>1

=>2

=>3

=>4

=>5

=>6

=>7

=>8

=>9

Matrix

[[1 2 3]

[4 5 6]

[7 8 9]]

Upper triangular matrix

[[1 2 3]

[0 5 6]

[0 0 9]]

Lower triangular matrix

[[1 0 0]

[4 5 0]

[7 8 9]]

Question 10

Define a 3 x 3 square matrix, Extract the main diagonal as vector. Create diagonal matrix from that extracted vector

```
import numpy as np

def MatrixMaker(rows, columns):
  a = [int(input("=>")) for i in range(rows * columns)]
  a = np.array(a)
  a = a.reshape(rows, columns)
  return a

rows = 3
columns = 3

print(f"Enter the elements of the {rows}x{columns} matrix")
matrix_1 = MatrixMaker(rows, columns)

# extracting vector and making diagonal matrix
diagonalMatrix = []

#extracting vector using loop that  only iterates the diagonal elements
vector = [matrix_1[i][j] for i, j in zip(range(rows) ,range(columns))]
```

```
vector = np.array(vector) #converting to vector

#making diagonal matrix using vector
for i in range(rows):
  for j in range(columns):
    if i == j :  diagonalMatrix.append(vector[i])
    else : diagonalMatrix.append(0)
#You can use a single line code for making diagonal matrix
#diagonalMatrix = [matrix[i][j] if i==j else 0  for i, j in  [(i, j) for i in
range(rows) for j in range(columns)]]
diagonalMatrix = np.array(diagonalMatrix)
diagonalMatrix = diagonalMatrix.reshape(rows, columns)

#printing all values
print("Matrix \n", matrix_1)
print("Vector \n",vector)
print("Diagonal Matrix \n",diagonalMatrix)
```

Output :

Enter the elements of the 3x3 matrix

=>1

=>2

=>3

=>4

=>56

=>6

=>7

=>8

=>9

Matrix

[[1 2 3]

[4 56 6]

[7 8 9]]

Vector

[1 56 9]

Diagonal Matrix

[[1 0 0]

[0 56 0]

[0 0 9]]

Question 11

Create an identity matrix of order 4

```
import numpy as np

rows = 4
columns = 4
identityMatrix = []

for i in range(rows):
    for j in range(columns):
        if i == j :
            identityMatrix.append(1)
        else :
            identityMatrix.append(0)

identityMatrix = np.array(identityMatrix)
identityMatrix = identityMatrix.reshape(rows,columns)
print(identityMatrix)
```

Output :

[[1 0 0 0]

[0 1 0 0]

[0 0 1 0]

[0 0 0 1]]

Question 12

Find transpose of a matrix

```
import numpy as np

def MatrixMaker(rows, columns):
  a = [int(input("=>")) for i in range(rows * columns)]
  a = np.array(a)
  a = a.reshape(rows, columns)
  return a
```

```
rows = int(input("Enter the number of rows in  Matrix : "))
columns = int(input("Enter the number of columns in Matrix : "))

print(f"Enter the elements of the first {rows}x{columns} matrix")
matrix_1 = MatrixMaker(rows, columns)

transpose = []
for i in range(columns):
    for j in range(rows):
        transpose.append(matrix_1[j][i])
transpose = np.array(transpose)
transpose = transpose.reshape(columns,rows)

print("Matrix :", matrix_1, sep="\n")
print("Transpose of matrix :", transpose, sep="\n")
```

Output :

Enter the number of rows in Matrix : 2

Enter the number of columns in Matrix : 2

Enter the elements of the first 2x2 matrix

=>1

=>2

=>3

=>4

Matrix :

[[1 2]

 [3 4]]

Transpose of matrix :

[[1 3]

 [2 4]]

Question 13

Print the inverse of a matrix

```
import numpy as np
from numpy.linalg import inv, det

dimension = int(input("Enter the no.of rows or columns in Matrix : "))
print('Enter the values of matrix')
```

```
matrix = [int(input()) for i in range(dimension**2)]
matrix = np.array(matrix)
matrix = matrix.reshape(dimension, dimension)

determinant = det(matrix)

if determinant == 0:
  inverse = 'does not exist'
else:
  inverse = inv(matrix)
print("Matrix : \n", matrix)
print("Inverse of Matrix : \n", inverse)
```

Output :

Enter the no.of rows or columns in Matrix : 2
Enter the values of matrix
1
2
3
4
Matrix :
[[1 2]
[3 4]]
Inverse of Matrix :
[[-2. 1.]
[1.5 -0.5]]

Question 14

Print the determinant of the matrix

```
import numpy as np
from math import sqrt

def determinant2D(matrix):
  determinant = 0
  diagonal1 = 1
  diagonal2 = 1
```

```
  for i in range(2):
    for j in range(2):
      if i == j :
        diagonal1 *= matrix[i][j]
      else :
        diagonal2 *= matrix[i][j]

  determinant = diagonal1 - diagonal2
  return determinant

def determinantOfMatrix(matrix, dimension):
  if(dimension < 3):
    determinant = determinant2D(matrix)
  else:
    determinant = 0
    for k in range(len(matrix[0])):
      array = []
      for i in range(dimension):
        for j in range(dimension):
            if i == 0 or j == k:
              continue
            else :
              array.append(matrix[i][j])
      array = np.array(array)
      dimension2 = int(sqrt(len(array)))
      array = array.reshape(dimension2, dimension2)
      if k % 2 == 0 :
        determinant += matrix[0][k] * determinantOfMatrix(array, dimension2)
      else :
        determinant -= matrix[0][k] * determinantOfMatrix(array, dimension2)
  return determinant

dimension = int(input("Enter the no.of rows or columns in Matrix : "))
print('Enter the values of matrix')
matrix = [int(input()) for i in range(dimension**2)]
matrix = np.array(matrix)
matrix = matrix.reshape(dimension, dimension)

print("Elements of matrix:\n",matrix)
print("determinant of matrix:\n",determinantOfMatrix(matrix, dimension))
```

Output :

Enter the no.of rows or columns in Matrix : 2

Enter the values of matrix

1
2
3
4
Elements of matrix:
[[1 2]
[3 4]]
determinant of matrix:
-2

Set 3

Question 1

Create an orthogonal matrix and check $Q^T * Q = Q * Q^T =$ Identity Matrix

```
import numpy as np


def product(matrix_1, matrix_2):
  product = []

  row = len(matrix_1)
  column1 = len(matrix_1[0])
  column2 = len(matrix_2[0])

  for i in range(row):
    for j in range(column2):
      sum = 0
      for k in range(column1):
          sum += matrix_1[i][k] * matrix_2[k][j]
      product.append(sum)

  product = np.array(product)
  product = product.reshape(row, column2)

  return product


def transpose(matrix):
  transpose = []

  rows = len(matrix)
  columns = len(matrix[0])

  for i in range(columns):
      for j in range(rows):
          transpose.append(matrix[j][i])

  transpose = np.array(transpose)
  transpose = transpose.reshape(columns,rows)
  return transpose


matrix = np.array([
```

```
    [1/3,2/3, -2/3],
    [-2/3,2/3,1/3],
    [2/3,1/3,2/3]])
print("Matrix :", matrix, sep="\n")

transpose = transpose(matrix)
print("\nTranspose of matrix :", transpose, sep="\n")

identityMatrix = np.array([
  [1, 0, 0],
  [0, 1, 0],
  [0, 0, 1]])

print("\nQ * Qtranspose : ", product(matrix, transpose), sep='\n')
print("\nQtranspose * Q : ", product(transpose, matrix), sep='\n')
print("\nHere we can see that Q * Qtranspose = Qtranspose * Q = ",
identityMatrix, sep='\n')
```

Output :

Matrix :

[[0.33333333 0.66666667 -0.66666667]

 [-0.66666667 0.66666667 0.33333333]

 [0.66666667 0.33333333 0.66666667]]

Transpose of matrix :

[[0.33333333 -0.66666667 0.66666667]

 [0.66666667 0.66666667 0.33333333]

 [-0.66666667 0.33333333 0.66666667]]

Q * Qtranspose :

[[1. 0. 0.]

 [0. 1. 0.]

 [0. 0. 1.]]

Qtranspose * Q :

[[1. 0. 0.]

[0. 1. 0.]

[0. 0. 1.]]

Here we can see that Q * Qtranspose = Qtranspose * Q =

[[1 0 0]

[0 1 0]

[0 0 1]]

Question 2

Print Rank of a matrix

```
import numpy as np

order = 3

print("Enter elements in matrix : ")
matrix = [[int(input("=>")) for j in range(order)] for i in range(order)]
matrix = np.array(matrix)

print("Matrix : ", matrix, sep='\n')

rank = np.linalg.matrix_rank(matrix)
print("Rank of matrix is : ", rank)
```

Output :

Enter elements in matrix :

=>1

=>2

=>3

=>4

=>5

=>6

=>7

=>8

=>8

Matrix :

[[1 2 3]

[4 5 6]

[7 8 8]]

Rank of matrix is : 3

Question 3

Calculate sparsity of a matrix

```
import numpy as np

def sparcity0fMatrix(matrix):

  count = 0
  for row in matrix:
    for element in row :
      if element == 0:
         count += 1

  rows = len(matrix)
  columns = len(matrix[0])
  TotalElements = rows * columns

  sparcity = count / TotalElements

  return sparcity

rows = int(input("Enter the number of rows : "))
columns = int(input("Enter the number of columns : "))

print("Enter elements in matrix : ")
matrix = [[int(input("=>")) for j in range(columns)] for i in range(rows)]
matrix = np.array(matrix)

sparcity = sparcity0fMatrix(matrix)
print("\nSparcity of given matrix is : ", sparcity)

if sparcity > 0.5 :
  print("The given Matrix is a sparse matrix")
else :
  print("The given matrix is not sparse matrix")
```

Output :

Enter the number of rows : 2

Enter the number of columns : 2

Enter elements in matrix :

=>0

=>1

=>0

=>0

Sparcity of given matrix is : 0.75

The given Matrix is a sparse matrix

Question 4

Print Eigen Values and eigen vectors of a matrix

```
import numpy as np
from numpy.linalg import eig

order = 3
print("Enter the elements in the matrix")
matrix = np.array([[int(input("=>")) for j in range(order)] for i in
range(order)])
print(matrix)


eigenValue, eigenVector = eig(matrix)
eigenValue = np.array(eigenValue)
eigenVector = np.array(eigenVector)

print("Eigen value of a matrix : ", eigenValue, sep='\n')
print("Eigen vector of a matrix : ", eigenVector, sep='\n')
```

Output :

Enter the elements in the matrix

=>1

=>2

=>3

=>0

=>5

=>6

=>7

=>8

=>9

[[1 2 3]

[0 5 6]

[7 8 9]]

Eigen value of a matrix :

[15.54400375 -1.54400375 1.   ]

Eigen vector of a matrix :

[[-0.24005684 -0.32012138 0.30215583]

[-0.48011368 -0.64024277 -0.79315905]

[-0.84372008 0.69829184 0.5287727]]

Question 5

Print Eigen Values and eigen vectors of a matrix and reconstruct the matrix

```
import numpy as np
from numpy import dot, diag
from numpy.linalg import eig, inv

order = 3
print("Enter the elements in the matrix")
matrix = np.array([[int(input("=>")) for j in range(order)] for i in
range(order)])
print(matrix)

eigenValue, eigenVector = eig(matrix)

InverseEigen = inv(eigenVector)

vectorDiagonal = diag(eigenValue)

rematrix = eigenVector.dot(vectorDiagonal).dot(InverseEigen)

print("Reconstructed matrix is :", rematrix, sep="\n")
```

Output :

Enter the elements in the matrix

=>1

=>2

=>3

=>4

=>5

=>6

=>7

=>8

=>9

[[1 2 3]

 [4 5 6]

 [7 8 9]]

eigen value of matrix is :

[1.61168440e+01 -1.11684397e+00 -1.30367773e-15]

eigen vector of matrix is :

[[-0.23197069 -0.78583024 0.40824829]

 [-0.52532209 -0.08675134 -0.81649658]

 [-0.8186735   0.61232756 0.40824829]]

Reconstructed matrix is :

[[1. 2. 3.]

 [4. 5. 6.]

 [7. 8. 9.]]

Question 6

Define 5 * 2 matrix data set, split it into x and y components and plot dataset as scatterplot.

```
import matplotlib.pyplot as plt
import numpy as np
```

```
print("Enter the elements in the matrix")
matrix = np.array([[int(input("=>")) for j in range(2)] for i in range(5)])

x,y=np.split(matrix,2,axis=1)

plt.scatter(x, y)
plt.show()
```

Output :

Enter the elements in the matrix

=>1

=>2

=>3

=>4

=>5

=>6

=>7

=>8

=>9

=>12