# Program - 3

**AIM** -Create/Define single dimension / multi-dimension arrays, and arrays with specific values like array of all ones, all zeros, array with random values within a range, or a diagonal matrix.

**SOFTWARE USED** - VS Code

**Description:** The code uses NumPy functions to generate these arrays. NumPy is a popular library for numerical operations in Python and is especially useful for creating and manipulating arrays. These examples demonstrate different ways to create arrays with specific values and shapes to suit various data processing needs.

### SOURCE CODE:

import numpy as np import numpy as np

# Create/Define single dimension / multi-dimension arrays, and arrays with specific values like array of all ones, all zeros, array with random values within a range, or a diagonal matrix.

# Single Dimensional

arr\_1d = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

# Multi Dimensional

arr\_4d = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])

#Array of ones

arr\_ones = np.ones((3, 3))

# Arrays of zeros

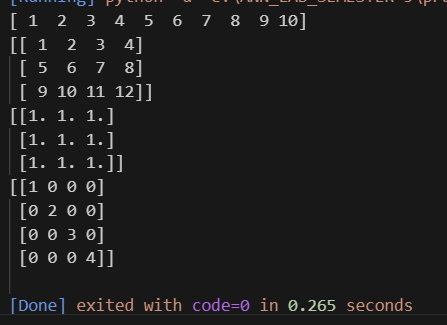
arr\_zeros = np.zeros((3, 3))

# Array with random values arr\_rand = np.random.rand(3, 3)

# Diagonal Matrix

arr\_diag = np.diag([1, 2, 3, 4]) print(arr\_diag)

**OUTPUT:**



# Program - 4

**AIM** - Use command to compute the size of a matrix, size/length of a particular row/column, load data from a text file, store matrix data to a text file, finding out variables and their features in the current scope

**SOFTWARE USED** - VS Code

### SOURCE CODE:

import numpy as np

# compute the size pf matrix

matrix = np.array([[1,2,3],[4,5,6],[7,8,9]])

matrix\_size = matrix.size

#get the size/length of particular row and column

num\_rows,num\_cols=matrix.shape

row\_length = len(matrix[ : ])

#length of first row

col\_length = len(matrix[ : ])

#length of second column

#store matrix data to a text file

matrix\_to\_store = np.array([[10,20,30],[40,50,60],[70,80,90]])

np.savetxt("matrix\_data.txt",matrix\_to\_store)

#finding out variable & their features in the current scope

#display result

print("size of the matrix",matrix\_size)

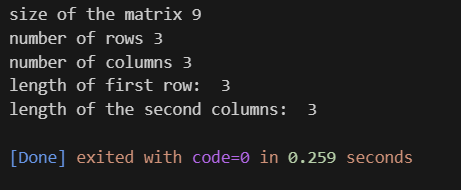
print("number of rows",num\_rows)

print("number of columns",num\_cols)

print("length of first row: ",row\_length)

print("length of the second columns: ",col\_length)

**OUTPUT:**



# Program - 5

**AIM** - Write a program to generate XOR function using McCulloch-Pitt’s neuron and appropriate values for weights, bias, and threshold.

**SOFTWARE USED** - VS Code

### SOURCE CODE:

# McCulloch-Pitts Neuron Function

def mcculloch\_pitts\_neuron(inputs, weights, bias, threshold):

    # Calculate weighted sum

    weighted\_sum = sum([inputs[i] \* weights[i] for i in range(len(inputs))]) + bias

    # Determine the output based on the threshold

    output = 1 if weighted\_sum >= threshold else 0

    return output

# XOR Truth Table Input Data

input\_data = [[0, 0], [0, 1], [1, 0], [1, 1]]

# Define appropriate values for Weights, Bias & Threshold for XOR

weights = [1, 1]

bias = -1

threshold = 1

# Evaluate the XOR function for each input

for inputs in input\_data:

    result = mcculloch\_pitts\_neuron(inputs, weights, bias, threshold)

    print(f"Input: {inputs}, Output: {result}")

**OUTPUT:**

