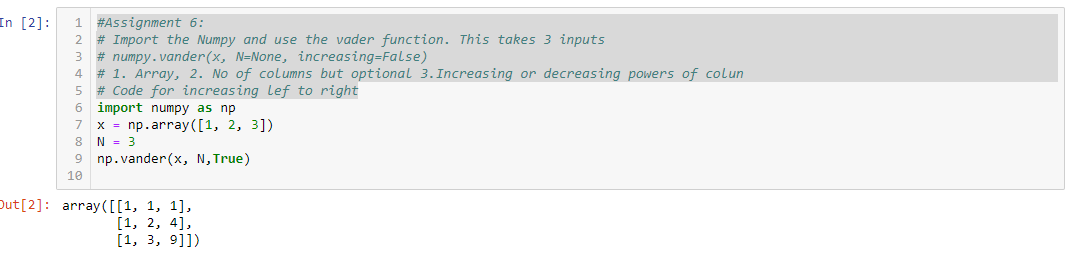
Write a function so that the columns of the output matrix are powers of the input vector.

The order of the powers is determined by the increasing boolean argument. Specifically, when increasing is False, the i-th output column is the input vector raised element-wise to the power of N - i - 1.

HINT: Such a matrix with a geometric progression in each row is named for Alexandre Theophile Vandermonde.

Note: Solution submitted via github must contain all the detailed steps.



#Assignment 6:

# Import the Numpy and use the vader function. This takes 3 inputs

# numpy.vander(x, N=None, increasing=False)

# 1. Array, 2. No of columns but optional 3.Increasing or decreasing powers of colun

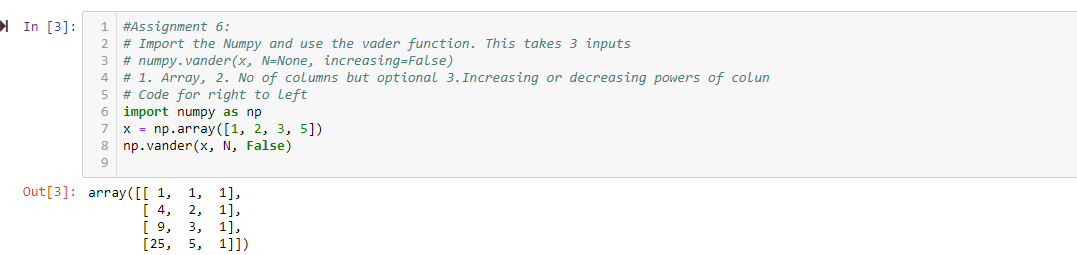
# Code for increasing lef to right

import numpy as np

x = np.array([1, 2, 3])

N = 3

np.vander(x, N,True)



#Assignment 6:

# Import the Numpy and use the vader function. This takes 3 inputs

# numpy.vander(x, N=None, increasing=False)

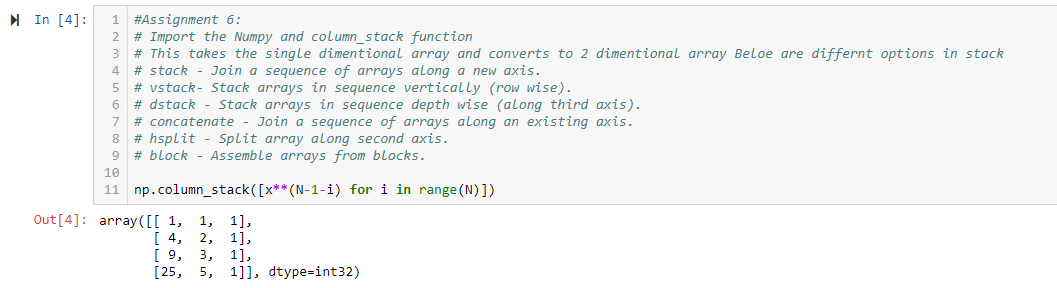
# 1. Array, 2. No of columns but optional 3.Increasing or decreasing powers of colun

# Code for right to left

import numpy as np

x = np.array([1, 2, 3, 5])

np.vander(x, N, False)



#Assignment 6:

# Import the Numpy and column\_stack function

# This takes the single dimentional array and converts to 2 dimentional array Beloe are differnt options in stack

# stack - Join a sequence of arrays along a new axis.

# vstack- Stack arrays in sequence vertically (row wise).

# dstack - Stack arrays in sequence depth wise (along third axis).

# concatenate - Join a sequence of arrays along an existing axis.

# hsplit - Split array along second axis.

# block - Assemble arrays from blocks.

np.column\_stack([x\*\*(N-1-i) for i in range(N)])