**SOURCE CODE:**

**Method 1: With use of vander function**

import numpy as np # Import Numpy

x = np.array([1, 2, 3,4, 5]) # Input matrix

increase = int(input('Enter in which order the matrix should be generated, Increase is True(1) or False(2)'))

def Generate\_vander(x,increase): # Function to generate matrix

if increase is 1:

a = np.vander(x, increasing=True) # Using Vander function with Increasing bolean object as True

elif increase is 2:

a = np.vander(x, increasing=False) # Using Vander function with Increasing bolean object as False

return (a) # return output

Generate\_vander(x,increase) # call function

**Method 2: Without using vander function**

enerate Vandermonde matrix without using vander function

import numpy as np # Import Numpy

N=int(input('Enter how many coluns to be added in matrix')) # Get how many columns to be generated

# Get Increase value True or False

increase = int(input('Enter in which order the matrix should be generated, Increase is True(1) or False(2)'))

x = np.array([1, 2, 3, 4, 5]) # Input array

def Generate\_matrix(N,increase): # Function to generate matrix

if order is 1: # If increase is True

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

elif order is 2: # If increase is False

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

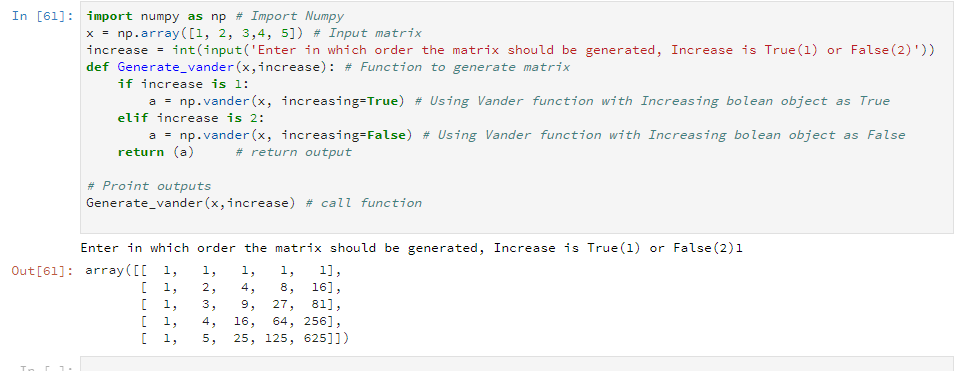
return y # return output

Generate\_matrix(N,increase) # call function

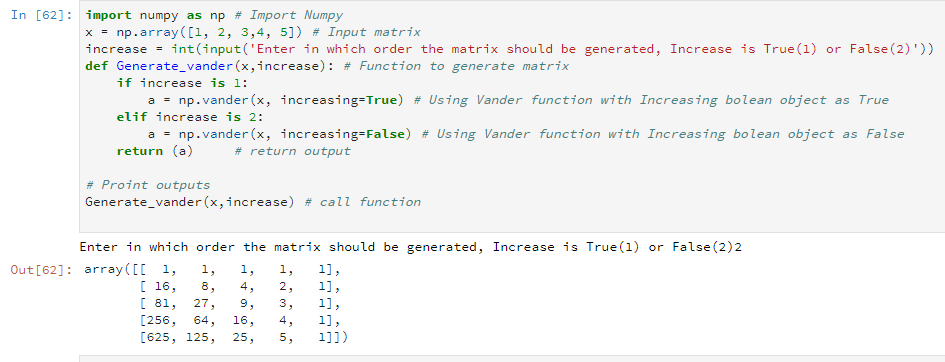
**OUTPUT:**

**Method 1 (With use of vander function):**

**Increase is True:**

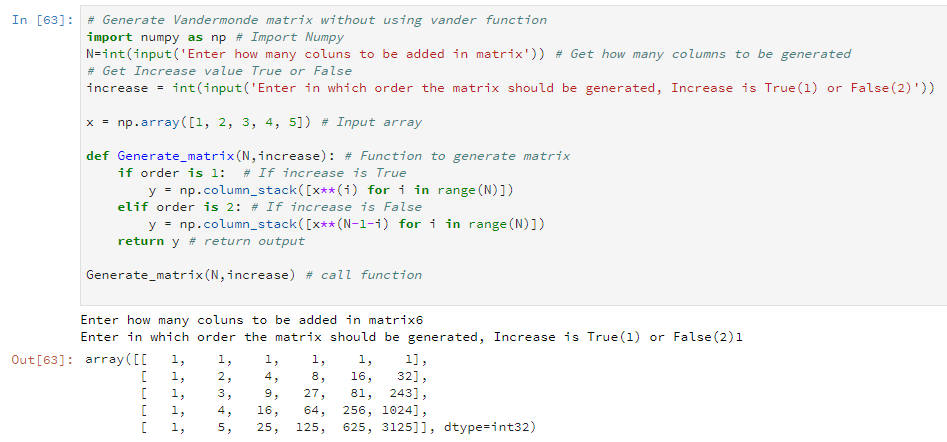


**Increase is False:**



**Method 2: Without using vander function**

**Increase is True:**



**Increase is False:**

