Problem Statement 1:

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.

```
In [1]: import numpy as np
        x = np.array([1, 2, 3,4, 5])
        print(x)
        x
        N = 5
        z = np.vander(x, N)
        print(z)
        np.column_stack([x**(N-1-i) for i in range(N)])
        #x = np.array([1, 2, 3, 5])
        np.vander(x)
        np.vander(x, increasing=True)
```

```
[1 2 3 4 5]
[[  1   1   1   1   1]
 [ 16   8   4   2   1]
 [ 81  27   9   3   1]
 [256  64  16   4   1]
 [625 125  25   5   1]]
```

```
Out[1]: array([[  1,   1,   1,   1,   1],
               [  1,   2,   4,   8,  16],
               [  1,   3,   9,  27,  81],
               [  1,   4,  16,  64, 256],
               [  1,   5,  25, 125, 625]])
```