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.

In [2]:
```
import numpy as np

#function to create Alexandre-Theophile Vandermonde matrix
def Vandermonde(list, N, flag = True):
    return np.vander(list, N, increasing = flag)

x = np.array([1, 2, 3, 4])
print("Alexandre-Theophile Vandermonde matrix when increasing is True:\n",Vandermonde(x, 4 , True))
print("")
print("Alexandre-Theophile Vandermonde matrix when increasing is False:\n", Vandermonde(x, 4, False))
print("")
print("Alexandre-Theophile Vandermonde matrix when increasing is False:\n", Vandermonde(x, 3, False))
```

```
Alexandre-Theophile Vandermonde matrix when increasing is True:
 [[ 1  1  1  1]
 [ 1  2  4  8]
 [ 1  3  9 27]
 [ 1  4 16 64]]

Alexandre-Theophile Vandermonde matrix when increasing is False:
 [[ 1  1  1  1]
 [ 8  4  2  1]
 [27  9  3  1]
 [64 16  4  1]]

Alexandre-Theophile Vandermonde matrix when increasing is False:
 [[ 1  1  1]
 [ 4  2  1]
 [ 9  3  1]
 [16  4  1]]
```