In [ ]:

```
# 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 [4]:

```
#creating a matrix with output columns based on input vector(N)

import numpy as np
x=np.array([1,2,3,4,5])
N=5
matrix = np.column_stack([x**(N-i-1) for i in range(N)])
print(matrix)
```

```
[[  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]]
```

In [2]:

```
#Similar matrix is created using np.vander function in numpy

import numpy as np
x=np.array([1,2,3,4,5])
N=5
matrix1 = np.vander(x,N,increasing=False)
print(matrix1)
```

```
[[  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]]
```

In [9]:

```
# The determitant for both the matrixs is the same
np.linalg.det(matrix)
```

Out[9]:

```
287.99999999999517
```

In [8]:

```
np.linalg.det(matrix1)
```

Out[8]:

```
287.99999999999517
```