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

In [1]:
```
import numpy as np
def vander1(x, N=None, increasing=False):
    x = np.asarray(x)
    if x.ndim != 1:
        raise ValueError("x must be a one-dimensional array or sequence.")
    if N is None:
        N = len(x)
    v=np.array([],dtype=int)
    for i in range(N):
        a_pow=x**i
        if increasing:
            v=np.append(v,a_pow)
            #v=np.vstack([a_pow,v])
        else:
            v=np.append(a_pow,v)
            #v=np.vstack([v,a_pow])
    v=v.reshape((N,len(x))).transpose()
    return v
```

In [2]:
```
x=[1,2,3,5]
N=3
vander1(x,N)
```

Out[2]:
```
array([[ 1,  1,  1],
       [ 4,  2,  1],
       [ 9,  3,  1],
       [25,  5,  1]])
```

In [ ]:
```
```