In [16]:
```
import numpy as np
```

In [21]:
```
def AlexTheoVander(iVector, n, increasing = False):
    if increasing:
        oMatrix = np.matrix([x**i for x in iVector for i in range(n)]).reshape(iVector.size, n)
    else:
        oMatrix = np.matrix([x**(n-i-1) for x in iVector for i in range(n)]).reshape(iVector.size, n)
    return oMatrix
```

In [22]:
```
iVector = np.array([2, 4, 6, 8, 10])
n = 5
```

In [23]:
```
oMatrix_asc = AlexTheoVander(iVector, n, True)
oMatrix_asc
```

Out[23]:
```
matrix([[    1,     2,     4,     8,    16],
        [    1,     4,    16,    64,   256],
        [    1,     6,    36,   216,  1296],
        [    1,     8,    64,   512,  4096],
        [    1,    10,   100,  1000, 10000]])
```

In [24]:
```
oMatrix_asc = alexTheoVander(iVector, n, False)
oMatrix_asc
```

Out[24]:
```
matrix([[   16,     8,     4,     2,     1],
        [  256,    64,    16,     4,     1],
        [ 1296,   216,    36,     6,     1],
        [ 4096,   512,    64,     8,     1],
        [10000,  1000,   100,    10,     1]])
```

In [32]:
```
def MovingAverage(inputvalue, k):
    z = 1
    output = np.convolve(inputvalue, np.ones(k), 'valid')/k
    for i in output:
        print("y{0} = {1: .2f}". format(z, i))
        z += 1
```

In [33]:
```
inputvalue = np.array([3, 5, 7, 2, 8, 10, 11, 65, 72, 81, 99, 100, 150])
k = 3
MovingAverage(inputvalue, k)
```

```
y1 =  5.00
y2 =  4.67
y3 =  5.67
y4 =  6.67
y5 =  9.67
y6 =  28.67
y7 =  49.33
y8 =  72.67
y9 =  84.00
y10 =  93.33
y11 =  116.33
```

In [ ]:
```

```