Given a sequence of n values x1, x2, ..., xn and a window size k>0, the k-th moving average of the given sequence is defined as follows:

The moving average sequence has n-k+1 elements as shown below. The moving averages with k=4 of a ten-value sequence (n=10) is shown below i 1 2 3 4 5 6 7 8 9 10 ===== == == == == == == i 1 2 3 4 5 6 7 8 9 10 ===== == == == == == == == == == == Input 10 20 30 40 50 60 70 80 90 100 y1 25 = (10+20+30+40)/4 y2 35 = (20+30+40+50)/4 y3 45 = (30+40+50+60)/4 y4 55 = (40+50+60+70)/4 y5 65 = (50+60+70+80)/4 y6 75 = (60+70+80+90)/4 y7 85 = (70+80+90+100)/4

Thus, the moving average sequence has n-k+1=10-4+1=7 values.

In [11]:
```
import numpy as np
def moving_average_sum(sequence, window):
    cumulativesum, moving_aves = [0], []
    for i, x in enumerate(sequence, 1):
        cumulativesum.append(cumulativesum[i-1] + x)
        if ( i >= window ):
            moving_ave = (cumulativesum[i] - cumulativesum[i- window])/window
            moving_aves.append(moving_ave)
    return moving_aves

print(moving_average_sum([10, 20, 30, 40, 50, 60, 70,80,90,100], 4))
print(moving_average_sum([3, 5, 7, 2, 8,10,11,65,72,81,99,100,150], 3))
```

```
[25.0, 35.0, 45.0, 55.0, 65.0, 75.0, 85.0]
[5.0, 4.666666666666667, 5.666666666666667, 6.666666666666667, 9.66666666666666
6, 28.666666666666668, 49.333333333333336, 72.66666666666667, 84.0, 93.33333333
333333, 116.33333333333333]
```

Write a function to find moving average in an array over a window: Test it over [3, 5, 7, 2, 8, 10, 11, 65, 72, 81, 99, 100, 150] and window of 3.

In [12]:
```
def mov_avg(a, b) :
    res = np.cumsum(a, dtype=float)
    res[b:] = res[b:] - res[:-b]
    return res[b - 1:] / b
```

In [13]:
```
x = [3, 5, 7, 2, 8, 10, 11, 65, 72, 81, 99, 100, 150]
moving_average(x,3)
```

Out[13]:
```
array([  5.        ,   4.66666667,   5.66666667,   6.66666667,
         9.66666667,  28.66666667,  49.33333333,  72.66666667,
        84.        ,  93.33333333, 116.33333333])
```

In [ ]:
```

```