In [2]:
```
# 2. Problem Statement
# 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
# ===== == == == == == == == == == ==
# 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.

# Problem Statement
# 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.

import numpy as np

def moving_average(a, n=3) :
# we use np.cumsum to get a cumulative sum of the numbers
    asum = np.cumsum(a, dtype=float)
#We are breaking the array as per desired number which is 'n' in our function
#and subtracting elements to get the desired result
    asum[n:] = asum[n:] - asum[:-n]
    return asum[n - 1:] / n

a=[3, 5, 7, 2, 8, 10, 11, 65, 72, 81, 99, 100, 150]
a1=np.array(a)
#rounding off the results to two digts
np.array(moving_average(a1,n=3)).round(2)
```

Out[2]:
```
array([  5.  ,   4.67,   5.67,   6.67,   9.67,  28.67,  49.33,  72.67,
        84.  ,  93.33, 116.33])
```