Assignment

Problem Statement:

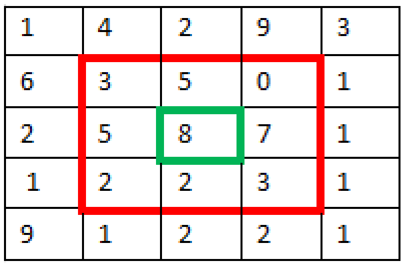
Create a matrix of size **1024 * 1024** filled with all random **INTEGER** values. Write a **C++ code using multithreading** to calculate the average of the values in the neighbourhood of N (radius) and assign it to the corresponding position in result matrix.
Note: N can take values from 1 to 5.

E.g.: Consider a 5x 5 Matrix **P(x, y)** as shown below. In this example we are going to calculate average around position (2,2) in P matrix and write the result in Q matrix at location (2,2). Now to compute an average at position **P(2, 2)** with radius **N=1**, you need to consider its neighbourhood as shown below and put the result after **rounding up** in an output matrix **Q(2, 2)**.

Q(2, 2) = CEIL((P(1,1) + P(1,2) + P(1,3) + P(2,1) + P(2,2) + P(2,3) + P(3,1) + P(3,2) + P(3,3)) / 9)

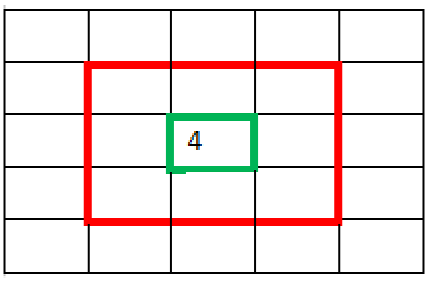
Q(2, 2) = CEIL ((3+5+0+5+8+7+2+2+3) /9) = 4

p(x,y) =



1	4	2	9	3
6	3	5	0	1
2	5	8	7	1
1	2	2	3	1
9	1	2	2	1

Q(x, y) =



		4		

Similarly, all the other elements of Q(x, y) are to be computed.

For a cases like P(0,0), P(0,1), P(1,0), P(4,4) etc where there are no neighbouring elemnts on some sides, the unavilable elements are assumed to be zero.

Therefore, Q(0,0) = CEIL((0+0+0+0+1+4+0+6+3)/9) = CEIL(1.5) = 2

Use 4 threads to compute Q(x,y).

Expected Solution:

Your solution should contain:

1) Approach/Pseudo code.
2) Generic C++ code which works for diffrent values of N ranging from 1 to 5.
3) Performance comparison when running single threaded application vs mutithreaded application.
4) Code should be compliable.