# Assignment 2: Block Reduction

Let us consider a 1D Neighborhood Operation say **reduce()** which is a spatial transformation that takes as input a 1D array **A** of size **N** and produces a 1D output array **B** of size **M** where M<N. This is obtained by considering a 1D window **W** of size **K** and sliding it along A in strides of **K.** The total number of overlaps between A and W is equal to **N/K**. The operation in context takes the average of the elements for each such overlap and produces one element of array **B.** The total number of elements for B is therefore M=N/K. The reduce operation is called repeatedly until the number of elements becomes less than K. For example:

Consider A=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16] where N = 16 and K=4. The first output array B is therefore of size 16/4=4 and is [2.5 6.5 10.5 14.5 ]. The first entry of B is B[0] = avg (A[0], A[1],A[2],A[3]). The reduce kernel is called again on B to produce [8.5]. Note the output for this repeated operation can be a 1D array or a single element depending on the values of N and K.

## Input Specifications

---

T
P,q
2^p array elements
where T is the number of test cases, p=log N and q =log K (N and K are powers of 2)

## Output:

---

The final reduced array

## Launch Specifications

---

For each invocation of the reduce kernel, the launch parameters should be

<<<(sqrt(N/K),sqrt(N/K),1),(K,1,1)>>> i.e launch a grid of 2D Blocks where each block has k threads and each block of threads uses the partial reduction code