3460:677 HOMEWORK 2

1. Consider the matrix addition problem from the previous homework:

   > A matrix addition takes two input matrices **B** and **C** and produces one output matrix **A**. Each element of the output matrix **A** is the sum of the corresponding elements of the input matrices **B** and **C**, that is, **A[i][j] = B[i][j] + C[i][j]**. For simplicity, we will only handle square matrices of which the elements are single-precision floating-point numbers.

   In the example above, can one use shared memory to reduce the global memory bandwidth consumption? (Hint: analyze the elements accessed by each thread and see if there is any commonality between threads.)

2. What type of incorrect execution behavior can happen if one forgets to use `syncthreads()` (**boldfaced** below) in this kernel?

```
// Matrix multiplication kernel – per thread code
__global__ void MatrixMulKernel (float* Md, float* Nd, float* Pd, int Width) {

    // allocate tiles in __shared__ memory
    __shared__ float s_m[TILE_WIDTH][TILE_WIDTH];
    __shared__ float s_n[TILE_WIDTH][TILE_WIDTH];

    // Calculate the row index of the Pd element and M
    int Row = blockIdx.y * TILE_WIDTH + threadIdx.y;
    // Calculate the column idenx of Pd and N
    int Col = blockIdx.x * TILE_WIDTH + threadIdx.x;

    float Pvalue = 0;

    // loop over the tiles of the input in phases
    for (int p = 0; p < width / TILE_WIDTH; p++) {

       // collaboratively load tiles into shared memory
       s_m[threadIdx.y][threadIdx.x] = Md[Row * Width + (p * TILE_WIDTH + threadIdx.x)];
       s_n[threadIdx.y][threadIdx.x] = Nd[(p * TILE_WIDTH + threadIdx.y) * Width + Col];
       __syncthreads();

       // dot-product between row of s_m and col of s_n
       for (int k = 0; k < TILE_WIDTH; ++k)
         Pvalue += s_m[threadIdx.y][k] * s_n[k][threadIdx.x];
       __syncthreads();
    }
    Pd[Row * Width + Col] = Pvalue;
}
```

3. Assuming capacity was not an issue for registers or shared memory, give one case that it would be valuable to use shared memory instead of registers to hold values fetched from global memory? Explain your answer.

Last updated 1.24.2014 by T. O'Neil.