Quiz Questions for Module 3

1. We are to process a 600X800 (800 pixels in the x or horizontal direction, 600 pixels in the y or vertical direction) picture with the PictureKernel(). That is m's value is 600 and n's value is 800.

   ```
   __global__ void PictureKernel(float* d_Pin, float* d_Pout, int n, int m) {
       // Calculate the row # of the d_Pin and d_Pout element to process
       int Row = blockIdx.y*blockDim.y + threadIdx.y;
       // Calculate the column # of the d_Pin and d_Pout element to process
       int Col = blockIdx.x*blockDim.x + threadIdx.x;
       // each thread computes one element of d_Pout if in range
       if ((Row < m) && (Col < n)) {
           d_Pout[Row*n+Col] = 2*d_Pin[Row*n+Col];
       }
   }
   ```

   Assume that we decided to use a grid of 16X16 blocks. That is, each block is organized as a 2D 16X16 array of threads. How many warps will be generated during the execution of the kernel?

   (A) 37*16
   (B) 38*50
   (C) 38*8*50
   (D) 38*50*2

   Answer: (C)
   Explanation: There are ceil(800/16.0) = 50 blocks in the x direction and ceil(600/16.0) = 38 blocks in the y direction. Each block contributes (16*16)/32 = 8 warps. So there are 38*50*8 warps.

2. In Question 1, how many warps will have control divergence?

   a. (A) 37 + 50*8
   b. 38*16
   c. 50
   d. 0

   Answer: (D)
   Explanation: The size of the picture in the x dimension is a multiple of 16 so there is no block in the x direction that has any threads in the invalid range. The size of the picture in the y dimension is 37.5 times of 16. This means that the threads in the last block are divided into halves: 128 in the valid range and 128 in the invalid range. Since 128 is a multiple of 32, all warps will fall into either one or the other range. There is no control divergence.

3. In Question 1, if we are to process an 800x600 picture (600 pixels in the x or horizontal direction and 800 pixels in the y or vertical direction) picture, how many warps will have control divergence?

   a. 37+50*8
   b. 38*16
   c. 50*8
   d. 0

   Answer: (C)
   Explanation: The size of the picture in the x dimension is 600, which is 37.5 times of 16. This means that every warp processing the right edge of the picture will have control divergence. There are 50*8 such warps (50 blocks, 8 warps in each block). Since the size of the picture in the y dimension is a multiple of 16, there is no more divergence in the warps that process the lower edge of the picture.

4. In Question 1, if are to process a 799x600 picture (600 pixels in the x direction and 799 pixels in the y direction), how many warps will have control divergence?

   a. 37+50*8
   b. (37+50)*8
   c. 50*8
   d. 0

   Answer: (A)
   Explanation: The number of warps processing the right edge remains 50*8, all of which will have control divergence. However, the warps processing the lower edge of the picture will also have control divergence. There are 38 of them. One of them is already counted for processing the right edge. So we have 50*8+38-1 = 50*8+37.

5. Assume the following simple matrix multiplication kernel

```
__global__ void MatrixMulKernel(float* M, float* N, float* P, int Width)
{
    int Row = blockIdx.y*blockDim.y+threadIdx.y;
    int Col = blockIdx.x*blockDim.x+threadIdx.x;
    if ((Row < Width) && (Col < Width)) {
        float Pvalue = 0;
        for (int k = 0; k < Width; ++k) {Pvalue += M[Row*Width+k] * N[k*Width+Col];}
        P[Row*Width+Col] = Pvalue;
    }
}
```

If we launch the kernel with a block size of 16X16 on a 1000X1000 matrix, how many warps will have control divergence?

a. 1,000
b. 500
c. 1,008
d. 508

Answer: (B)
Explanation: There will be 63 blocks in the horizontal direction. 8 threads in the x dimension in each row will be in the invalid range. Every two rows form a warp. Therefore, there are 1000/2 =500 warps that will straddle the valid and invalid ranges in the horizontal direction. As for the warps in the bottom blocks, there are 8 warps in the valid range and 8 warps in the invalid range. Threads in these warps are either totally in the valid range or invalid range.

6. If a CUDA device's SM (streaming multiprocessor) can take up to 1,536 threads and up to 8 thread blocks. Which of the following block configuration would result in the most number of threads in each SM?

a. 64 threads per block
b. 128 threads per block
c. 512 threads per block
d. 1,024 threads per block

Answer: (C)
Explanation: (A) and (B) are limited by the number of thread blocks that can be accommodated by each SM. (D) is not a divider of 1,536, leaving 1/3 of the thread space open. (C) results in 3 blocks and fully occupies the capacity of 1,536 threads in each SM.