Assignment 7

The due date for submitting this assignment has passed. **Due on 2020-03-18, 23:59 IST.**
As per our records you have not submitted this assignment.

1) For the following code snippet for reduction, find the total number of memory write accesses between line number 6 and 10 per block. The blockDim.x is 64 and warp size is 16.

**Kernel**

1. unsigned int tid = threadIdx.x;
2. unsigned int i = blockIdx.x*blockDim.x + threadIdx.x;
3. __shared__ float sdata[64];
4. sdata [ tid ] = (i < n) ? g_idata [i] : 0;
5. __syncthreads ();
6. for (unsigned int s=blockDim.x/2; s>0; s>>=1)
7. { if (tid < s)
8. sdata [ tid ] += sdata [tid + s];
9. __syncthreads ();
10.}

**Number of write accesses**

No, the answer is incorrect. 
Score: 0
2) Find the total number of memory read accesses between line number 6 and 10 per block as described in question 1.

Number of read accesses ______________

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: Numeric) 4

3) In the given code snippet for reduction using Algorithm Cascading, there are some mistakes. Identify the line number(s) for the probable error(s). Multiple choices can be possible for this question.

1. .......
2. unsigned int tid = threadIdx.x;
3. unsigned int i = blockIdx.x * 2 * (blockDim.x) + threadIdx.x;
4. unsigned int gridSize = blockSize*2*gridDim.x;
5. sdata[tid] = 0;
6. while (i < n) {
7.     sdata[tid] += g_idata[i] + g_idata[i+gridSize];
8.     i ++;
9. }
10. .......

Options:
- a) 3
- b) 4
- c) 7
- d) 8
Consider the following kernel snippet. Identify a probable cause that is constraining the performance of the kernel.

```c
__global__ void kernel(float *odata, const float *idata)
{
    __shared__ float tile[TILE_DIM][TILE_DIM];
    int x = blockIdx.x * TILE_DIM + threadIdx.x;
    int y = blockIdx.y * TILE_DIM + threadIdx.y;
    int width = blockDim.x * TILE_DIM;

    for (int j = 0; j < TILE_DIM; j += BLOCK_ROWS)
        tile[threadIdx.y+j][threadIdx.x] = idata[(y+j)*width + x];
    __syncthreads();
    x = blockIdx.y * TILE_DIM + threadIdx.x; // transpose block offset
    y = blockIdx.x * TILE_DIM + threadIdx.y;
    for (int j = 0; j < TILE_DIM; j += BLOCK_ROWS)
        odata[(y+j)*width + x] = tile[threadIdx.x][threadIdx.y+j];
}
```

Select the correct option

A. Branch Divergence
B. Uncoalesced Global Memory Loads
C. Shared memory bank conflicts
D. Uncoalesced Global Memory Stores
No, the answer is incorrect.
Score: 0
Accepted Answers:
C.

5) 20 points
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 B=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] where N = 8 and k=2. The output array B is therefore of size 16/2=8 and is [1.5 3.5 5.5 7.5]. The first entry of B is B[0] = \text{avg}(A[0], A[1])

Consider the following incomplete code-snippet.

```c
__global__ void 1DPool(int* A, int* B, int M, int N, int k)
{
    int tid = (blockIdx.y * blockDim.x + blockIdx.x) * blockDim.x + threadIdx.x;
    int bound = (a)
    int blockSize = (b)
    for(int s=1;s<blockSize;s*=2)
    {
        if(tid%2*s == 0 && tid+s<bound)
        --syncthreads();
    }
    if(threadIdx.x == 0)
        B[(c)] = A[(d)];
}
```
The kernel is launched with a 2D grid of 1D blocks where each block is responsible for reducing k consecutive elements to one element.

Choose the correct matching and pairing for the following options.

(a) i) blockDim.x
(b) ii) (1+(blockIdx.y * gridDim.x + blockIdx.x)) * blockDim.x
(c) iii) blockIdx.y * blockDim.x + blockIdx.x
(d) iv) (blockIdx.y * blockDim.x + blockIdx.x) * blockDim.x

Options:

A) a-i, b-ii, c-iii, d-iv
B) a-ii, b-i, c-iii, d-iv
C) a-iv, b-i, c-ii, d-iii
D) a-iii, b-ii, c-i, d-iv

No, the answer is incorrect.
Score: 0
Accepted Answers: B.

6)
Consider an array A where A[i]=1 if i is divisible by 4 and A[i] = 0 otherwise. The array has 1022 elements. Let us consider a reduction which computes XOR of all the elements of the array using a binary XOR operator. What is the final value of this reduction operation at the end? ________________

Options:

a) 0
b) 1
No, the answer is incorrect.
Score: 0
Accepted Answers:
a.