Quiz Questions for Module 2

1. If we want to allocate an array of v integer elements in CUDA device global memory, what would be an appropriate expression for the second argument of the cudaMalloc() call?
2. n
3. v
4. n \* sizeof(int)
5. v \* sizeof(int)

Answer: (D)

Explanation: This one should be self-evident.

1. If we want to allocate an array of n floating-point elements and have a floating-point pointer variable d\_A to point to the allocated memory, what would be an appropriate expression for the first argument of the cudaMalloc() call?
2. n
3. (void \*) d\_A
4. \*d\_A
5. (void \*\*) &d\_A

Answer: (D)

Explanation: &d\_A is pointer to a pointer of float. To convert it to a generic pointer required by cudaMalloc() should use (void \*\*) to cast it to a generic double-level pointer.

1. If we want to copy 3000 bytes of data from host array h\_A (h\_A is a pointer to element 0 of the source array) to device array d\_A (d\_A is a pointer to element 0 of the destination array), what would be an appropriate API call for this in CUDA?
2. cudaMemcpy(3000, h\_A, d\_A, cudaMemcpyHostToDevice);
3. cudaMemcpy(h\_A, d\_A, 3000, cudaMemcpyDeviceTHost);
4. cudaMemcpy(d\_A, h\_A, 3000, cudaMemcpyHostToDevice);
5. cudaMemcpy(3000, d\_A, h\_A, cudaMemcpyHostToDevice);

Answer: (C)

Explanation: See Lecture 2.2 slides.

1. How would one declare a variable err that can appropriately receive returned value of a CUDA API call?
2. int err;
3. cudaError err;
4. cudaError\_t err;
5. cudaSuccess\_t err;

Answer: (C)

Explanation: See Lecture 2.2 slides.

1. If we want to allocate using Unified Memory an array of n elements of type double and have a pointer variable d\_A to point to the allocated memory, what would be the appropriate API call?
2. cudaMalloc((void \*\*)& d\_A, n \* sizeof(double) );
3. cudaMallocManaged((void \*\*)& d\_A, n \* sizeof(double) );
4. cudaMallocHost((void \*\*)& d\_A, n \* sizeof(double) );
5. cudaMallocManaged( d\_A, n);

Answer: (B)

Explanation: See Lecture 2.6 slides.