ECE 6360 Parallel and Heterogeneous Computing

Homework 6: CUDA Language and API

Label each item as part of the CUDA API or the CUDA language:

`cudaMalloc` `__global__` `threadIdx` `cudaMemcpy` `__device__` `cudaGetDeviceCount`

When is PTX code compiled into binary?

The following program crashes and the driver resets. What is the problem?

```
__global__ void mmult(float* A, float* B, float* C, int N) {
  int row = blockIdx.y * blockDim.y + threadIdx.y;
  int col = blockIdx.x * blockDim.x + threadIdx.x;


  float sum = 0.0f;
  for(int k = 0; k < N; k++) {
    sum += A[row * N + k] * B[k * N + col];
  }
}
```

Once you fix the crash, the code runs but doesn't produce a result. Why?

The following code contains at least 6 errors. Note each error at the appropriate line:

```
float host_log(float v) { return log(v); }

__global__ int scale(float* x, int n, float a) {
  int i = blockIdx.x * blockDim.x + threadIdx.x;
  if(i < n) x[i] = host_log(a) * x[i];
  return 0;
}
int main() {
  int N = 30000;
  float* x = new float[N];
  // fill x
  float* gpu_x;
  size_t bytes = N;
  cudaMalloc(gpu_x, bytes);
  cudaMemcpy(x, gpu_x, bytes, cudaMemcpyHostToDevice);
  scale<<<(N + 255) / 256, 256>>>(x, N, 2.0f);
  cudaMemcpy(gpu_x, x, bytes, cudaMemcpyDeviceToHost);
}
```