(/mp/9999?show=code) Lab Tour with Device Query Attempt

Attempt Summary

Submit Attempt for Grading

Remember to answer the questions before clicking.

Dataset Id:	None
Created:	less than a minute ago (2022-01-30 18:54:52 +0000 UTC)
Status:	Correct solution for this dataset.

Timer Output

Kind	Location	Time (ms)	Message
GPU	main.cu::14	0.208434	Getting GPU Data.

Logger Output

Level	Location	Message
Trace	main::30	There is 1 device supporting CUDA
Trace	main::37	Device 0 name: GRID K520
Trace	main::39	Computational Capabilities: 3.0

Level	Location	Message
Trace	main::41	Maximum global memory size: 4294770688
Trace	main::43	Maximum constant memory size: 65536
Trace	main::45	Maximum shared memory size per block: 49152
Trace	main::48	Maximum block dimensions: 1024 x 1024 x 64
Trace	main::51	Maximum grid dimensions: 2147483647 x 65535 x 65535
Trace	main::52	Warp size: 32

Program Code

```
#include <wb.h>

//@@ The purpose of this code is to become familiar with the submission
//@@ process. Do not worry if you do not understand all the details of
//@@ the code.

int main(int argc, char **argv) {
  int deviceCount;

  wbArg_read(argc, argv);

  cudaGetDeviceCount(&deviceCount);

  wbTime_start(GPU, "Getting GPU Data."); //@@ start a timer

  for (int dev = 0; dev < deviceCount; dev++) {
    cudaDeviceProp deviceProp;

    cudaGetDeviceProperties(&deviceProp, dev);

    if (dev == 0) {
      if (deviceProp.major == 9999 && deviceProp.minor == 9999) {
        wbLog(TRACE, "No CUDA GPU has been detected");
        return -1;
      } else if (deviceCount == 1) {
        //@@ WbLog is a provided logging API (similar to Log4J).
        //@@ The logging function wbLog takes a level which is either
```

```
28          //@@ OFF, FATAL, ERROR, WARN, INFO, DEBUG, or TRACE and a
29          //@@ message to be printed.
30          wbLog(TRACE, "There is 1 device supporting CUDA");
31        } else {
32          wbLog(TRACE, "There are ", deviceCount,
33                " devices supporting CUDA");
34        }
35      }
36
37      wbLog(TRACE, "Device ", dev, " name: ", deviceProp.name);
38      wbLog(TRACE, " Computational Capabilities: ", deviceProp.major, ".",
39            deviceProp.minor);
40      wbLog(TRACE, " Maximum global memory size: ",
41            deviceProp.totalGlobalMem);
42      wbLog(TRACE, " Maximum constant memory size: ",
43            deviceProp.totalConstMem);
44      wbLog(TRACE, " Maximum shared memory size per block: ",
45            deviceProp.sharedMemPerBlock);
46      wbLog(TRACE, " Maximum block dimensions: ",
47            deviceProp.maxThreadsDim[0], " x ", deviceProp.maxThreadsDim[1],
48            " x ", deviceProp.maxThreadsDim[2]);
49      wbLog(TRACE, " Maximum grid dimensions: ", deviceProp.maxGridSize[0],
50            " x ", deviceProp.maxGridSize[1], " x ",
51            deviceProp.maxGridSize[2]);
52      wbLog(TRACE, " Warp size: ", deviceProp.warpSize);
53    }
54
55    wbTime_stop(GPU, "Getting GPU Data."); //@@ stop the timer
56
57    return 0;
58  }
59
```