Write a program to output the key components of an NVIDIA GPU for all GPUs installed in the machine. You will need the following cuda library functions.

```
int devCount;
cudaGetDeviceCount(&devCount);

cudaDeviceProp devProp;
cudaGetDeviceProperties(&devProp, 0);
```

NOTE: If devCount=3 then the devices are numbered 0, 1, 2. Feed the appropriate number for the second parameter of cudaGetDeviceProperties() to get a specific GPU properties.

This is what a cudaDeviceProp looks like. (I think I have the data type correct, let me know otherwise).

```
struct cudaDeviceProp {
   int major;                          /*****/
   int minor;                          /*****/
   char name[256];                     /*****/
   unsigned long totalGlobalMem;       /*****/
   unsigned sharedMemPerBlock);        /*****/
   int regsPerBlock;                   /*****/
   int warpSize;                       /*****/
   unsigned long memPitch;             /*****/
   int maxThreadsPerBlock;             /*****/
   unsigned maxThreadsPerMultiProcessor; /*****/
   int maxBlockPerMultiProcessor;
   int maxThreadsDim[4];               /*****/
   unsigned long maxGridSize[4];       /*****/
   int clockRate;                      /*****/
   unsigned int totalConstMem;         /*****/
   unsigned int textureAlignment;      /*****/
   bool deviceOverlap;                 /*****/
   int multiProcessorCount;            /*****/
   int concurrentKernels;
   int memoryBusWidth;
   int integrated;
   int asyncEngineCount;
   int deviceOverlap;
   int computeMode;
   boolean kernelExecTimeoutEnabled;   /*****/
   /* etc – there are more items */
}
```

The program should have an int main() and a printDevProp(const cudaDeviceProp *).
The printDevProp function only need to printout the values /*****/.

Use a .cu extension on the program. Example: deviceQuery.cu

To compile
  nvcc deviceQuery.cu -o DQ
To execute
  ./DQ

There will be additional compile options later.

Submit three files:
1) Your documented source code
2) The output from running the program on aw01
3) The output from running the program on your BOSS node in your cluster.

Here is the output with a system that has only 1 GPU.

```
CUDA Device #0
Major revision number:          8
Minor revision number:          9
Name:                           NVIDIA GeForce RTX 4070 Ti
Total global memory:            12569739264
Total shared memory per block:  49152
Total registers per block:      65536
Warp size:                      32
Maximum memory pitch:           2147483647
Maximum threads per MP:         1536
Maximum threads per block:      1024
Maximum resident blocks per MP: 32
Maximum resident warps per MP:  48
Maximum dimension 0 of block:   1024
Maximum dimension 1 of block:   1024
Maximum dimension 2 of block:   64
Maximum dimension 0 of grid:    2147483647
Maximum dimension 1 of grid:    65535
Maximum dimension 2 of grid:    65535
Clock rate:                     2610000
Total constant memory:          65536
Texture alignment:              512
Concurrent copy and execution   Yes
Number of multiprocessors:      60
ConcurrentKernels:              1
Memory bus width:               192
Integrated:                     0
AsyncEngineCount:               2
Device Overlap:                 1
Compute Mode:                   0
Kernel execution timeout:       Yes

Press any key to exit...
```