prop.cu

```
#include <cuda.h>
#include <cuda_runtime_api.h>
#include <stdio.h>
#include <stdlib.h>

int main()
{
    cudaDeviceProp prop;

    int count;
    cudaGetDeviceCount(&count);

    for (int i=0; i< count; i++) {
        cudaGetDeviceProperties(&prop, i);
        printf("   --- General Information for device %d ---\n", i);
        printf("Name:  %s\n", prop.name);
        printf("Compute capability:  %d.%d\n", prop.major, prop.minor);
        printf("Clock rate:  %d\n", prop.clockRate);

        printf("Device copy overlap:  ");
        printf(prop.deviceOverlap?"Enabled\n":"Disabled\n");
        printf("Kernel execution timeout :  " );
        printf(prop.kernelExecTimeoutEnabled?"Enabled\n":"Disabled\n");
        printf("\n" );

        printf("   --- Memory Information for device %d ---\n", i);
        printf("Total global mem:  %ld\n", prop.totalGlobalMem);
        printf("Total constant Mem:  %ld\n", prop.totalConstMem);
        printf("Max mem pitch:  %ld\n", prop.memPitch);
        printf("Texture Alignment:  %ld\n", prop.textureAlignment);
        printf("\n");

        printf("   --- MP Information for device %d ---\n", i);
        printf("Multiprocessor count:  %d\n", prop.multiProcessorCount);
        printf("Shared mem per mp:  %ld\n", prop.sharedMemPerBlock);
        printf("Registers per mp:  %d\n", prop.regsPerBlock);
        printf("Threads in warp:  %d\n", prop.warpSize);
        printf("Max threads per block:  %d\n", prop.maxThreadsPerBlock);
        printf("Max thread dimensions:  (%d, %d, %d)\n", prop.maxThreadsDim[0],
                                        prop.maxThreadsDim[1], prop.maxThreadsDim[2]
);
        printf("Max grid dimensions:  (%d, %d, %d)\n", prop.maxGridSize[0],
                                        prop.maxGridSize[1], prop.maxGridSize[2]);
        printf("\n");
    }

    return 0;
}
```