GPU matrix-vector multiplication

Note: You can not use dynamically allocated two-dimensional arrays for matrices efficiently in CUDA,- please use one-dimensional arrays from now on.

Exercise 4:

Write a CUDA C program for matrix-vector multiplication on the GPU - you can use the code from last week as a starting point.

1. Write a 'naive' version, where each thread computes one element of the output vector.
2. Compare the kernel runtime to your fastest OpenMP version. Did you get any speed-up? Is it as you would expect?
3. Compare with the `DGEMV` function for GPUs provided by Nvidia's CUBLAS library.
4. Make a version that can run simultaneously on two GPUs by splitting the task equally between them (you may assume that the matrix size is an even number).