GPU Programming
Hands-on Lab 3 worksheet

Name: Luke Luke Goodall **Marks: 10**

Block size x * y (Insert your values)	Kernel execution time (ms)	Achieved occupancy (%)	Global load throughput (GB/s)	Global memory load efficiency (%)	Global store throughput (GB/s)	Global store efficiency (%)
x * y (orig 2D grid best)	2.071635	0.844	66.044	100	33.022	100
x * y (orig 2D grid worst)	2.275763	0.624	74.623	80	43.461	68.7
x * y (1D grid best)	2.084732	0.818	65.92	100	32.96	100
x * y (1D grid worst)	2.534342	0.606	71.31	80	41.558	68.6
x * y (16 elem best)	12.528832	0.721	97.887	12.5	49.042	12.5
x * y (16 elem worst)	15.760435	0.955	66.508	12.5	33.321	12.5

Discussion of task 5:
For the 3 best kernel execution times:

- they had higher achieved occupancy, except for 16 elems
- they had lower global load throughput, except for 16 elems
- they had higher global memory load efficiency, except 16 elems had the same

- they had lower global store throughput, except 16 elems
- they had higher global store efficiency, except 16 elems had the same

With all the exceptions I triple-checked that I was reading 16 elems correctly, even in Lab2 it had many exceptions. Overall from the observations, I would deduce that:

- global memory load efficiency and global store efficiency are correlated with better performance. This is based on the fact that the best kernels had higher laid and store efficiency, and both 16 elems had very low load and store efficiency compared to the other kernels. This makes sense as the profiler shows us most of these programs' time is spent in managing memory, and much less in computation

- achieved occupancy and throughput (load and store) do not necessarily correlate to better performance as you can have a high throughput and occupancy, but it might be inefficient. In other words, they are utilising the resources to a high extent, but wastefully. So they can be indicators of good performance but not in every situation