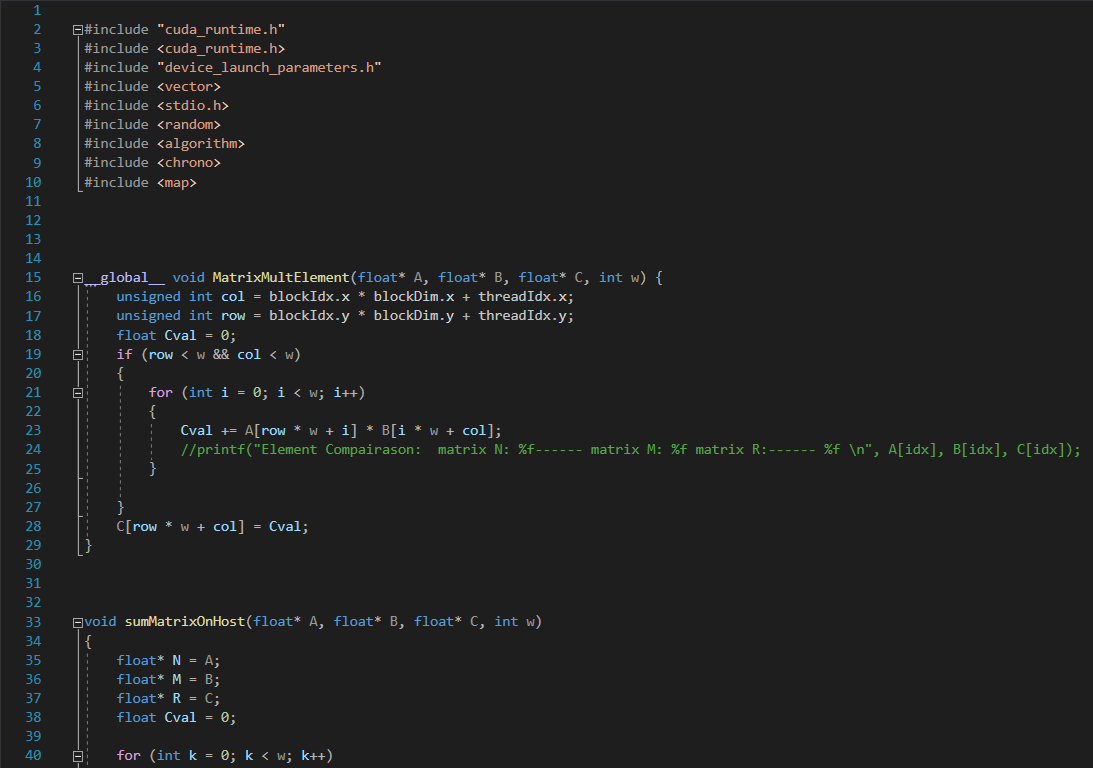
Machine Problem Assignment 3

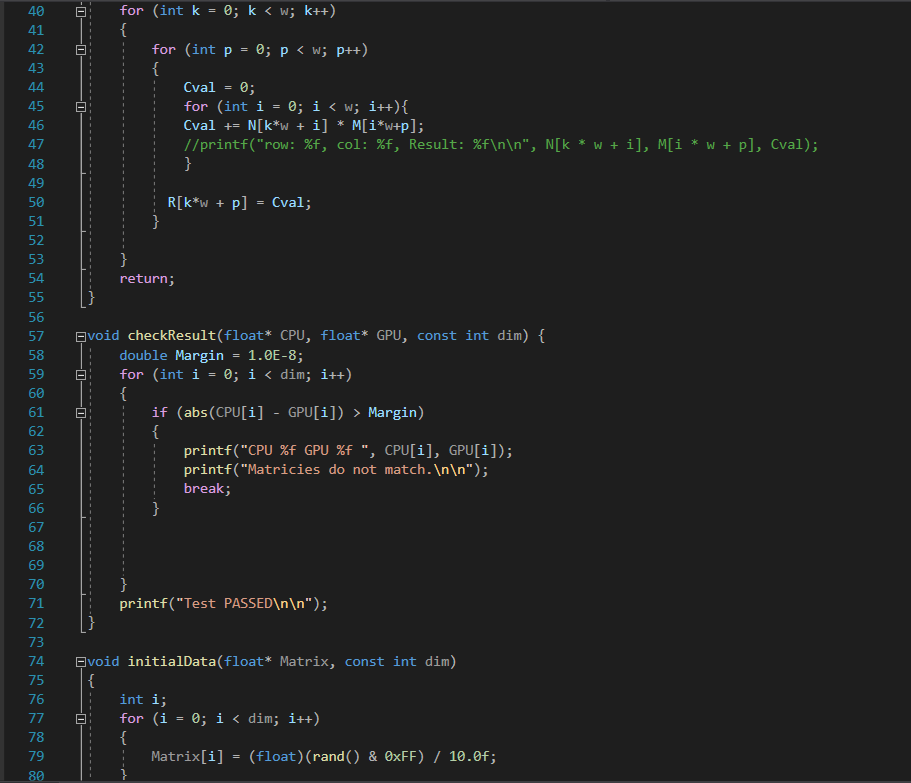
Name: Christopher Masloub

Student Number: 20052223

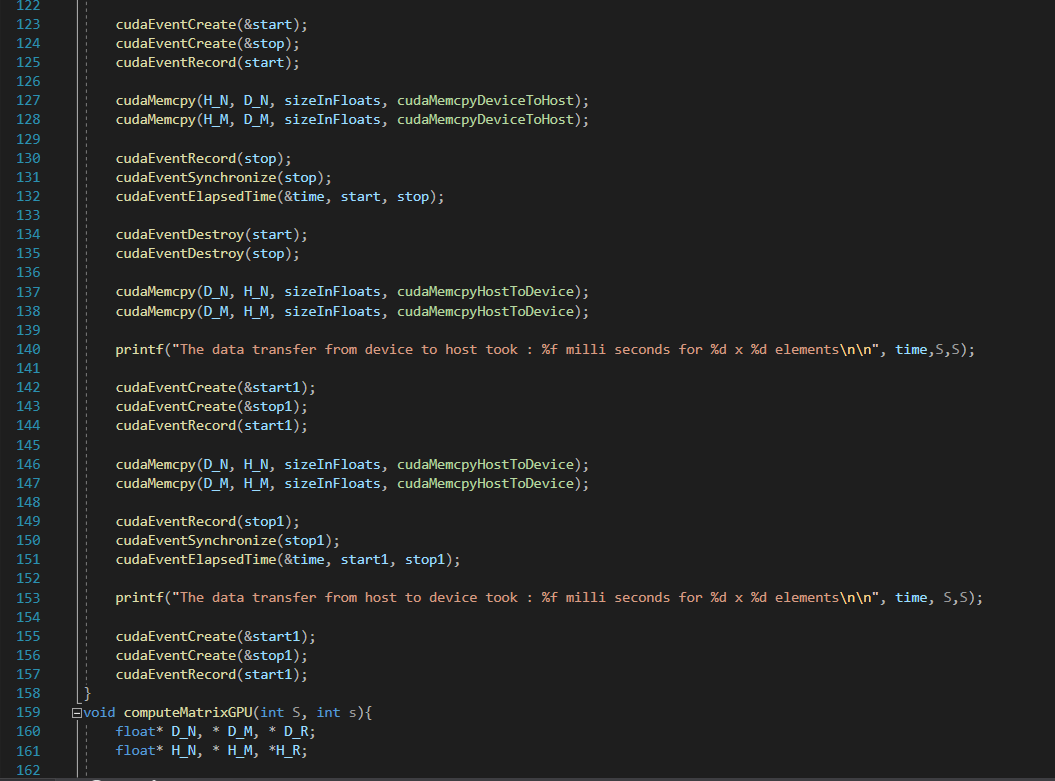
DISCLAIMER: This report was done on my own laptop using a gtx 1650 mobile due to the fact that the computation was taking too long on the remote PC especially since I had to run the CPU portion multiple times to get good median values. On the remote PC with my slow internet, the operation would often stop due to the congestion from everyone using it. My code is included in the uploaded Kernal.cu and my outputs are in a text file in the submission folder in order to make the report cleaner.

## Code:

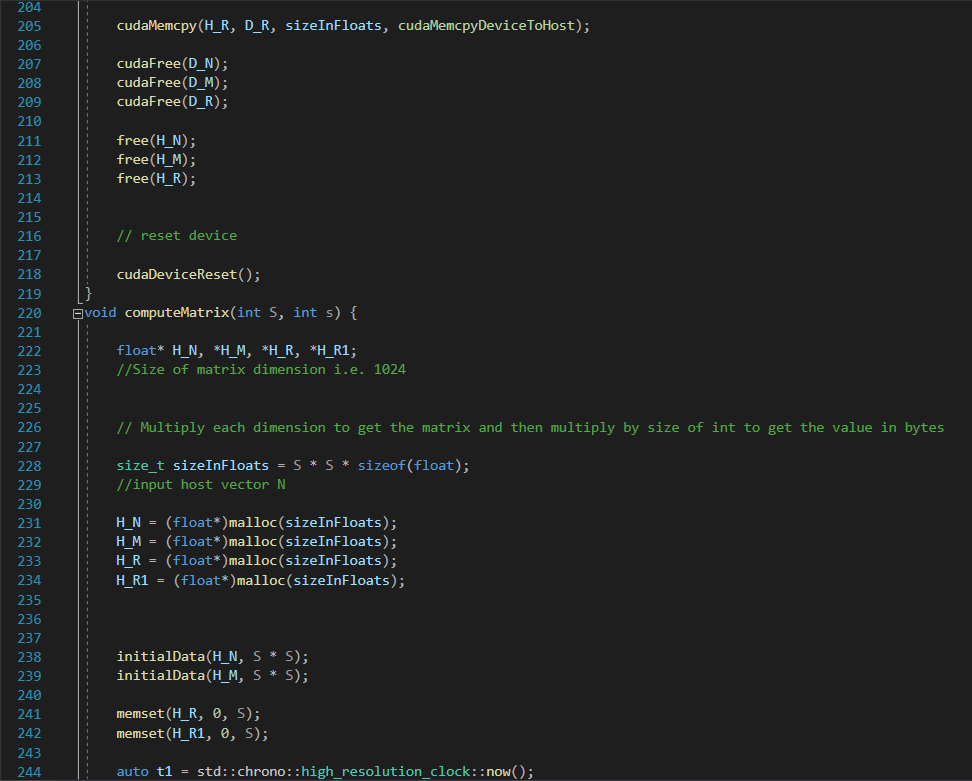


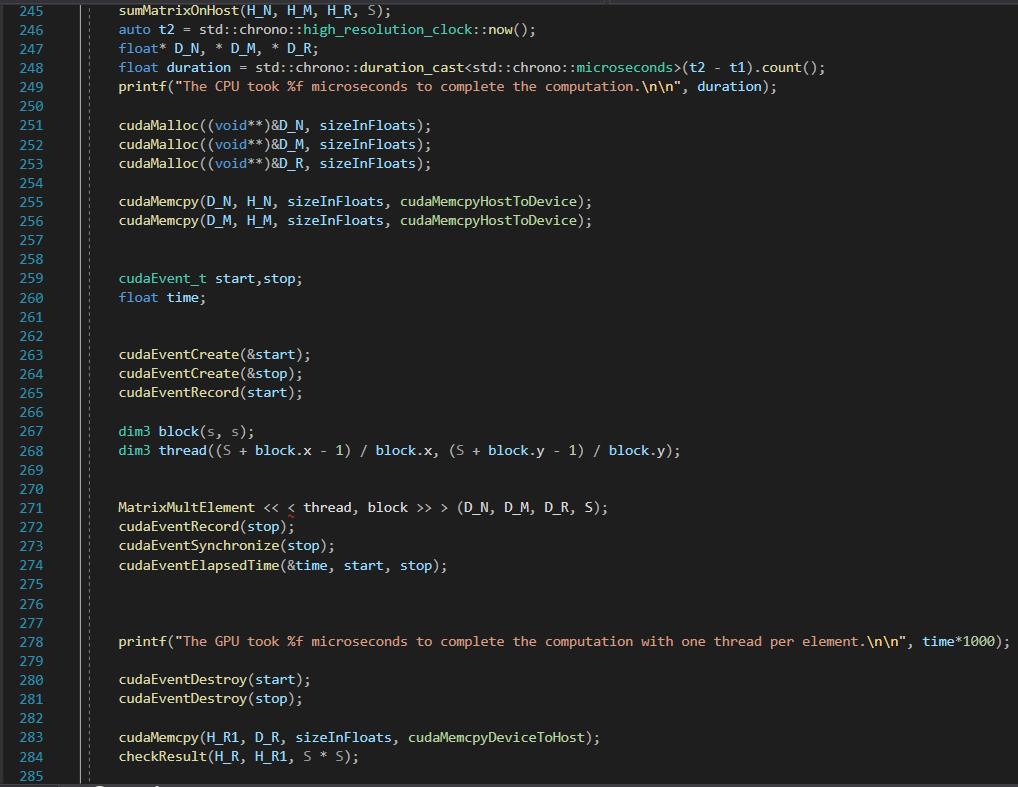


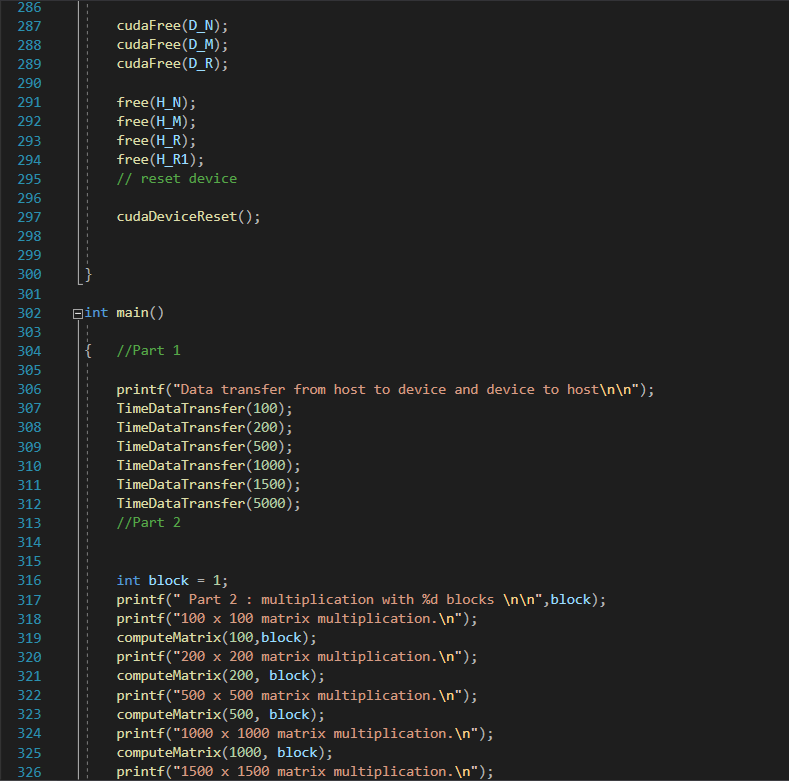


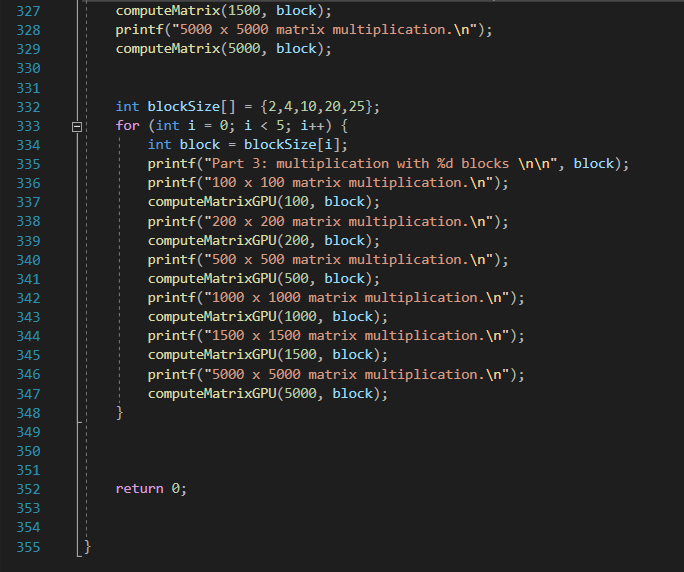




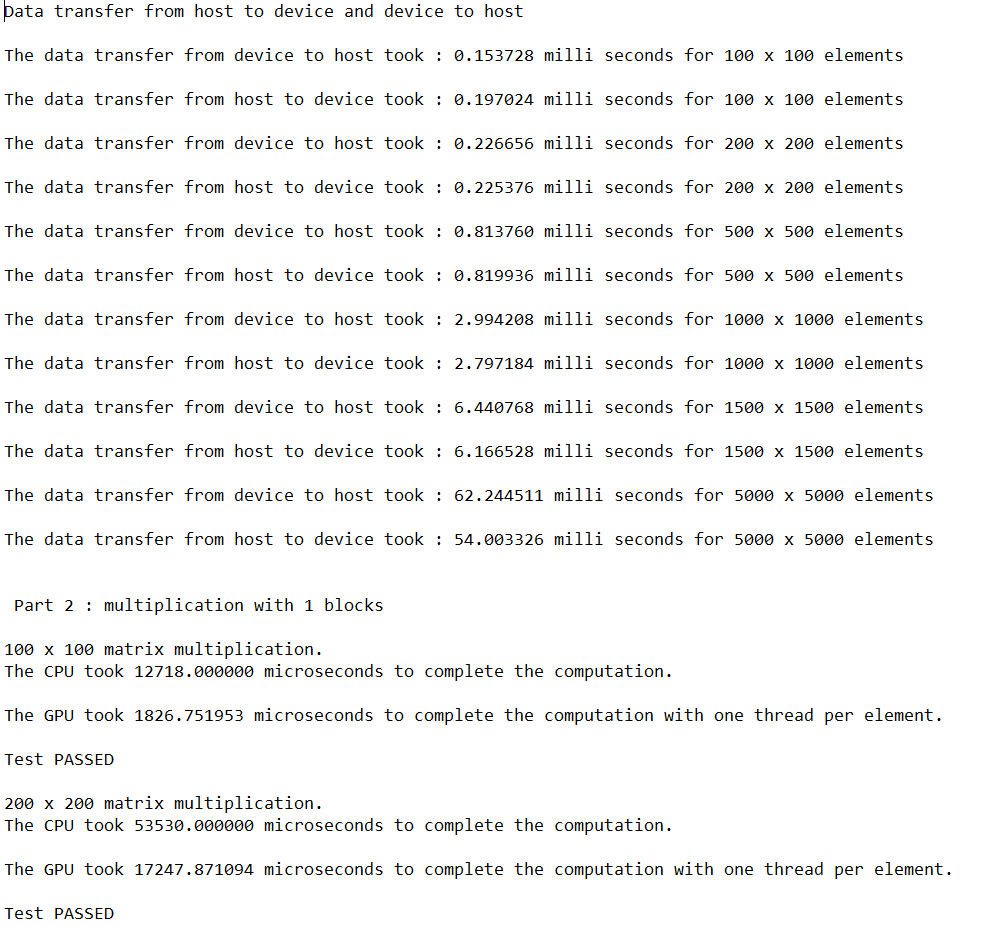


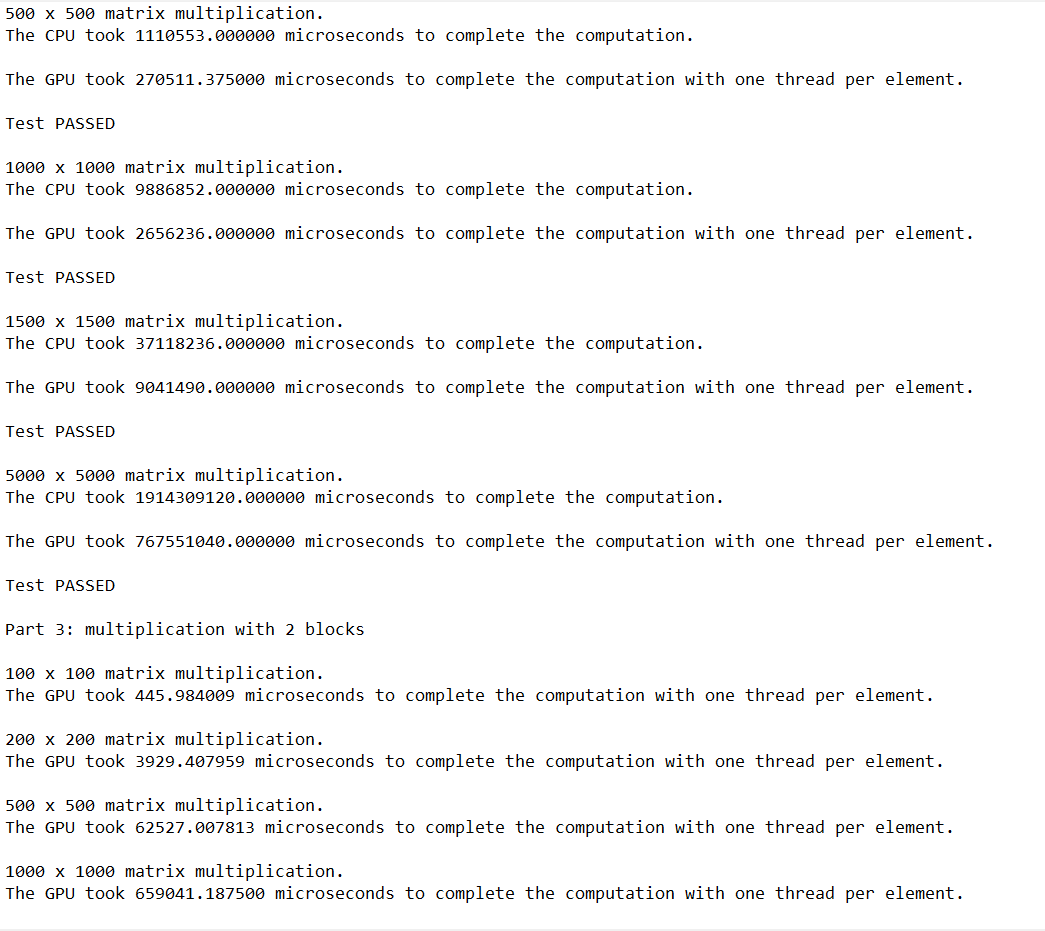


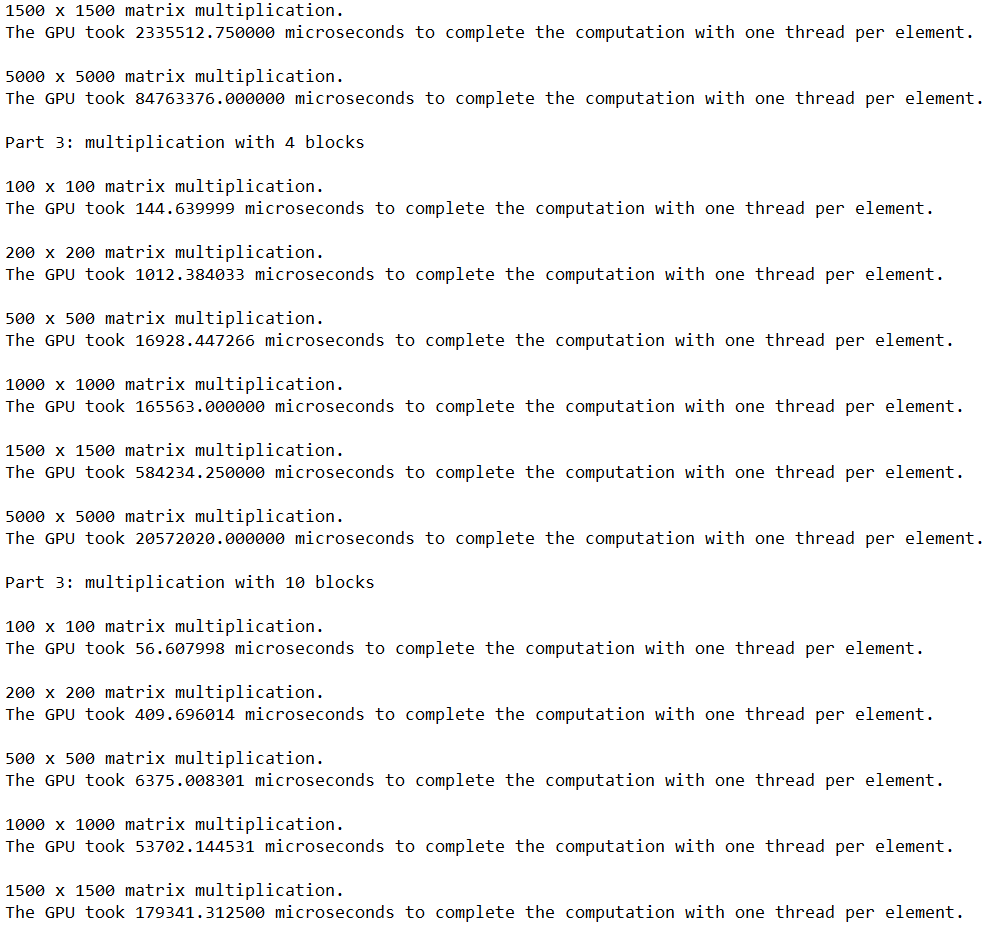


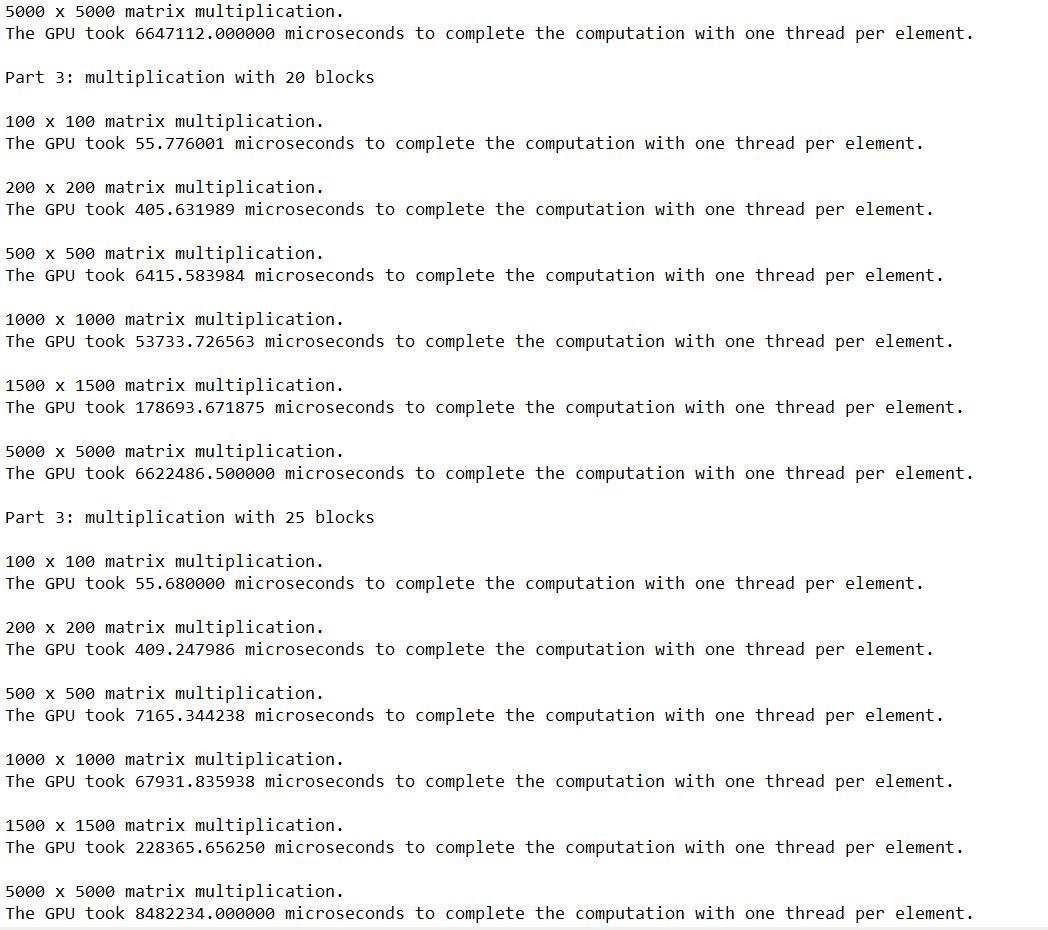


## Output:









# Part 1:

The following Chart and Table demonstrate the time in milliseconds that it takes to transfer the different sized matrices from host to device as well as device to host. Although very similar, the transfer time from host to device was on average slightly faster than device to host.

|  |  |  |
| --- | --- | --- |
|  | Transfer Time (milliseconds) | |
| Matrices Sizes | Host to Device | Device to Host |
| 100 x 100 | 0.197024 | 0.153728 |
| 200 x 200 | 0.225376 | 0.226656 |
| 500 x 500 | 0.819936 | 0.813760 |
| 1000 x 1000 | 2.797184 | 2.994208 |
| 1500 x 1500 | 6.166528 | 6.440768 |
| 5000 x 5000 | 54.003326 | 62.244511 |

# Part 2:

The following graph and table demonstrate the matrix multiplication compute times for both GPU with one element per thread and using the CPU in sequence. It is evident from the results that GPU is much exponentially faster than the CPU in almost every test. Therefore, it is conclusive that matrix multiplication heavily benefits from parallelism and the gpu should always be used to compute matrix multiplication. Even if the the transfer times from the host to the device and device to the host are factored in, their effect on computation time is negligent in comparison to the computation time difference between the GPU and the CPU making it still worthwhile to use the GPU especially for multiplication of bigger matrices.

|  |  |  |
| --- | --- | --- |
|  | Computation Time (Microseconds) | |
| Matrix Size | GPU (One Element per Thread) | CPU |
| 100x100 | 1826.751953 | 12718.000000 |
| 200x200 | 17247.871094 | 53530.000000 |
| 500x500 | 270511.375000 | 1110553.000000 |
| 1000x1000 | 2656236.000000 | 9886852.000000 |
| 1500x1500 | 9041490.000000 | 37118236.000000 |
| 5000x5000 | 767551040.000000 | 1914309120.000000 |

# Part 3:

The following Graph and Table demonstrates the computation time of matrix multiplication using different block sizes. From the findings, it is evident that generally increasing block size equates to faster processing times with diminishing returns. This is evident from the computing time difference between using 2 vs 4 blocks which is very big compared to 4 vs 10. From the table another phenomenon can be observed. When the block size is increased beyond 16, the computation time starts to increase. This is likely due to the increased number of localized memory swaps that bigger blocks need to perform. Due to the small size of local memory available with a block, increasing the block size for the computation leads to a larger percentage of the input matrix being stored inside local memory. Furthermore, beyond a certain point, when the block size is increased, more memory swaps are needed than for smaller blocks.

Part a)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Block Width 1 | BW 2 | BW 4 | BW 10 | BW 20 | BW 25 |
| Times Loaded | 2Nx2N | NxN | (N/2)x(N/2) | (N/10)x(N/10) | (N/20)x(N/20) | (N/25)x(N/25) |

Part b) Memory Access

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Block Width 1 | BW 2 | BW 4 | BW 10 | BW 20 | BW 25 |
| Computations | 2Nx2N | 4Nx4N | 8Nx8N | 16Nx16N | 32Nx32N | 64Nx64N |
| Ration of memory to computation | 1:1 | 1:4x4 | 1:16x16 | 1:64x64 | 1:512x512 | 1:2048x2048 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Processing Time (milliseconds) | | | | |
| Matrix Size | Block Width 2 | Block Width 4 | Block Width 10 | Block Width 20 | Block Width 25 |
| 100x100 | 445.984009 | 144.639999 | 56.607998 | 55.776001 | 55.68 |
| 200x200 | 3929.407959 | 1012.384033 | 409.696014 | 405.631989 | 409.247986 |
| 500x500 | 62527.007813 | 16928.447266 | 6375.008301 | 6415.583984 | 7165.344238 |
| 1000x1000 | 659041.1875 | 165563 | 53702.144531 | 53733.726563 | 67931.835938 |
| 1500x1500 | 2335512.75 | 584234.25 | 179341.3125 | 178693.671875 | 228365.65625 |
| 5000x5000 | 84763376 | 20572020 | 6647112 | 6622486.5 | 8482234 |