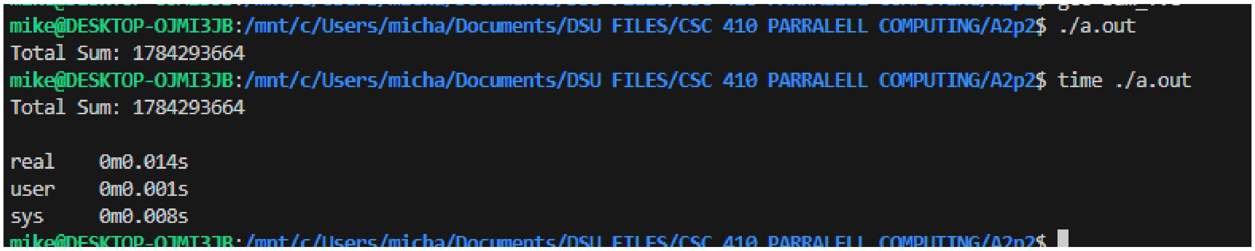
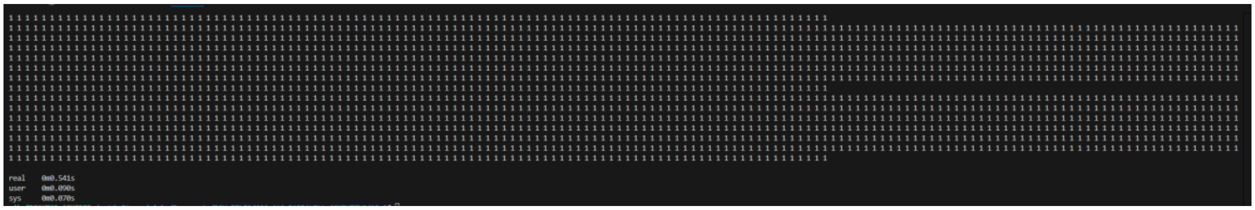
Task 1:



```
mike@DESKTOP-OJMI3JB:/mnt/c/Users/micha/Documents/DSU FILES/CSC 410 PARRALELL COMPUTING/A2p2$ ./a.out
Total Sum: 1784293664
mike@DESKTOP-OJMI3JB:/mnt/c/Users/micha/Documents/DSU FILES/CSC 410 PARRALELL COMPUTING/A2p2$ time ./a.out
Total Sum: 1784293664

real    0m0.014s
user    0m0.001s
sys     0m0.008s
mike@DESKTOP-OJMI3JB:/mnt/c/Users/micha/Documents/DSU FILES/CSC 410 PARRALELL COMPUTING/A2p2$
```

Task 2:



```
real    0m0.541s
user    0m0.090s
sys     0m0.070s
```

Task 3:

Array Sum					
	Seq.	1 Thread	2 Thread	3 Thread	4 Thread
Time	0.13	0.13s	0.19s	.013s	0.014s

Matrix Multiply					
	Seq.	1 Thread	2 Thread	3 Thread	4 Thread
Time	0.19	0.19s	0.18s	.017s	0.017s

Task 4:

The execution time did not vary much based on how many cores were used, this could be because of the time to create and join the cores was not offsetting the gain. I upped the array size to 100000000 and there were improvements in the multiple threads vs single and sequential which should be equal.