

## Task 1:

```
Matrices allocated successfully.
Matrices initialized successfully.
Matrix multiplication complete!
Resulting Matrix C:
10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10
```

Open MP conversion is working on 10x10 matrix of 1s.

When switching to the 1000 I was getting errors initializing the matrices so I switched to dynamically allocated memory:

```
$ time ./a.exe
Matrices allocated successfully.
Matrices initialized successfully.
Matrix multiplication complete!
Resulting Matrix C:

real    0m1.268s
user    0m0.000s
sys     0m0.016s
```

Now working as expected, I turned off the printMatrix function due to output size.

Task 1						
	Sequential	1 thread	2 thread	5 thread	7 thread	9 thread
Time						
pThread		5.40	2.66	1.25	.987	.954
Open MP		5.89	2.73	1.268	1.023	1.049

For this task, the static and guided open MP loops took about the same but the dynamic was around 10-15% faster. Also as expected, the pThread implementation was faster than the OpenMP static but about the same as the dynamic speed, if not slightly slower.

## Task 2:

```
micha@DESKTOP-OJMI3JB MINGW64 ~/Documents/DSU FILES/CSC 410 PARRALELL COMPUTING/A4 (main)
$ time ./a.exe
Estimated value of  $\pi$ : 3.141593

real    0m22.154s
user    0m0.000s
sys     0m0.015s
```

Estimate matches pi. The time was so high because I was using dynamic scheduling, switching to static saved 10x and was running at 2.9 s vs 22.1 s for 1 thread.

Task 2						
	Sequential	1 thread	2 thread	5 thread	7 thread	9 thread
Time		2.93	1.52	0.758	0.593	0.555