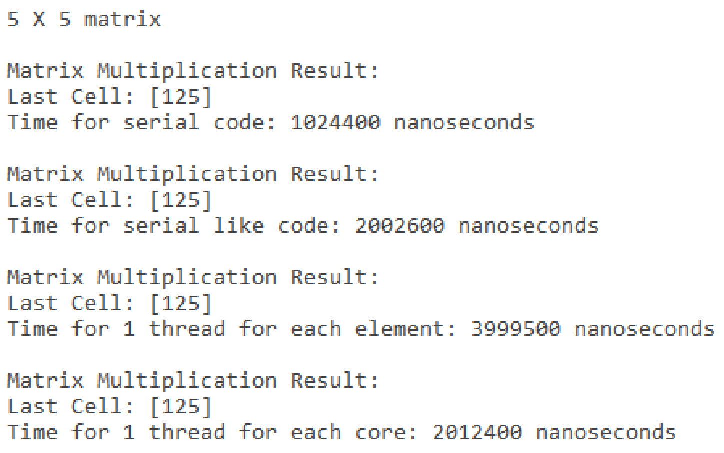
Analysis:

Note: The following are screenshots of a (5 X 5), (100 X 100), (400 X 400) and (700 X 700) matrix



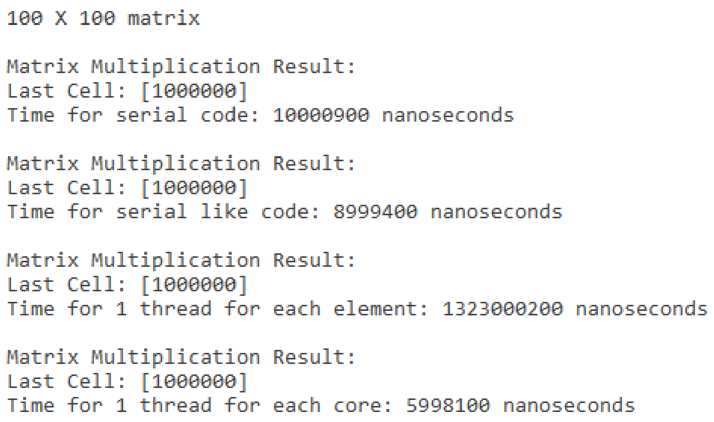
```
5 X 5 matrix

Matrix Multiplication Result:
Last Cell: [125]
Time for serial code: 1024400 nanoseconds

Matrix Multiplication Result:
Last Cell: [125]
Time for serial like code: 2002600 nanoseconds

Matrix Multiplication Result:
Last Cell: [125]
Time for 1 thread for each element: 3999500 nanoseconds

Matrix Multiplication Result:
Last Cell: [125]
Time for 1 thread for each core: 2012400 nanoseconds
```



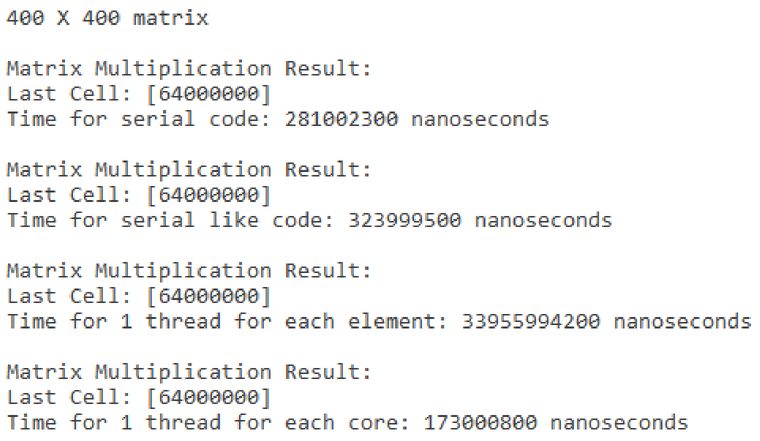
```
100 X 100 matrix

Matrix Multiplication Result:
Last Cell: [1000000]
Time for serial code: 10000900 nanoseconds

Matrix Multiplication Result:
Last Cell: [1000000]
Time for serial like code: 8999400 nanoseconds

Matrix Multiplication Result:
Last Cell: [1000000]
Time for 1 thread for each element: 1323000200 nanoseconds

Matrix Multiplication Result:
Last Cell: [1000000]
Time for 1 thread for each core: 5998100 nanoseconds
```



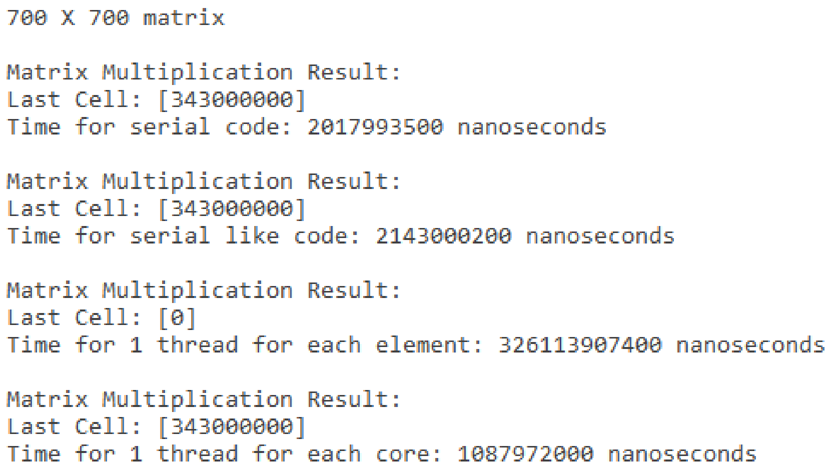
```
400 X 400 matrix

Matrix Multiplication Result:
Last Cell: [64000000]
Time for serial code: 281002300 nanoseconds

Matrix Multiplication Result:
Last Cell: [64000000]
Time for serial like code: 323999500 nanoseconds

Matrix Multiplication Result:
Last Cell: [64000000]
Time for 1 thread for each element: 33955994200 nanoseconds

Matrix Multiplication Result:
Last Cell: [64000000]
Time for 1 thread for each core: 173000800 nanoseconds
```



```
700 X 700 matrix

Matrix Multiplication Result:
Last Cell: [343000000]
Time for serial code: 2017993500 nanoseconds

Matrix Multiplication Result:
Last Cell: [343000000]
Time for serial like code: 2143000200 nanoseconds

Matrix Multiplication Result:
Last Cell: [0]
Time for 1 thread for each element: 326113907400 nanoseconds

Matrix Multiplication Result:
Last Cell: [343000000]
Time for 1 thread for each core: 1087972000 nanoseconds
```

Summary:

- 5x5
  - Slowest: 1 thread for each element program
    - 3,999,500 ns
  - Fastest: Serial program
    - 1,024,400 ns
- 100x100
  - Slowest: 1 thread for each element program
    - 1,323,000,200 ns
  - Fastest: 1 thread for each core program
    - 5,998,100 ns
- 400x400
  - Slowest: 1 thread for each element program
    - 33,955,994,200 ns
  - Fastest: 1 thread for each core program
    - 173,000,800 ns

Observations:

For my smaller matrix size (i.e 5x5) the serial programs ran faster than my threaded programs. As the matrix size increased(i.e 100x100 & 400x400) my 1 thread for each core program was consistently the fastest runtime. However, my 1 thread for each element program became consistently the slowest runtime. This is because too many threads can hurt performance. Each thread ends up fighting/ competing to complete their task first which makes the program run slower. My max matrix size was 400 X 400, this was largest size I could get to before my 1 thread for each element program crashed. As you can see from my screenshots, I was able to get a higher size (i.e 700 X 700), but it wouldn't complete the multiplication for the 1 thread for each element program.