**Lab 2 - OpenMP Data**

Hansa Chen (WC1369)

| | | N = 10,000 | |
|---|---|---|---|
| | | Time Taken | Speedup |
| Number of Threads | 1 | 0.039298 | 1 |
| | 2 | 0.025103 | 1.56547 |
| | 5 | 0.01078 | 3.64545 |
| | 10 | 0.010591 | 3.71051 |
| | 20 | 0.0100897 | 3.89486 |
| | 100 | 0.413625 | 0.09501 |

| | | N = 100,000 | |
|---|---|---|---|
| | | Time Taken | Speedup |
| Number of Threads | 1 | 2.723503 | 1 |
| | 2 | 1.359501 | 2.00331 |
| | 5 | 0.654443 | 4.16156 |
| | 10 | 0.353802 | 7.69782 |
| | 20 | 0.248639 | 10.95364 |
| | 100 | 3.087315 | 0.88216 |

For N=10,000, we saw a steady increase of speedup all the way until 20 threads, afterwards speedup decrease untill a significant decresease at 100 threads. For N= 100,000, we saw a steady incease of speedup (some almost at 11x) until 20 threads. Afterwards speedup decress and again at 100 threads, the speedup decress dramatically. Based on the data, it shows that more threads does not always equal a higher speedup. It takes resources to divde up all the numbers to each thread, have thread do calculation, then gather all thread answers together. Perhaps we see a decrease in speedup after 10 and 20 threads respectivaly are caused by having a greater peformance cost. As a result makes the overhead cost increase with thread count over the optimised.