Report

	P = 1	P = 2	P = 4	P = 8	P = 16
N = 100	804733991	451533714	250827406	52940492029	865011212
N = 1000	79757741184	40370480637	20619337458	71383002528	17628794836
N = 1e4					
N = 1e5					

The run times displayed above are in nanoseconds. The empty cells above could not be computed in a reasonable time due to the slow speed of cereberus. There are multiple interesting things to observe about this data set. First, for both N = 100 and N = 1000, if the amount of threads used is set to 8, a significant slowdown occurs. However, this slowdown does not persist through the addition of more threads in the case of N = 1000. For the smaller problem of N = 100, 4 threads seems to be the optimal amount to perform the algorithm the fastest. But as the problem size increases the increase in the amount of threads to 16 appears to give the fastest run time. Another interesting observation that I made during this assignment was that the serial version would perform better than the parallel implementation if the array size was N = 100. Only until the problem was scaled to N = 1000 was I able to achieve the appropriate speedup for the heat algorithm.