

**561 - OpenMp**  
**Cailla Rose O'Shea**  
**13305691**

To compile all code, run **make all**.

**Gaussian Elimination:**

For the Gaussian Elimination Code, I looked into the methods discussed in Numerical Recipes. I had some trouble getting any speed up for a long time, trying different omp directives. Finally I used a simple directive and used some optimization flags and got the expected speedup. Unfortunately my code would not allow for problem sizes over 1000, I feel if I could go to these higher sizes, an even greater speedup could be achieved.

**Files:**

gauss.c (serial code), threadGauss.c (openmp added), gdata.dat (results from different problem sizes), gplot.p (gnuplot script file), Gauss.ps (postscript file showing speedup).

**Sieve of Eratosthenes:**

I wrote a serial code for the sieve of eratosthenes which I believe worked pretty fast. I used getopt to allow the user to input the problem size. I then allocated memory for an array to hold the numbers up to the input value to check if prime.

I used the gettimeofday function to time the calculation.

I then used OpenMP to try to speed up the calculation. I used the basic directives: **#pragma omp parallel for** to encompass the entire calculation loops.

I experimented with problem sizes up to 100'000'000 and with using different thread amounts. As expected there was linear speedup in the code as I increased the thread size. With thread number = 8 however, the speedup was only observed as the problem size increased. The overhead of spawning threads outweighed the benefit of parallel implementation.

**Files:**

serialSieve.c (serial code), threadSieve.c (openmp added), data.dat (results), plot.p (gnuplot script), Sieve.ps (postscript with graph).