COMP 460 – Homework 2

1. Write multi-threaded Java (or Python) code using static scheduling to multiply two integer nxn matrices. Think carefully about granularity of work for each thread. You will initialize the matrices with random or a formula based integer values. Time the code for only the multiplication part (not initialization) on each run with varying values of n. You should start at n=10000 and go in increments of 2000 up until 20000. You will use cs460 via Guacacmole to use up to 8 threads. You will plot (i) number of threads vs time for the highest and the lowest value of n , and (ii) size of the matrix on the x-axis vs time on the y-axis for number of threads =1 and 8. So four total plots in all. You need to handle heap overflow errors. Make reasonable observations in terms of speedup, or runtimes or time-complexity.

2. Develop and implement multi-threaded Java (or Python) code for factoring large integers. You can use the straightforward algorithm with some simple efficiency improvements as discussed in class. You must first make a conscious choice as to which type of scheduling you will use, static or dynamic and write the reasoning for it before you implement it. Your reasoning is important in both self-assessment as well as for me to gauge your level of understanding of the material. Also plot some graphs involving run-times, threads used and size of the integer in some suitable form that portray your work in the best light. That is quite a bit of freedom, flexibility, and yes, responsibility. Reason wisely! (Yes, You will use cs460 via Guacamole to benchmark the code.)