Question 1:
After successfully implementing a procedure to determine if a provided matrix
is positive definite using cholesky decomposition, I found that the estimated
flop count of $O(n^3)$ corresponds closely to the matrix size vs time graph that
was produced by said procedure. This graph can be found under 'cholesky.png'
in this submission. This time complexity makes sense because the code has to
iterate through three for-loops of that are from roughly 0 to n, meaning that
we will have a flop count of roughly $(^1/_3)n^3 + O(n^2)$, which reduces to a time
complexity of $O(n^3)$.

Question 2:
This code uses Gaussian Elimination with partial pivoting to solve Ax = b where
$A \in R^{n \times n}$, $b \in R^n$, and $x \in R^n$. The code overwrites the entries of A that are zeroed
out with the m values calculated within the process. This saves memory but makes the
code slightly more complex. We only use partial pivoting to solve this instead of
complete pivoting, which means we shift the rows around instead of every individual
entry. This function has a time complexity of $O(n^3)$ but a flop count of
$(^2/_3)n^3 + O(n^2)$ as we calculated in lecture. The partial pivoting of this doesn't
save a ton of time, but it does help with rounding errors. This is because dividing
by a very small number will cause a lack of precision, so we make sure the code will
always divide by the absolute largest number to keep adequate precision while only
slightly increasing time complexity (i.e. only adding a n^2 factor).