Joshua Reno
CS 4235: Intro to Information Security, Fall 2018
Project 3: Crypto – Have fun with RSA

2. Computing d involves getting the modular inverse of e mod phi(n). This was done by multiplying phi(n) by a constant, incrementing by 1, and dividing the result by e while increasing the constant in a while loop.
3. If moduli share a prime factor, the corresponding public keys can be computed using GCD. This results in keys that appear strong but are actually weak. Regarding the code, we run gcd and, using p, we compute phi(n). Then, we perform the same operation in task 2, using phi(n) and e to compute d.
4. The same message is sent to three people using the same the same exponent results in a situation where someone may intercept the messages and use Chinese Remainder theorem to get the original message. I recovered the message by using Chinese remainder theorem. We can use CRT to find a value such that c1 mod N1 is equivalent to c2 mod N2 is equivalent to c3 mod N3. This value, which we will label C, is equal to m^3 mod N1*N2*N3. Since m^3 < N1*N2*N3, C = m^3. From this, we can easily find m.