Homework Number: 6
Name: Dylan Huynh
ECN login: huynh38
Due Date: 2/28/2023

In part 1, the first part of rsa is generating the prime numbers for p and q. The prime_generate function is called, which uses the PrimeGenerator class to find two primes. The function call findPrime will generate a prime number by randomly picking a number based on the number of bits PrimeGenerator was constructed with, and then will check the primality of the number based on if it prime from the Miller-Rabin algorithm. Then, it will check that p and q are not equal to 0, the two leftmost bits are set, and both numbers minus one are coprime to e, and if these are correct it will write p and q to a file. For encryption, the program will take a message block of 128, pad it to the right if it is not, and then it will pad 128 bits to the left. Then it will raise the message block to the exponent e, which was selected as 65537, and that result will be modularized by the product of p and q, which we retrieved from text files. The result of the modulo operation is then written as a 256 bit block in hexadecimal. In decryption, we break the ciphertext into 256 bit blocks, and use the chinese remainder theorem to decrypt the message. To accomplish this, we find the totient of n by multiplying p-1 and q-1, and then finding the multiplicative inverse of e modular the totient. The original message is found by breaking the message to the power of d into its residues from p and q. The reconstruction factors are found by multiplying the primes by the multiplicative inverse with respect to the other prime. The reconstruction can be made by multiplying the corresponding residues and reconstruction factors and adding them together, and modularizing by n. This returns the final message when the first 128 bits are taken out.

In part 2, encryption is done for three different sets of p and q, and in this time the selected e is 3. This is problematic because shown in this program, if three of the same plaintext messages are transmitted, the original plaintext can be recovered if e = 3. To break the RSA encryption, it takes the 3 public keys, the three encrypted messages, and uses the chinese remainder theorem to reconstruct M^3. The program does this by first calculating the Mi integers with respect to N, which is the integer made of the coprime public keys. It then uses the found Mis to find the multiplicative inverse of them with respect to their Mi's public key. These to numbers are then multiplied to the message, and the sum of all three of these messages modularized by N reconstruct M^3. The cube root of this number is then calculated, and that is the original message. Here, the individual messages can be interpreted as the residues, and thus the construction works because they were modularizes by the keys during encryption.