Zijian Gao
26319560
2/29/2016
CS311 HW3

31.6-2)

```
MODULAR-EXPONENTIATION_RL(a,b,n):
    d = 1
    let bk..b0 be the binary representation of B
    for i = 0 to k
        if(bi = 1)
            d = d * a mod n
        a = a * a mod n
    return d
```

Proof of Correctness:

b is binary, which can be represented as $b = b_k 2^k + b_{k-1} 2^{k-1} + ... + b_0$. By exponent rules, $a^b = a^{b_k 2^k} \bullet a^{b_{k-1} 2^k} \bullet ... \bullet a^{b_0 2^0}$. At each step, a represents $a^{2^i} \bmod n$. d is an accumulator that represents the past i-1 exponentiations. Every '1' bit value for b_i necessitates another multiplication since b_i can only be the values 0 and 1 and $a^{0 \bullet 2^i} = 1$ so when b_i is 0, no multiplication needs to be done. Since it is true that for some x where $x \equiv a \bmod n$ and y where $y \equiv b \bmod n$, the product xy satisfies $xy \equiv ab \bmod n$, modulus can be multiplied through products. Here, we have $a' \equiv a^2 \bmod n$ and so for $d' = da'$, $da' \equiv da^2 \bmod n$. This applies from i = 1 to k. For i = 0, if b_i= 1 it is trivially true that d would take the correct value $d = 1 \bullet a^1 \bmod n = a^{b_0 2^0}$. Since the return value d for the algorithm takes the correct value at each step, it is correct.

31.7-1)

The prime factorization of n=319 is 11,29. $\phi(n) = 319(1 - \frac{1}{11})(1 - \frac{1}{29}) = 280$.

The modular inverse is 187, found by the euclidean algorithm $3^{-1} \equiv x \bmod 280$, 280 = 3(93) + 1, 1 = 280 - 93 (3), $3^{-1} \equiv -93 \equiv 187 \bmod 280$.

The secret key should be S=(187,319).
The public key is P=(3,319).
P(100) = 100^3 mod 319 = 254