1. Alice uses a trick to speed her RSA signature generation. Suppose her modulus is \( n = pq \), where the primes \( p \) and \( q \) have about the same length. Let \( b \) be the number of bits in \( n \), so that the length of \( p \) and \( q \) is about \( b/2 \) bits. If the decryption exponent is \( d \), then Alice signs the plaintext \( M \) as \( S = D(M) = M^d \mod n \). The trick replaces this fast exponentiation with \( b \)-bit numbers by two fast exponentiations with \( b/2 \)-bit numbers. (This makes the signature generation run about four times faster.) Let \( M_p = M \mod p \), \( M_q = M \mod q \), \( d_p = d \mod (p-1) \) and \( d_q = d \mod (q-1) \). The length of each of these four numbers is about \( b/2 \) bits. Alice computes \( S_p = M_p^{d_p} \mod p \) and \( S_q = M_q^{d_q} \mod q \) by fast exponentiation. Now the signature \( S \equiv S_p \pmod{p} \) and \( S \equiv S_q \pmod{q} \), so Alice computes \( S = D(M) \) from \( S_p \) and \( S_q \) by the Chinese remainder theorem. In the application of the Chinese remainder theorem, some numbers may be precomputed. The result is that \( S = (fS_p + gS_q) \mod n \) where \( f \) and \( g \) are precomputed constants.

Find the constants \( f \) and \( g \).

2. Find all the square roots of 58 modulo 77. Show all your work. Use an algorithm which would work for 200-digit numbers in place of 2-digit numbers, assuming the factorization of the modulus is given.

3. Alice and Bob use the coin-tossing protocol described in class. Alice chooses 200-digit primes \( p \) and \( q \) as her secret. Because of a defect in his random number generator, Bob always chooses \( x \) in \( \sqrt{n} < x < n/10000 \). Does this fault matter?

   a. Can either Alice or Bob almost always win the coin toss? Explain your answer.

   b. Would the same fault in the random number generator (that is, Bob always choosing \( c \) in \( \sqrt{n} < c < n/10000 \)) affect the zero-knowledge proof protocol described in class? Explain your answer.