1. Find the solution of the system\
\[
\begin{align*}
x &\equiv 3 \pmod{5} \\
x &\equiv 2 \pmod{6} \\
x &\equiv 1 \pmod{7}
\end{align*}
\]
in $\mathbb{Z}_{210}$, using the Chinese Remainder Theorem and the extended Euclid’s algorithm. Show all your work.

2. On number theory basics.

(a) Show that the RSA decryption operation is correct (i.e., $(x^e)^d \mod n = x$) for all $x \in \mathbb{Z}_n$ even if $x \notin \mathbb{Z}_n^*$. (Hint: Show the correctness both in $\mathbb{Z}_p$ and in $\mathbb{Z}_q$, and argue by the CRT that it must be correct in $\mathbb{Z}_n$.)

(b) Show that, for a prime $p$,
\[
\varphi(p^i) = (p - 1)p^{i-1}.
\]

(c) Show that, for co-prime $m_1$ and $m_2$,
\[
\varphi(m_1m_2) = \varphi(m_1)\varphi(m_2).
\]

(d) Use the results in the previous two parts to obtain $\varphi(n)$ for an arbitrary $n$. (Hint: Consider the prime factorization of $n$, and then combine the previous results by the CRT to obtain $\varphi(n)$.)

3. Alice and Bob are very good friends and don’t mind sharing the same RSA modulus $n$. Of course, to have their own different private keys, they use different public exponents, $e_1$, $e_2$. Moreover $e_1$ and $e_2$ are relatively prime. A common friend Charlie sends a message $x$ to both, encrypting it with their respective RSA keys, $y_1 = x^{e_1} \mod n$, $y_2 = x^{e_2} \mod n$. Show how Eve, who knows the public keys of Alice and Bob and observes the ciphertexts $y_1$ and $y_2$, can find out the message $x$. Describe explicitly how you use Extended Euclidean Algorithm in your solution.
4. On ElGamal signatures. (You can assume that \( g \) has a prime order \( q \) instead of \( p - 1 \), if you like.)

(a) Show that if Eve can learn the value of \( k \) Alice used in an ElGamal signature, she can compute Alice’s private key.

(b) Suppose Alice’s random number generator is broken and it always produces the same \( k \) value. How can Eve detect this from the signatures Alice issues?

(c) Knowing that Alice used the same \( k \) value in two different signatures, describe how Eve can compute that \( k \) value used, and then Alice’s private key \( \alpha \).

5. A protocol to establish a fresh session key using long-term, certified Diffie-Hellman public keys is as follows:

- The system has a common prime modulus \( p \) and a generator \( g \). Each party \( i \) has a long-term private key \( \alpha_i \in \mathbb{Z}_{p-1} \) and a public key \( P_i = g^{\alpha_i} \mod p \).

- To establish a session key between \( A \) and \( B \), party \( A \) generates a random \( R_A \in \mathbb{Z}_{p-1} \), computes \( X_A = \alpha_A + R_A \mod p - 1 \), and sends \( X_A \) to \( B \). Similarly, \( B \) computes a random \( R_B \in \mathbb{Z}_{p-1} \), \( X_B = \alpha_B + R_B \mod p - 1 \), and sends \( X_B \) to \( A \).

- \( A \) computes the session key as 
  \[
  K_{A,B} = (g^{X_B} P_B^{-1})^{R_A} \mod p
  \]
  and \( B \) computes 
  \[
  K_{B,A} = (g^{X_A} P_A^{-1})^{R_B} \mod p.
  \]

(a) Show that the protocol is correct (i.e., \( K_{A,B} = K_{B,A} \)).

(b) Show that a passive attacker Trudy who has broken a session key \( K_{A,B} \) between Alice and Bob can compute any future session keys between these two parties.

(c) Describe a simple addition to the session key computation which will preclude this and any similar attacks on this protocol.