1. Find all integer solutions to the following sets of equations.

a) \[ x \equiv 7 \pmod{15} \]
\[ x \equiv 8 \pmod{16} \]

Putting these equations together, we get (for arbitrary \( k \) and \( j \)):
\[ x = 7 + 15k = 8 + 16j \]
\[ 7 + 15k = 8 + 16j \]
\[ 15k = 1 + 16j \]
\[ 15k \equiv 1 \pmod{16} \]
\[ k = 15^{-1} \cdot 1 \]
\[ k = 15^{-1} \]

Now, we need to compute \( 15^{-1} \). We could use Euclid’s algorithm, but 15 is sufficiently small to do this by inspection. We note that \( 15 \times 15 = 225 \) and \( 14 \times 16 = 224 \). So, we get that \( k = 15^{-1} = 15 \).

Substituting this back into our formula for \( x \) we get:
\[ x = 7 + 15 \times 15 \]
\[ x = 232 \]

This is one solution. The set of all solutions is given by
\[ x = 232 + 240m \forall m \in \mathbb{Z} \]

b) \[ x \equiv 7 \pmod{15} \]
\[ x \equiv 8 \pmod{18} \]

Putting these equations together, we get (for arbitrary \( k \) and \( j \)):
\[ x = 7 + 15k = 8 + 18j \]
\[ 7 + 15k = 8 + 18j \]
\[ 15k = 1 + 18j \]
\[ 15k \equiv 1 \pmod{18} \]

But, in this case 15 and 18 are co-prime. So \( 15^{-1} \) does not exist. We check all values of \( k \), but cannot find any such \( k \) that satisfy this equation. Thus, we must conclude that there are no solutions.
2. Suppose \((G, *)\) is a group. Define a new operation \(\star\) on \(G\) to be:

\[ g \star h = h \ast g \]

Prove that \((G, \star)\) is also a group.

Recall that for a pair \((G, \star)\) to be a group, it must satisfy four properties: the operation must be closed under \(G\), the operation must be associative, there must be an identity element, and each element must have an inverse.

We know that \((G, \ast)\) is a group, so it satisfies all of those properties. We use this to show that \((G, \star)\) is also a group.

Closure:
\((G, \ast)\) is closed, so we have \(\forall g, h \in G, g \ast h \in G\). Since this is true for all \(g, h\), it must be true that \(\forall g, h \in G, h \ast g = g \ast h \in G\). Thus, we get that \((G, \star)\) is closed.

Identity:
\((G, \ast)\) has an identity element (let’s call it \(e\)) such that \(\forall g \in G, e \ast g = g \ast e = g\). But this means that for \((G, \star)\) we get the following \(\forall g \in G, g \star e = e \star g = g\). So, \(e\) is an identity element for \((G, \star)\).

Inverse:
Each element in \((G, \ast)\) has an inverse. That is, \(\forall g \in G, \exists h \in G\) s.t. \(g \ast h = h \ast g = g\). That means \(\forall g \in G, \exists h \in G\) s.t. \(g \star h = h \star g = g\), implying that every element in \((G, \star)\) has an inverse.

Associativity:
The operation \(\ast\) is associative. That is for all \(g, h, j\):

\[ g \ast (h \ast j) = (g \ast h) \ast j \]
\[ g \ast (j \star h) = (g \star h) \ast j \]
\[ (j \star h) \star g = j \star (h \star g) \]

This means that \(\star\) is associative.

Thus, we have that \((G, \star)\) satisfies all the four properties of groups, so it is indeed a group.
3. Consider the group \((\mathbb{Z}_7^*, \otimes)\). Show that this group is isomorphic to \((\mathbb{Z}_6, \oplus)\), and find an isomorphism \(f\). Furthermore, show that each element \(g\) is a generator of \((\mathbb{Z}_7^*, \otimes)\) if and only if \(f(g)\) is a generator of \((\mathbb{Z}_6, \oplus)\).

Let us write out the operation tables for each group.

\[
\begin{array}{cccccccc}
\otimes & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
1 & 1 & 2 & 3 & 4 & 5 & 6 \\
2 & 2 & 4 & 6 & 1 & 3 & 5 \\
3 & 3 & 6 & 2 & 5 & 1 & 4 \\
4 & 4 & 1 & 5 & 2 & 6 & 3 \\
5 & 5 & 3 & 1 & 6 & 4 & 2 \\
6 & 6 & 5 & 4 & 3 & 2 & 1 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
\oplus & 0 & 1 & 2 & 3 & 4 & 5 \\
\hline
0 & 0 & 1 & 2 & 3 & 4 & 5 \\
1 & 1 & 2 & 3 & 4 & 5 & 0 \\
2 & 2 & 3 & 4 & 5 & 0 & 1 \\
3 & 3 & 4 & 5 & 0 & 1 & 2 \\
4 & 4 & 5 & 0 & 1 & 2 & 3 \\
5 & 5 & 0 & 1 & 2 & 3 & 4 \\
\end{array}
\]

We propose the following isomorphism between these two groups \(f : \mathbb{Z}_6 \rightarrow \mathbb{Z}_7^*\).

\[
f(0) = 1 \\
f(1) = 3 \\
f(2) = 2 \\
f(3) = 6 \\
f(4) = 4 \\
f(5) = 5
\]

Let us verify by looking at a combined table, where the two different colours indicate the two different groups.

\[
\begin{array}{cccccccc}
\oplus & \otimes & 0 & 1 & 1 & 3 & 2 & 2 & 3 & 6 & 4 & 4 & 5 & 5 \\
\hline
0 & 1 & 0 & 1 & 1 & 3 & 2 & 2 & 3 & 6 & 4 & 4 & 5 & 5 \\
1 & 3 & 1 & 3 & 2 & 2 & 3 & 6 & 4 & 4 & 5 & 5 & 0 & 1 \\
2 & 2 & 2 & 2 & 3 & 6 & 4 & 4 & 5 & 5 & 0 & 1 & 1 & 3 \\
3 & 6 & 3 & 6 & 4 & 4 & 5 & 5 & 0 & 1 & 1 & 3 & 2 & 2 \\
\end{array}
\]
We see that in the table each element always appears beside its corresponding element in the isomorphism. So this is indeed a valid isomorphism.

Now, we show that \( f(g^n) = f(g)^n \) for all \( n \). We show this by induction.

Basis step: \( f(g) = f(g) \). This is clearly true, so the base case is satisfied.

Induction hypothesis: Suppose that \( f(g^k) = f(g)^k \).

Now, we use the property that for any isomorphism we have \( f(g \oplus h) = f(g) \otimes f(h) \). In particular, we get

\[
\begin{align*}
  f(g^{k+1}) &= f(g) \otimes f(g) \\
  f(g^{k+1}) &= f(g)^k \otimes f(g) \quad \text{(by the induction hypothesis)} \\
  f(g^{k+1}) &= f(g)^{k+1}
\end{align*}
\]

By induction, the original statement \( f(g^n) = f(g)^n \) must be true for all \( n \).

Thus, we get that if \( g \) generates all elements of \( \mathbb{Z}_6 \), then the sequence \( f(g^k) \) generates all elements of \( \mathbb{Z}_7^* \), which is equal to the sequence \( f(g)^k \), so we get that \( f(g) \) is a generator for \( \mathbb{Z}_7^* \).

On the other hand, we get that if \( g \) does not generate all elements of \( \mathbb{Z}_6 \), then the sequence \( f(g^k) \) cannot generate all elements of \( \mathbb{Z}_7^* \), which is equal to the sequence \( f(g)^k \), so we get that \( f(g) \) is not a generator for \( \mathbb{Z}_7^* \).

Thus, an element \( g \) is a generator of \( (\mathbb{Z}_7^*, \otimes) \) if and only if \( f(g) \) is a generator of \( (\mathbb{Z}_6, \oplus) \).

Remark: This solution does not use any particular properties of \( (\mathbb{Z}_6, \oplus) \) and \( (\mathbb{Z}_7^*, \otimes) \), and thus can be straight-forwardly extended to apply to any two isomorphic cyclic groups. In this question it would also suffice to enumerate all of the generators of each of the two groups and show that they correspond in the isomorphism.
4. Suppose Alice wishes to send the same secret message using Public Key Cryptography to her two friends Bob and Charlie. Bob and Charlie know about each other and can communicate (but, of course, must also do so securely).

There are several ways to do this. Alice could separately send the message to Bob and Charlie (using Public Key Cryptography separately for her two friends). Alternatively, if her two friends can communicate, Alice could send the message to Bob and subsequently Bob could send the message to Charlie (both using Public Key Cryptography).

Another method would be for Bob and Charlie to share a decryption function. Then, Alice only needs to encrypt the message once. She can send the same encryption of the message to both Bob and Charlie. Describe in detail how such a protocol could work.

The idea is that Bob and Charlie share the secret key used to encrypt Alice’s message. But, they need to use Public Key Cryptography to share the secret key.

We propose the following protocol:

Bob generates encryption and decryption functions \( E_1, D_1 \).
Charlie generates encryption and decryption functions \( E_2, D_2 \).
Charlie sends \( E_2 \) to Bob.
Bob encrypts \( E_2(D_1) \) and sends the result to Charlie.
Charlie computes \( D_2(E_2(D_1)) \) to recover \( D_1 \).

Now, Bob and Charlie have a shared decryption function \( D_1 \).

Bob sends \( E_1 \) to Alice.
Alice encrypts \( E_1(M) \) and sends the same encrypted message \( E_1(M) \) to both Bob and Charlie.
Bob and Charlie separately compute \( D_1(E_1(M)) \) to recover \( M \).

Bob and Charlie both recovered the message \( M \), but Alice only sent one encryption of the message, and neither Bob nor Charlie had to forward the message to the other.
5. Suppose Alice and Bob communicate using the standard RSA cryptography method.

a) Suppose that their public key is \((n=391, e=25)\). Given that \(\varphi(n)=384\), what is the private decryption key \(d\)?

b) Suppose that their public key is \((n=247, e=49)\). Given that a factoring of 247 is 247\(=13\times19\), what is the private decryption key \(d\)?

Let us recall how we generate the two encryption functions using the RSA method.
We picked our modulus \(n=pq\) as a product of two primes, such that we can easily compute \(\varphi(n)=(p-1)(q-1)\). Then we picked our encryption and decryption exponents \(e, d\) such that \(ed\equiv1\pmod{\varphi(n)}\).

a) We don’t actually need \(n\) in this instance. We can compute \(d=e^{-1}\) with respect to the modulus \(\varphi(n)\).

We could compute \(e^{-1}=25^{-1}\) by inspection or Euclid’s method. Since 384 is rather large, let us use Euclid’s method.

\[
\begin{align*}
9 &\equiv 384 \pmod{25} \\
7 &\equiv 25 \pmod{9} \\
2 &\equiv 9 \pmod{7} \\
1 &\equiv 7 \pmod{2} \\
384 &\equiv 15\times25 + 9 \\
25 &\equiv 2\times9 + 7 \\
9 &\equiv 1\times7 + 2 \\
7 &\equiv 3\times2 + 1
\end{align*}
\]

So, we get that \(25^{-1}=169\), thus the private key is \(d=169\).

b) We know that a factoring of \(n=13\times19\). So, we get \(\varphi(n)=(13-1)(19-1)=216\). We now need to compute \(d=e^{-1}\) with respect to the modulus \(\varphi(n)\).

We could compute \(e^{-1}=49^{-1}\) by inspection or Euclid’s method. Since 216 is rather large, let us use Euclid’s method.

\[
\begin{align*}
20 &\equiv 216 \pmod{49} \\
9 &\equiv 49 \pmod{20} \\
2 &\equiv 20 \pmod{9} \\
1 &\equiv 9 \pmod{2} \\
216 &\equiv 4\times49 + 9 \\
49 &\equiv 2\times20 + 9 \\
20 &\equiv 2\times9 + 2 \\
9 &\equiv 4\times2 + 1
\end{align*}
\]
\[ 1 = 9 - 4 \times 2 \]
\[ 1 = 9 - 4 \times (20 - 2 \times 9) \]
\[ 1 = 9 \times 9 - 4 \times 20 \]
\[ 1 = 9 \times (49 - 2 \times 20) - 4 \times 20 \]
\[ 1 = 9 \times 49 - 22 \times 20 \]
\[ 1 = 9 \times 49 - 22 \times (216 - 4 \times 49) \]
\[ 1 = 97 \times 49 - 22 \times 216 \]

So, we get that \( 49^{-1} = 97 \), thus the private key is \( d = 97 \).