Reading:

1. Péter Gács's slides page 26 - 52.
2. Chapter 1 section 1 to 5 Discrete Mathematics and Its Applications (7th Edition).

Please write out the intermediate steps when you construct the truth tables. For example, to do $((p \wedge q) \rightarrow (p \vee q))$, you might want to have a column for $p \wedge q$ and another for $p \vee q$, before you calculate results for $((p \wedge q) \rightarrow (p \vee q))$.

The enumeration of the truth assignment needs to be in an increasing order in binary encoding, e.g., 000, 001, ..., 111.

1 Truth Tables

1: Construct Truth Tables (10 pts) **

Construct a truth table for each of these compound propositions. Note that in the following $p \leftrightarrow q$ is defined as $(p \leftarrow q) \wedge (q \leftarrow p)$. Do you reading to find truth table for the operation.

1. $((p \wedge q) \rightarrow (p \vee q))$.
2. $(p \vee q) \rightarrow (p \oplus q)$.
3. $(p \leftrightarrow q) \oplus (\neg p \leftrightarrow \neg r)$

Here is an example of a truth table by LaTeX.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

2: Logical equivalences (10 pts) ***

You can show the following logical equivalences by truth tables or by equivalence laws.

1. Show that $(p \rightarrow q) \vee (p \rightarrow r)$ and $p \rightarrow (q \vee r)$ are logically equivalent.
2. Show that $\neg p \rightarrow (q \rightarrow r)$ and $q \rightarrow (p \vee r)$ are logically. equivalent

3: First Order Logic On Finite Domain (10 pts) **

Suppose the domain of the propositional function P (x, y) consists of pairs x and y, where $x \in \{1, 2\}$ and $y \in \{1, 2, 3\}$. Write out the following propositions using disjunctions ($\vee$) and conjunctions ($\wedge$). (This problem help you see that in finite domain you can express the same set of statement without quantifiers.)

(a) $\forall x \forall y P(x, y)$,

(b) $\exists x \exists y P(x, y)$,

(c) $\exists x \forall y P(x, y)$,

(d) $\forall y \exists x P(x, y)$.

4: Nested Quantifiers On Finite Domain and Loops (10 pts) ***

In Page 58 of Discrete Mathematics and Its Applications (7th Edition) there is a remark "THINKING OF QUANTIFICATION AS LOOPS". Suppose now you need to write a code block to check if a first order logic statement with nested quantifier to be true or not. For example, let P(x,y) be the same as Question 1, and x, y now range over 0 to 100 (not including 100), to check if $\forall x \forall y P(x, y)$, you might want to do

```
boolean AA_P_xy() // A for ``forall'',
{
    for(i=0;i<100; i++)
    {
        for(j=0;j<100; j++)
        {
            if(P(x,y)==false)) return false; // one pair of such (x,y) is enough
        }
    }
    return true;
}
```

Listing 1: Java example

You tasks, write code block for checking

1. $\exists x \exists y P(x, y)$, name your function as "EE_P_xy".
2. $\exists x \forall y P(x, y)$, name your function as "EA_P_xy".
3. $\forall x \exists y P(x, y)$, name your function as "AE_P_xy".

5: contrapositve proof (10 pts)

Should that if $a + b + c > 2022$, then one of the following must hold:

(a) $a > 2000$.

(b) $b > -10 + \sqrt{3}$.

(c) $c > 32 - \sqrt{3}$.