1. Prove that whenever \( p_1, \ldots, p_n \) is a list of two or more propositions,

\[ \neg(p_1 \lor p_2 \lor \ldots \lor p_n) \]

is logically equivalent to

\[ \neg p_1 \land \neg p_2 \land \ldots \land \neg p_n \]

Use mathematical induction, and the fact that \( \neg(p \lor q) \) is equivalent to \( \neg p \land \neg q \) (De Morgan’s law).

2. Prove by induction that if \( a \equiv b \mod m \) then \( a^n \equiv b^n \mod m \) for all \( n \geq 0 \).

3. Verify that the program segment

\[
\begin{align*}
\text{if } x < y & \text{ then } \\
m & := x \\
\text{else} & \\
m & := y
\end{align*}
\]

is correct with respect to the initial assertion \( T \) and the final assertion

\[
(x \leq y \land m = x) \lor (x > y \land m = y)
\]

4. This program computes quotients and remainders:

\[
\begin{align*}
r & := a \\
q & := 0 \\
\text{while } r \geq d & \text{ begin} \\
& r := r - d \\
& q := q + 1 \\
\text{end}
\end{align*}
\]

The program assumes that \( d > 0 \) and \( a > 0 \).

Prove that

\[
d > 0 \land 0 \leq r \leq a \land a = dq + r
\]

is a loop invariant for the while loop. In other words, show that if

\[
d > 0 \land 0 \leq r \leq a \land a = dq + r \land r \geq d
\]

is true at the beginning of any iteration of the loop, then

\[
d > 0 \land 0 \leq r \leq a \land a = dq + r
\]

is true afterwards.

5. Briefly, why does this invariant guarantee that the program can only terminate with a correct answer.