1. Structural Induction
(a) Consider the following recursive definition of strings \( \Sigma^* \) over the alphabet \( \Sigma \).

**Basis Step:** \( \varepsilon \) is a string

**Recursive Step:** If \( w \) is a string and \( a \in \Sigma \) is a character, then \( wa \) is a string.

Recall the following recursive definition of the function \( \text{len} \):

\[
\text{len}(\varepsilon) = 0 \\
\text{len}(wa) = 1 + \text{len}(w)
\]

Now, consider the following recursive definition:

\[
\text{double}(\varepsilon) = \varepsilon \\
\text{double}(wa) = \text{double}(w)aa.
\]

Prove that for any string \( x \), \( \text{len}(\text{double}(x)) = 2\text{len}(x) \).

(b) Consider the following definition of a (binary) **Tree**:

**Basis Step:** \( \bullet \) is a Tree.

**Recursive Step:** If \( L \) is a Tree and \( R \) is a Tree then \( \text{Tree}(\bullet, L, R) \) is a Tree.

The function \( \text{leaves} \) returns the number of leaves of a Tree. It is defined as follows:

\[
\text{leaves}(\bullet) = 1 \\
\text{leaves}(\text{Tree}(\bullet, L, R)) = \text{leaves}(L) + \text{leaves}(R)
\]

Also, recall the definition of \( \text{size} \) on trees:

\[
\text{size}(\bullet) = 1 \\
\text{size}(\text{Tree}(\bullet, L, R)) = 1 + \text{size}(L) + \text{size}(R)
\]

Prove that \( \text{leaves}(T) \geq \frac{\text{size}(T)}{2} \) for all Trees \( T \).

2. Regular Expressions
(a) Write a regular expression that matches base 10 non-negative numbers (e.g., there should be no leading zeroes).

(b) Write a regular expression that matches all non-negative base-3 numbers that are divisible by 3.

(c) Write a regular expression that matches all binary strings that contain the substring "111", but not the substring "000".