1. Use induction to prove that the following function \( g(x, y) \) computes the GCD of \( x \) and \( y \), for all positive integers \( x \) and \( y \):

\[
g(x, y) = \begin{cases} 
x & \text{if } x = y \\
g(x - y, y) & \text{if } x > y \\
 g(x, y - x) & \text{if } x < y \\
\end{cases}
\]

2. Read the descriptions of the three algorithms for computing GCD that appear in Levitin §1.1: Euclid’s Algorithm, the Consecutive Integer checking algorithm, and the middle-school procedure.

Implement Euclid’s algorithm and the consecutive integer checking algorithm in your favorite programming language. Measure their execution times, and compare how long each takes to run on pairs of natural numbers of 8, 16, 32, 64, 128, 256 and 512 digits. (Depending on your language, you may need to use a special package to represent large integers.)

3. Use induction to prove that

\[ 2^{n+1} - 1 = 2^0 + 2^1 + 2^2 + \cdots + 2^n \]

4. Prove that

\[ a^{\log_b x} = x^{\log_b a} \]

5. Prove that

\[ \log_b a = \frac{1}{\log_a b} \]

6. Use induction to prove that any postage amount (in whole cents) greater than 7¢ can be made up from a sufficient number of 5¢ stamps and 3¢ stamps.