SAMPLE MIDTERM QUESTIONS

1. a) Show that $p \to q \equiv \neg p \vee q$.

   b) Let p be the proposition "Every person older than 18 years in this country can have a driving license,"
   and let $A(x)$ and $B(x)$ be the predicates,
   $A(x)$: "x is in this country"
   $B(x)$: "x is older than 18 years"
   $C(x)$: "x can have a driving license"
   where the universe of discourse is the set of all people in the world.
   Express the proposition p by using quantifiers, logical connectives and the predicates $A(x)$, $B(x)$ and $C(x)$. Then form the negation of the proposition so that no negation is to the left of a quantifier. Next, express the negation in simple English.

   c) Suppose that the proposition "every person older than 18 years in this country can have a driving license" is true, and also that a person x_1 in this country can not have a driving license. Does it mean that x_1 is not older than 18 years? Explain your answer.

2. a) Use mathematical induction to show that $\forall n \in Z^+$

$$\neg(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \equiv \neg p_1 \vee \neg p_2 \vee \cdots \vee \neg p_n.$$

   b) Show that $\neg\forall x\ A(x) \equiv \exists x\ \neg A(x)$, where $A(x)$ is a predicate and the universe of discourse is a finite set with elements $x_1, x_2, ..., x_n$.

3. a) Prove or disprove that 5^x is $O(5^{2x})$.

   b) Prove or disprove that 5^x is $\Theta(5^{2x})$.
   (In both parts, use the definition of big-O notation.)

4. a) Show that the inverse of a modulo m is **unique** in modulo m if it exists, that is, if $\gcd(a, m) = 1$.

   b) Show that the linear congruence $ax \equiv b \pmod{m}$ has a **unique** solution in modulo m whenever $\gcd(a, m) = 1$.
   (Hint: In both parts, use proof by contradiction to show uniqueness. Assume two different solutions exist and show that it is not possible.)

5. Show **by strong induction** that $\forall n \in Z^+,\ f(n) < (\frac{5}{3})^n$, where $f(n)$ is the n-th Fibonacci number defined by the following recurrence relation:

$$f(n) = f(n-1) + f(n-2) \qquad n \geq 2, \quad \text{and} \quad f(0) = 0, \quad f(1) = 1.$$

6. Consider the following pseudocode:

function Compute(a: real number, n: positive integer)
if $n = 1$ **then**
    $p = a$
else
    $p = a + \text{Compute}(a, n - 1)$
return p

a) Find out what this function computes given a and n. Verify your answer **by induction**.

b) Write down the complexity function $T(n)$ of the above algorithm as a recurrence relation.

c) Find big-Θ (big Theta) complexity of the algorithm. Measure the complexity in terms of comparison and addition operations.