1 Evaluating Quantifications

Evaluate each of the quantifications below. State whether they are true or false, and explain why this is true.

You should assume that all variables are in the domain \( \mathbb{Z} \) (that is, the integers) - although a few quantifications will also limit the domain to only some integers.

(a) \( \forall x \forall y, xy \geq 0 \)
(b) \( \forall x \exists y, xy \geq 0 \)
(c) \( \exists x \forall y, xy \geq 0 \)
(d) \( \forall (n \geq 3) \exists (1 < f < n), f \mid n \)
(e) \( \exists (n \geq 3) \forall (1 < f < n), f \mid n \)
(f) \( \forall x \exists y, x^2 \geq y \)
(g) \( \exists x \forall y, x^2 \geq y \)
(h) \( \forall (c > 0) \exists n_0 \forall (n > n_0), (n^2 \geq cn) \)

Reminder: \( f \mid n \) means “\( f \) divides \( n \)” - that is, \( f \) is a factor of \( n \). Likewise, \( f \nmid n \) means “\( f \) does not divide \( n \)”.
2 Quantifications and Code

Suppose that I have written a method \( p() \) in Java. It takes two integer parameters, and returns an integer. Both integer inputs are allowed to to range from 0 to 99 (inclusive).

(a) Write a method \( p2() \) in Java, which takes a single parameter \( \text{int} \ x \), and which returns the value of the quantification below.

   \[ \exists y, \ p(x, y) = 0 \]

   Remember: You only need to loop over the valid range of \( p() \) - not the range of all possible ints!

(b) Write a method \( q() \) in Java, which takes no parameters and returns the value of the following quantification:

   \[ \exists x \forall y, \ p(x, y) = 0 \]

   I encourage you to use a helper method - although it’s not required.
3 Induction

Prove each of the following conjectures **using induction**.

(a) **Conjecture:**
\[ \sum_{i=0}^{n} (5^i) = \frac{5^{n+1} - 1}{4}, \quad n \in \mathbb{N} \]

(b) **Instructor’s Note:** After reviewing this, I realized that this conjecture holds for all non-negative values of \( n \), not just the even ones. My solution originally used \( k + 1 \) in the Inductive Step, whereas it should have used \( k + 2 \) - but I didn’t notice, because the algebra worked for \( k + 1 \) as well as \( k + 2 \). I’ve updated the proof below to use \( k + 2 \).

**Conjecture:**
\( (7^n - 4) \) is divisible by 3, for all **even**, non-negative values of \( n \).

(c) **Conjecture:**
\[ \forall n \in \mathbb{Z}^+, (1 \cdot 1! + 2 \cdot 2! + \ldots + n \cdot n!) = (n + 1)! - 1 \]

(d) **Conjecture:**
Use induction over \( n \) to prove that, if \( x > -1 \) and \( n \in \mathbb{Z}^+ \), then \( (1 + x)^n \geq 1 + nx \).

(e) **Conjecture:**
In any three sequential numbers from the Fibonacci sequence, exactly two of the numbers are odd.

**NOTE 1:** For this problem, assume that the first two numbers in the Fibonacci sequence are 0, 1. (However, this conjecture would still be true if you assumed that the first two were 1, 1.)

**NOTE 2:** Remember to consider overlapping sequences! For instance, if you assume something is true about \( F_k, F_{k+1}, F_{k+2} \), then you need to prove something about \( F_{k+1}, F_{k+2}, F_{k+3} \), which has **two values in common** with the previous set.
4 Structural Induction (with example)

Consider the following theorem (and its proof). After you have read this proof, prove the same conjecture again - but this time, use the “root plus subtrees” strategy for structural induction.

Conjecture:
A non-empty k-ary tree with n nodes has n – 1 edges.

Base Case: 1 node
A tree with only a single node has no edges between the nodes, and thus the conjecture holds trivially.

Inductive:
Assume that the conjecture holds for any tree which has exactly n nodes. We will prove that it also holds for any tree which has exactly n + 1 nodes.

Any tree with n + 1 nodes is a tree with n nodes, plus a single new leaf, added as a child of a certain node. The new, larger tree has one more edge than the previous one - but it also has one more node.

By the I.H., any tree with n nodes has n – 1 edges; thus, the tree with n + 1 nodes certainly had n edges.

Thus, the inductive step holds.

Summary
Thus, the conjecture holds for all non-empty trees.

*Note that you can’t assume that it’s a binary tree!

5 Structural Induction (from scratch)

Prove the following conjecture using structural induction:

Conjecture:
A non-empty binary tree with n internal nodes has no more than 2n + 1 nodes.