Question 2: Technical Deep Dive

Section 1: Problem Definition

Goal:
The objective here is for the prover to show that they know an `x` such that `x^2 + x + 7 = 9` without actually revealing the value of `x` itself. This setup is classic zero-knowledge proof (ZKP) territory since we're trying to verify the prover's knowledge of `x` while keeping it hidden.

Public and Private Inputs:
In this ZKP circuit, we can break down inputs as follows:

- **Public Inputs:**
  - The value `9`, which is the expected result of the equation `x^2 + x + 7`.
  - The specific form of the equation itself, `x^2 + x + 7`.
- **Private Inputs:**
  - The actual value of `x`, which the prover knows and wants to keep private.

In simpler terms, the verifier will see the equation and the result (public), but not the value of `x` itself (private).

Section 2: ZK Protocol Selection

Protocol Choice:
For this scenario, I'd go with **Groth16**. There are a few reasons for choosing it:

1. **Efficiency:**
   Groth16 is known for having very efficient proof sizes and verification times. This is helpful in cases like ours where we just need a quick and simple proof of a single equation. The proof size is small, which makes it a practical option here.
2. **Security:**
   Groth16 provides a strong level of security, with good resistance to attacks. It's widely used and has been battle-tested in several real-world applications, making it a reliable choice for privacy and security.
3. **Ease of Implementation:**
   Since this is a fairly simple equation, Groth16 is manageable to implement, especially with existing libraries. It requires a "trusted setup" (a starting phase to generate cryptographic parameters), but for a basic use case, this can be straightforward and relatively efficient to perform.