Begin each problem on a separate page.

**Problem 1  Bob and his socks**

Bob is color-blind. His sister always teases him before going to school by telling him “Your socks are mismatched! Quick! Go change them before the bus comes.” Bob is thus always chasing after the bus in the morning. At some point, however, he realizes that his sister sometimes enjoys making him run even when his socks are matched. Assuming Bob has socks that are of only two different colors, help him design a protocol that he can use with his sister to decide whether she is teasing him or telling him the truth about his mismatched socks.

**Problem 2  Perfectly-Hiding Commitments**

A two-round perfectly-hiding commitment scheme is a triple of efficient algorithms $(\text{Gen}, \text{Com}, \text{Ver})$ satisfying the following properties.

**Correctness:** For all security parameters $n$ and inputs $\alpha$,

$$\Pr[k \leftarrow \text{Gen}(1^n); (c, d) \leftarrow \text{Com}(k, \alpha) : \text{Ver}(k, c, d, \alpha) = 1] = 1$$

**Computationally binding:** For any $n$ and nPPT $C^*$, there is a negligible function $\varepsilon$ such that

$$\Pr[k \leftarrow \text{Gen}(1^n); (c, d_1, d_2, \alpha_1, \alpha_2) \leftarrow C^*(k) : \text{Ver}(k, c, d_1, \alpha_1) = \text{Ver}(k, c, d_2, \alpha_2) = 1 \land \alpha_1 \neq \alpha_2] \leq \varepsilon(n)$$

**Perfectly hiding:** For all $n$ and all inputs $\alpha$ and $\beta$, the following distributions are identical:

$$\{k \leftarrow \text{Gen}(1^n); (c, d) \leftarrow \text{Com}(k, \alpha) : (k, c)\} = \{k \leftarrow \text{Gen}(1^n); (c, d) \leftarrow \text{Com}(k, \beta) : (k, c)\}$$

Consider the following two-round protocol for committing to a $n$-bit value, $\alpha$:

1. $\text{Gen}(1^n)$ randomly selects $k = (p, g, h)$ such that $p$ is a prime of the form $2q + 1$ where $q$ is a $n$-bit prime (i.e., $p$ is a safe prime), and $g$ and $h$ are generators in $\mathbb{QR}_p = \{y \in \mathbb{Z}_p^* : y = x^2 \text{ for some } x \in \mathbb{Z}_p^*\}$.

2. $\text{Com}(k, \alpha)$ for $\alpha$ in $\mathbb{Z}_q$ selects a random $d \in \mathbb{Z}_q$ and outputs the commitment message $c = g^d h^\alpha \mod p$ and the decommitment message $d$.

3. $\text{Ver}(k, c, d, \alpha)$ outputs 1 if and only if $c = g^d h^\alpha \mod p$.

Provide a discrete logarithm assumption for some appropriate group, and show that the above protocol is a perfectly-hiding commitment scheme.
Problem 3  Binding and Hiding, Not!

Prove that there do not exist 2-round commitment schemes that are both statistically hiding and binding (i.e., hiding and binding properties both hold against computationally unbounded Turing machines). Note that the question talks about statistical binding and hiding, not perfectly binding and hiding.

Problem 4  Quasi-Poly Zero-Knowledge Proofs

Assuming the existence of one-way permutations, construct a 3-round proof for graph isomorphism with negligible soundness, that is zero-knowledge using quasi-polynomial time simulators (i.e., running time bounded by \( n^{O(\log n)} \)). Prove that your protocol is complete, sound and zero knowledge.

Problem 5  Constant-Round Zero-Knowledge

Let \((\text{Gen}, \text{Com}, \text{Ver})\) be a perfectly hiding commitment scheme. Consider the following five-round proof system for ISO (the language of all pairs of graphs \((G_0, G_1)\) such that \(G_0\) is isomorphic to \(G_1\)) with negligible soundness error.

1. The prover selects \(g \leftarrow \text{Gen}(1^n)\) and sends \(g\) to the verifier.
2. The verifier chooses a \(n\)-bit random string \(r\), selects \((c, d) \leftarrow \text{Com}(g, r)\) and sends \(c\) to the prover.
3. The prover randomly selects \(n\) graphs \(C_1, \ldots, C_n\) such that \(C_i\) is isomorphic to \(G_0\) and sends \(C_1, \ldots, C_n\) to the verifier.
4. The verifier sends \(d\) and \(r\) to the prover.
5. If \(\text{Ver}(g, c, d, r) = 1\) then for each graph \(C_i\), the prover sends the verifier a random isomorphism mapping \(G_b\) to \(C_i\) if the \(i^{th}\) bit of \(r\) is \(b\).

Prove that the above protocol is zero-knowledge. Remember a proof requires reductions!