1. The LSB is a hardcore bit of RSA.
   Let $n, e$ be an RSA public key. Suppose that there exists a polynomial time algorithm $A$ that given $y = x^e \mod n$ computes the LSB of $x$. Show how to use $A$ as a subroutine to construct a polynomial time algorithm $B$ that solves the RSA problem (i.e., for every $y = x^e \mod n$ the algorithm $B$ computes $x$). Analyze the running time of $B$ as a function of the running time of $A$ and the length of the binary representation of $n$.

2. Let $p$ be a large prime number and $g$ a generator of $\mathbb{Z}_p^*$. Peggy picks at random a secret value $x \in \mathbb{Z}_p$ and calculates $Y = g^x \mod p$.
   (a) Suggest a Zero-Knowledge protocol for the DLOG problem. I.e., a protocol that will let Peggy prove that she knows $x$, the DLOG of $Y$ modulo $n$.
   (b) Prove that your protocol is an interactive proof for the DLOG problem.
   (c) Prove that your protocol is Zero-Knowledge.
   (d) Is your protocol perfect Zero-Knowledge, or computational Zero-Knowledge? Explain.

3. Let $G = (V,E)$ be a graph. Hamiltonian cycle for a graph is a cycle in the graph that visits each vertex exactly once. Finding a Hamiltonian cycle for a graph is known to be a NP-complete problem.
   Peggy wants to prove to Victor that she knows a Hamiltonian cycle for a given graph $G$, without revealing the cycle. So she went to wikipedia, where she found the following protocol:

**Protocol 1**
Repeat $t$ times:
1. Peggy chooses a random permutation $\pi$ on $V$, and sends to Victor $G_1 = (\pi(V), \pi(E))$.
2. Victor chooses a random $i \in \{1, 2\}$ and sends it to Peggy.
3. If $i = 1$, Peggy sends $\pi$ to Victor.
   If $i = 2$, Peggy reveals a Hamiltonian cycle in $G_1$.
4. Victor checks Peggy’s answer.

   (a) Show that this protocol is an interactive proof for the Hamiltonian cycle problem. In particular, show how Peggy calculates her answer in step 3, and how Victor checks her answer in step 4.
(b) Show that this protocol is not Perfect Zero-Knowledge.

Victor suggested changing the protocol by using a commitment scheme:

**Protocol 2**

Repeat $t$ times:

1. Peggy chooses a random permutation $\pi$ on $V$, and computes $G_1 = \pi(G) = (\pi(V), \pi(E))$.
2. For each of the $\frac{|V|(|V|-1)}{2}$ possible edges on $G_1$ Peggy computes a different commitment, and sends the commitments to Victor. I.e., for each pair of vertices $(u_1, u_2) \in \pi(V)$, she sends commitment on 1 if $(u_1, u_2) \in \pi(E)$, and commitment on 0 otherwise.
3. Victor chooses a random $i \in \{1, 2\}$ and sends it to Peggy.
4. If $i = 1$, Peggy sends $\pi$ to Victor, and reveals all the commitments on the graph.
   If $i = 2$, Peggy reveals commitments on $|V|$ edges that form a Hamiltonian cycle in $G_1$.
5. Victor checks Peggy’s answer.

(c) What kind of commitment should we use (with perfect secrecy or perfect binding)? Explain.

(d) Is this protocol Zero-Knowledge? If your answer is yes, prove it, and determine if the protocol is Perfect Zero-Knowledge or Computational Zero-Knowledge. Otherwise, explain what Victor can learn from the protocol. You don’t have to show that the protocol is an interactive proof.

4. Let $S$ be the following 4-bit to 4-bit S-box (taken from Serpent)

$$S[x] = \{1, 13, 15, 0, 14, 8, 2, 11, 7, 4, 12, 10, 9, 3, 5, 6\}$$

Write a small (up to 45 lines) C program to calculate the difference distribution table of $S$. Submit the program and its output.