Q1: Homomorphic Encryption (10 points)

(a) (5 points) Show that the above Elgamal encryption scheme is homomorphic with respect to multiplication.

$$E(m_1) * E(m_2) = (g^{y_1}, m_1 * h^{y_1}) * (g^{y_2}, m_2 * h^{y_2}) = (g^{y_1+y_2}, (m_1 * m_2) * h^{y_1+y_2})$$
$$= (g^y, (m_1 * m_2) * h^y) = E(m_1 * m_2)$$

(b) (5 points) Show that the above Elgamal encryption scheme is not homomorphic with respect to addition.

Assume: $m_1 = 1, m_2 = 3, \mathrm{q} = 5, \mathrm{G} =< \mathrm{q} >, \mathrm{g} = 3, \mathrm{x} = 2, \mathrm{h} = \mathrm{g}^x \; mod \; q = 9 \; mod \; 5 = 4$

Show that: $E(m_1) * E(m_2) = E(m_1 + m_2)$

Recall due to (a): $E(m_1) * E(m_2) = E(m_1 * m_2), hence \; show: E(m_1 * m_2) = \; E(m_1 + m_2)$, or specifically show: $E(1 * 3) = \; E(1 + 3), \; E(3) = \; E(4)$

$$E(3) = (c_1, c_2) = (3^{y_3}, 3 * 4^{y_3})$$
$$E(4) = (c_1, c_2) = (3^{y_4}, 4 \; * 4^{y_4}) = (3^{y_4}, 4^{1+y_4})$$

To equate them, Let $y_4 = y_3 \; and \; show \; that \; c_1 of \; E(3) = c_1 \; of \; E(4), and \; show \; that \; c_2 \; of \; E(3) = c_2 \; of \; E(4)$

$3^{y_3} = 3^{y_3}$ → True for c_1

$$4^{y_3} = 4^{1+y_3}, then \; 1 = 4^1, which \; is \; False \; for \; c_2!$$

Hence, Elgamal Encryption scheme is not homomorphic with respect to addition.

Q2: Homomorphic-Based Yao Millionaire Problem (15 points)

(a) (5 points) Explain why does the Homomorphic based protocol for Yao's millionaire problem (in Lecture 11 slides 22-23) fail when using unpadded RSA?

Because it is insecure, as unpadded RSA produces the same plaintext for the same ciphertext. Furthermore, the matrix T will produce the same encryption for 1 since it is calculated using $C = M^e \; mod \; N$, which will reveal Sender information to the Receiver.

(b) (10 points) Design a protocol that uses unpadded RSA. Verify that your protocol works by implementing your proposed protocol using the notebook file ("Yao RSA.ipnyb").

Done.

Q3: Oblivious Transfer (OT) (10 pts)

(a) (10 points) Design a simple protocol for 1-out-of-n OT starting from 1-out-of-2 OT. Assume that both Alice and Bob are honest-but-curious. i.e., they follow the protocol but from time to time they collect extra information looking for exposing private data about each other. In your protocol, Alice and Bob can access the 1-out-of-2 functionality n times. Explain your protocol n details (Hint: Think of how to extend 1-out-of-2 to 1-out-of-3 and then generalize it to 1-out-of-n)

The sender will have *n* messages, and the receiver has an index *i*, and the receiver wishes to receive the *i*-th message among the sender's messages, without the sender learning *i*. Furthermore, the sender wants to ensure that the receiver receive only one of the *n* messages.

Step 1 Alice	1- Generates an RSA key paid PK = (N,e) and SK = (d) 2- Generate n random values, r0, r1, r2 … r_n, and she sends them to Bob along with PK
Step 2 Bob	Bob picks a value (v) between 0 and n, and select r_v
Step 3 Bob	Bob generates a random value k and blinds it with r_v by computing: $x = r_v + k^e \; mod \; N$ and sends it to Alice
Step 4 Alice	Alice does not know which of r_n Bob did choose. Alice computes $k_0 = (x - r_0)^d \; mod \; N, k_1 = (x - r_1)^d \; mod \; N, \ldots k_n = (x - r_n)^d \; mod \; N$
Step 5 Alice	Alice combines the n secret messages with each of the possible keys, i.e. $m'_0 = m_0 + k_0, m'_1 = m_1 + k_{1,} \ldots m'_n = m_n + k_n$, and she sends them to Bob
Step 6 Bob	Bob knows which of the n messages can be unblinded with k, so he is able to compute exactly one of the messages $m_v = m'_v - k$