Assignment 1
Burhanuddin Mogul
CS 408 001 Cryptography & Internet Security

#1
Consider all numbers that are less than p^c that are not relatively prime to p^c, or all numbers that are a multiple of p, so p, 2p, 3p, … p^(c-1)p. There are p^(c-1) such numbers as all of these numbers divide p^c.
Now to get all the numbers that are relatively prime to p^c we can take the p^(c-1) non-relative prime numbers and subtract them from all numbers up to p^c.
So that would give us p^c – p^(c-1) which would equal
p^c – p^(c-1) = p^(c-1) (p – 1) = (p^c) (1/p) (p -1) = p^c (1 – 1/p)

This shows by counting that φ(p^c) = p^c(1-1/p)

#2
Given a, b, e1, e2, n, GCD(e1,e2) = 1, and a^e1 = b^e2 (mod n) we can prove c^e1 = b (mod n) as follows

As e1 and e2 are relatively prime, we know that there exists integers X and Y such that
e1X + e2Y = 1

We can also see that if we raise both sides of a^e1 = b^e2 (mod n) by Y we get a^e1Y = b^e2Y (mod n)

Let's say c = b^X(a^Y)
Then we can see that c^e1 = (b^X * a^Y)^e1 = b^Xe1 * a^Ye1 = b^e1X * b^e2Y = b ^(e1X + e2Y) = b^1

Therefore, anyone can compute c such that c^e1 = b mod n

#3
A)
If e is set to 1, the plaintext will b the same as the ciphertext as when the plaintext is raised to the power of one and the modulo is taken with a large number, the same value will be returned.

If e is set to 2, it is a small number and does not provide enough variation and finding square roots modulo n is easy. This would allow for decryption without private key

B)
φ(35) = φ(5) * φ(7) = 4 * 6 = 24
We need to choose an e such that GCD(e, φ(n)) = 1, all valid e's are
1, 5, 7, 11, 13, 17, 19, 23

For each of the values above for e, the modular inverse with mod 24 are the same as e, therefore all e equal d for n = 35

C)
Given c ≡ m^3 (mod 101) we know c = 3
φ(101) = 100 as 101 is prime
So we are looking for d such that 3d ≡ 1 (mod 100)
From the Extended Euclidean Algorithm, we know that 67 is inverse of 3 (mod 100)

So d = 67and therefore
m ≡ c^67 (mod 101)

D)
As p is a large *prime* number, we know that $\varphi(p) = p - 1$.
We can find a d such that ed = 1 mod (p-1), or the multiplicative inverse of e modulo (p-1).
The Extended Euclidean Algorithm can be used to find d.

#4
Both schemes are not deterministic as they use a random r which improves upon RSA but:
The first scheme, [A = r^e, B = m + r], there exists a linear relationship with B = m + r, and does not provide any semantic security.

While in the second scheme [A = r, B = (m+r) ^ e], B can be rewritten as (m + A)^ e, and now the attacker can easily find m as they have both A and e.