Problem 1

A. Let $n = \prod_{i=1}^{i=l} p_i^{k_i}$, where $p_1, \ldots, p_l$ are the prime factors of n. Then, by definition, we have $\phi(n) = \prod_{i=1}^{i=l} p_i^{k_i-1}(p_i - 1)$.

Now, if n has an odd prime factor, then corresponding to that factor p_i, the term $p_i - 1$ in the above product is even, and consequently, $\phi(n)$ is even. Similarly, if n is a power of 2, with the corresponding $k_i > 1$, then the term 2^{k_i-1} is even, so $\phi(n)$ is even.

Thus the only number n for which $\phi(n)$ is odd is 2. The value of $\phi(n)$ in this case is 1. ($n = 1$ is also a valid answer, because $\phi(n)$ can be defined to be 1.)

B. Let $d = \prod_{i=1}^{i=l} p_i^{k_i}$. Then, $m = \prod_{i=1}^{i=l'} p_i^{k'_i}$ with $k'_i \geq k_i, \forall i \leq l$. Now we have $\phi(d) = \prod_{i=1}^{i=l} p_i^{k_i-1}(p_i - 1)$, and $\phi(m) = \prod_{i=1}^{i=l'} p_i^{k'_i-1}(p_i - 1)$. Then, $\phi(m)/\phi(d) = \prod_{i=1}^{i=l} p_i^{k'_i-k_i} \times \prod_{i=l+1}^{i=l'} p_i^{k_i-1}(p_i - 1)$, which is an integer. Thus $\phi(d)|\phi(m)$.

Problem 2

A. We first compute $(xSGP + e)P^{-1}$ (note $P^{-1} = P^T$). This gives $xSG + eP^T$; note that there are still t bit errors in this message. Therefore, xSG is a codeword of G with at most t errors, and we know we can recover it in polynomial time. Finally, as S is invertible we can compute S^{-1} and compute $x = xSS^{-1}$ in polynomial time.

B. First generate all 2^k messages. Next, 'encrypt' each message and combine it with each possible n-bit vector with weight t. Finally, map each encoded message back to the original message that generated it. Lookups can be done in time proportional to the message size. The table would require $O(2^k \binom{n}{t} nk)$ space.

C. The initial code can correct 50-bit errors. As we only care about correcting up to 10 bit-errors we can use the extra 40 bits to add an n-bit noise vector with weight 40. The maximum combined errors is at most 50, so we can still recover the message.

D. An advantage of the cryptosystem is that it has an error-correcting mechanism built into it, which as we saw in part (c) we can combine with the encryption. Another advantage is that the McEliece crypsosystem is a candidate for post-quantum cryptography. A possible disadvantage is the size of the obejcts required to encrypt and decrypt messages (both the private and public keys are large matrices).

Problem 3

A. Knowing e_1 and e_2, Eve first computes r and s such that $re_1 + se_2 = 1$. She can do this by using Euclid's algorithm. Then she computes $m_1^r \times m_2^s = m^{re_1} \times m^{se_2} \mod n = m$.

B. We can assume that N_A, N_B and N_C are coprime, otherwise Eve can factor them and obtain the message m. Now we have $m_A = m^3 \mod N_A$, $m_B = m^3 \mod N_B$, and $m_C = m^3 \mod N_C$. Applying the chinese remainder theorem, Eve can compute $m^3 \mod N_A N_B N_C$, because N_A, N_B and N_C are pairwise coprime. Now, $m < \min(N_A, N_B, N_C)$. So, $m^3 < N_A N_B N_C$, and $m^3 \mod N_A N_B N_C$ is simply m^3. Eve simply takes the cube root of this number and obtains the message m.

Problem 4

For each pair (y, z), Bob can compute the message $m = \frac{z}{y^{11}} \mod (x^3 + 2x + 1)$. This can be done by writing a simple program that computes the inverse of y modulo $x^3 + 2x + 1$, and then computes the product $z \times (y^{-1})^{11} \mod (x^3 + 2x + 1)$.

A	1	26	J	$1 + x^2$	7	S	$1 + 2x^2$	16
B	2	13	K	$2 + x^2$	3	T	$2 + 2x^2$	20
C	x	1	L	$x + x^2$	19	U	$x + 2x^2$	25
D	$1 + x$	18	M	$1 + x + x^2$	22	V	$1 + x + 2x^2$	17
E	$2 + x$	11	N	$2 + x + x^2$	8	W	$2 + x + 2x^2$	23
F	$2x$	14	O	$2x + x^2$	12	X	$2x + 2x^2$	6
G	$1 + 2x$	24	P	$1 + 2x + x^2$	10	Y	$1 + 2x + 2x^2$	21
H	$2 + 2x$	5	Q	$2 + 2x + x^2$	4	Z	$2 + 2x + 2x^2$	9
I	x^2	2	R	$2x^2$	15			

Figure 1: Polynomials and their corresponding logs to base x modulo $x^3 + 2x + 1$.

The question, however, provides us a "trapdoor" that greatly simplifies the calculation, and enables us to find the solution by hand. Recall that x is a generator of the group. For any polynomial y in the group, we can easily compute a such that $x^a = y \mod (x^3 + 2x + 1)$. The respective values are displayed in the table on the next page. Now, for each pair (y, z), we first read off the corresponding a_1 and a_2 from the table. Then, the decoded message is simply $zy^{-11} = x^{a_2 - 11a_1 \mod 26}$. We compute $a_2 - 11a_1 \mod 26$ and read off the corresponding letter from the table below. This gives us the message: GALOISFIELD.

Problem 5

Roughly. Randomly pick an number x in Z_n (if $gcd(x, n) > 1$ then you have factored n). Calculate $y = x^2 \pmod n$. Use the square root routine to calculate the square root z of y. If $z \neq \pm x$ then keep x and z, else pick another random number and repeat. With probability 1/2 we will succeed on each trial so after a polylogarithmic number of trials the probability of failure is $1/n^k$. Once we have found $z \neq \pm x$ which are both square roots of y we have $x^2 = z^2 \pmod n$ so $x^2 - z^2 = 0 \pmod n$ and $(x - z)(x + z) = 0 \pmod n$ so $(x - z)$ or $(x + z)$ must not be relatively prime to n and we can find a factor using $gcd(n, x - z)$ and $gcd(n, x + z)$.