1. (Compression)

   (a) Assign probability 1 at the root, and half the value of a node to each of its children.

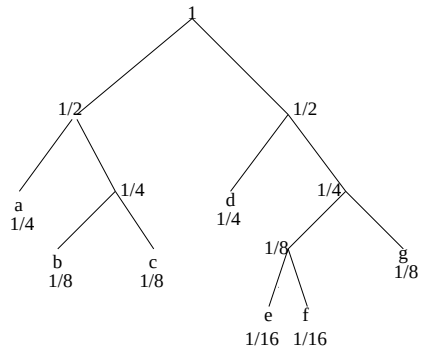

Figure 1: Huffman Tree with Probabilities

   (b) Using the entropy formula, we get (where log is logarithm base 2) $-(1/2 \times \log(1/2)) - (1/4 \times \log(1/4)) - (1/8 \times \log(1/8)) - (1/8 \times \log(1/8)) = 1/2 \times 1 + 1/4 \times 2 + 1/8 \times 3 + 1/8 \times 3 = 1/2 + 1/2 + 3/8 + 3/8 = 1.75$. So, we need at least 1.75 bits per symbol on the average. To transmit a million symbol string, we need to transmit at least 1.75×10^6 bits.

   (c) i. The trie is shown below.

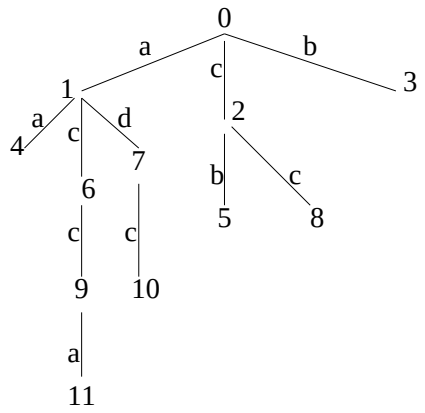

Figure 2: The trie using Lempel-Ziv Code

index	word	transmission
0	$\langle\rangle$	none
1	a	$(0, a)$
2	c	$(0, c)$
3	b	$(0, b)$
4	aa	$(1, a)$
5	cb	$(2, b)$
6	ac	$(1, c)$
7	ad	$(1, d)$
8	cc	$(2, c)$
9	acc	$(6, c)$
10	adc	$(7, c)$
11	$acca$	$(9, a)$

Table 1: The transmissions corresponding to the given trie

ii. First, we compute the dictionary in Table 1.
The transmitted string is the concatenation of strings in the second column: *acbaacbacadccaccadcacca*.

(d) Each pair in an LZ transmission represents a string. Once a pair is transmitted, the corresponding string is added to the dictionary. A pair is created only if the corresponding string is not in the dictionary. Therefore, no pair can occur twice in a transmission.

2. (Error Correction)

(a) For H_0, the answer is 1, by inspection. For H_1, the answer is 01, by inspection. For H_{n+1}, where $n > 0$, we show that the result is a string of 0.
The form of H_{n+1} is

$$H_{n+1} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

where A, B and C are H_n and D is $\overline{H_n}$.
Pair a row in the upper half with the corresponding row in the lower half and take their exclusive-or. The portions of the two rows that belong to A and C are identical; so, the left half of the result is all zeroes. The portions of the two rows that belong to B and D are complements; so, the right half of the result is all ones. Thus, the result is of the form $00\cdots11\cdots$. Doing this for all pairs, we create 2^n rows, where $n > 0$; i.e, an even number of identical rows. Their exclusive-or is a row of zeroes.
You can also prove this result by induction on n. Do as before for H_0 and H_1. For H_2, also prove the result by inspection. Then apply induction.

10	9	8	7	6	5	4	3	2	1
d	d	c	d	d	d	c	d	c	c
1	0	1	1	1	0	0	0	1	1

Table 2: Hamming code encoding of two strings

(b) Since we are required to detect 2 errors and correct 1 error, the minimum distance between codewords has to be at least 3. Let us start with Hadamard matrix, H_3. It provides 8 codewords where the distance between any two codewords is 4. Take out the last bit of each codeword to get 7 bit codewords, where the distance between any two codewords is at least 3 (hence, all these codewords are distinct). Choose any 6 codewords out of this set.

(c) The theorem is correct. Under the condition that we don't know which bit is corrupted, parity code can be used to detect the presence of one error. In the RAID architecture we know which bit is corrupted, i.e., which disk has failed, and then, the corrupted bit can be restored.

(d) The string is 6 bit long. So, we will have to add 4 check bits. The data bits, 101100, are stored at positions 10, 9, 7, 6, 5, 3, positions that are not powers of 2. So ones appear in positions 10, 7, 6; see Table 2. Take the exclusive-or of these positions to get: $1010 \oplus 0111 \oplus 0110 = 1011$, which we store in the check bits, at positions 8, 4, 2, 1 in sequence, as shown in Table 2.

3. (Cryptography)

(a) Factor 55 to get $p = 5$ and $q = 11$. So, $\phi(n)$ is $4 \times 10 = 40$. Given

$$d \times e \overset{\text{mod } \phi(n)}{\equiv} 1, \text{ i.e.,}$$
$$d \times 3 = 40 \times k + 1, \text{ for some } k,$$

we look for the smallest number of the form $40 \times k + 1$ which is divisible by 3. This is 81. So, $d \times 3 = 81$, or $d = 27$.

Alice's message is "01 02". So, we have to compute $(01)^{27} \bmod 55$ and $(02)^{27} \bmod 55$, and then use these numbers as indices to letters in the Roman alphabet.

$(01)^{27} \bmod 55$ is 1. So, the first transmitted letter is "a". Next,

$$\begin{array}{ll} & 2^{27} \bmod 55 \\ = & \{\text{arithmetic}\} \\ & (2^9)^3 \bmod 55 \\ = & \{\text{Modular Simplification Rule}\} \\ & (2^9 \bmod 55)^3 \bmod 55 \\ = & \{\text{simplify: } 2^9 \bmod 55 = 512 \bmod 55 = 17\} \end{array}$$

$$
\begin{aligned}
& 17^3 \bmod 55 \\
= \quad & \{\text{Modular Simplification Rule}\} \\
& ((17^2 \bmod 55) \times 17) \bmod 55 \\
= \quad & \{\text{arithmetic: } 17^2 \bmod 55 = 289 \bmod 55 = 14\} \\
& (14 \times 17) \bmod 55 \\
= \quad & \{\text{arithmetic: } (14 \times 17) \bmod 55 = 238 \bmod 55 = 18\} \\
& 18 \\
\equiv \quad & \{18^{th} \text{ letter of the alphabet is "r"}\} \\
& \text{"r"}
\end{aligned}
$$

Alice's message is "ar".

(b) Let the i^{th} plaintext block be p_i, and the encrypted block be b_i, $i > 0$. Let the secret key be denoted by b_0. Then,

$$b_i = p_i \oplus b_{i-1},\ i > 0$$

Then, for any $i > 0$, $b_i \oplus b_{i-1} = p_i \oplus b_{i-1} \oplus b_{i-1} = p_i$. All b_i except b_0 are available as transmitted blocks. Hence all blocks except block 1 can be decrypted.