Section 1.5

3) **b)**

$$\begin{pmatrix} - & 1 & 0 & - \\ - & 1 & 1 & - \\ - & 1 & 0 & - \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

d)

$$\begin{pmatrix} - & - & - & 1 & 1 \\ - & - & - & 1 & 0 \\ - & - & - & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

5) The null space of H is given by the equations.

$$\begin{aligned} 0 &= c_1 + c_3 \\ 0 &= c_1 + c_2 + c_4 \\ 0 &= c_1 + c_5 \end{aligned}$$

Thus, c_3, c_4, and c_5 are check bits, while c_1 and c_2 are free. We know that $(0, 0, 0, 0, 0)$ is in the null space, and we see from the equations that $(0, 1, 0, 1, 0)$ is as well. These are two code words that have Hamming distance 2. We could have read this second 5-tuple directly from the matrix by noting that the 2nd and 4th columns are identical.

6) **b)** There is no column consisting only of 0's, so it can be used for single-error-detection. However, the first and third columns are identical, so it cannot be used for single-error-correction.

d) The second column of the matrix consists entirely of 0's. Therefore, the null space of this matrix cannot be used for single-error-correcting or -detecting codes.

8) A code with 3 information positions and 3 check positions can be the null space of a 3×6 matrix. To make it single-error-correcting, we need the matrix to have no 0 column and no two columns identical. One example of such a matrix is the following

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

The code words generated by this matrix are:

$$\begin{array}{lll} (000000) & (100110) & (010101) \\ (110011) & (001011) & (101101) \\ (011110) & (111000) & \end{array}$$

10) Recall that for a single-error-correcting group code with m information bits, the number of check bits r must satisfy $2^r - 1 - r \geq m$. The reasoning in this formula is as follows: The group code arises as the null space of a matrix with r rows and $m + r$ columns. Since each column is r bits in length, there are 2^r possible columns that can appear. The first criterion for a single-error-correcting code is that the 0 column cannot appear. This leaves us with $2^r - 1$ possible columns. The other criterion is that no column can be repeated. Therefore, after we pick r distinct columns for the check bits, we are left with $2^r - 1 - r$ possible columns to fill the remaining m places in the matrix. The inequality follows.

Therefore, we must find r such that $2^r - 1 - r \geq 20$. It is easy to check that $r = 5$ works, and is the smallest such value. For $m = 32$, $r = 6$ is the smallest number of check bits that satisfies the condition.

12) Consider the possible values for d. $d = 1$ if and only if (i) is false, by Theorem 1.14. By Theorem 1.15, $d = 2$ if and only if (i) is true and (2) is false.

Now let us consider the case when $d = 3$. Then (i) and (ii) are true, and we must show that (iii) is false. There exists some code word a of weight 3; that is, $W(a) = 3$ and $Ha = 0$. Let i, j, and k be the three places of a that are 1 so that $a = e_i + e_j + e_k$, where e_l has 1 in the lth slot at 0's elsewhere. So

$$0 = Ha = He_i + He_j + He_k.$$

But He_i is the ith column of H, and likewise for j and k. Therefore, the kth column of H is the sum of the ith column and jth column. So (iii) is false.

Now suppose $d \geq 4$. Then (i) and (ii) are true. If the kth column is the sum of the ith column and the jth column, then $H(e_i + e_j + e_k) = 0$. But then $e_i + e_j + e_k$ would be a code word of weight 3, contradiction the fact that the minimum weight must equal d in a group code. Therefore (iii) must be true.

Information Theory

1) $\log_2 4000 = 5 + 3\log_2 5 \approx 11.966$, so to encode 4000 equally likely symbols, 12 bits are needed. $\log_2 5000 = 3 + 4\log_2 5 \approx 12.288$, so to encode 5000 equally likely symbols, 13 bits are needed.

2) The entropy of these probabilities is:

$$-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{8}\log_2\frac{1}{8} - \frac{1}{8}\log_2\frac{1}{8} = \frac{1}{2} + \frac{1}{2} + \frac{3}{8} + \frac{3}{8} = 1.75$$

Following the algorithm in the notes yields this code:

$$\begin{aligned} \text{`}bear' &= 0 \\ \text{`}raven' &= 10 \\ \text{`}eagle' &= 110 \\ \text{`}whale' &= 111 \end{aligned}$$

It is easy to see that this is indeed an optimal code. The code 00, 01, 10, 11 has entropy

$$4\left(-\frac{1}{4}\log_2\frac{1}{4}\right) = \log_2 4 = 2.$$

So our optimal code, with entropy 1.75 is indeed more efficient.

3) BREW can occur because each of three the digraphs in it has a nonzero probability of occuring: 'BR' has probability 1/16; 'RE' has probability 1/8, and 'EW' has probability 1/16. BERW cannot occur because both the digraphs 'BE' and 'RW' have 0 probability of occuring. The entropy for this code is:

$$-\frac{7}{16}\log_2\frac{7}{16} - 5\left(\frac{1}{16}\log_2\frac{1}{16}\right) - 2\left(\frac{1}{8}\log_2\frac{1}{8}\right) = \frac{7}{4} - \frac{7}{16}\log_2 7 + \frac{5}{4} + \frac{3}{4} \approx 2.52$$

bits/digraph.

[As a sidenote: the optimal coding by symbols found in Question 2 would give the following table of probabilities for digraphs:

$first\backslash second$	$bear$	$raven$	$eagle$	$whale$
$bear$	1/4	1/8	1/16	1/16
$raven$	1/8	1/16	1/32	1/32
$eagle$	1/16	1/32	1/64	1/64
$whale$	1/16	1/32	1/64	1/64

One can calculate from this table that the entropy of using the 'optimal coding by symbols' to code digraphs is 3.5 bits/digraph. Therefore, the 'optimal coding by symbols' is not optimal in the sense of coding digraphs.]

An optimal coding by digraphs can be more efficient than an optimal coding by symbols for the following reason: Given two symbols A and B, an 'optimal coding by symbols' would treat the appearance of the digraphs AB and BA with equal probability. This is not necessarily the case, and so just as an 'optimal coding by symbols' is more efficient than a coding that treats all symbols as equally likely, an 'optimal coding by digraphs' would be still more efficient because it takes into consideration the differences in probabilities of appearance between transpositions of symbols.