1. (a) We concatenate the codewords $\psi(f) = 0010$, $\psi(a) = 11$, $\psi(d) = 10$, $\psi(e) = 0011$, and $\psi(d) = 10$ to get the encoded message 00101110001110.

(b) We read the encoded message from left to right, inserting a comma every time we have read a complete codeword $0001, 11, 0010, 0011$. Since $0001 = \psi(c)$, $11 = \psi(a)$, $0010 = \psi(f)$, and $0011 = \psi(e)$, this decodes to $cafe$.

(c) There are many sets of frequencies that work. A very simple one is to assign probability 2^{-n} to every letter with codeword length n, but I will use a slightly different set of frequencies which will be more useful in part (d).

Let us set $p_a = p_d = 0.3$, $p_b = 0.2$, $p_c = p_g = 0.04$, and $p_e = p_f = 0.06$.

We start the construction of the Huffman tree with a set of isolated vertices, In each step of the construction of the Huffman tree, we take the two letters q_1, q_2 with the lowest frequencies, replace them by a new letter r which has frequency $p_r = p_{q_1} + p_{q_2}$, and attach nodes labelled q_1 and q_2.

The step-by-step construction looks as follows:

	alphabet + frequencies	current forest
replace c and g by x_1	a b d e f x_1 0.3 0.2 0.3 0.06 0.06 0.08	a b d e f x_1 / \ c g
replace e and f by x_2	a b d x_2 x_1 0.3 0.2 0.3 0.12 0.08	a b d x_2 x_1 / \ / \ e f c g
replace x_1 and x_2 by x_3	a b d x_3 0.3 0.2 0.3 0.2	a b d x_3 / \ x_2 x_1 / \ / \ e f c g
replace b and x_3 by x_4	a d x_4 0.3 0.3 0.4	a d x_4 / \ b x_3 / \ x_2 x_1 / \ / \ e f c g
replace a and d by x_5	x_5 x_4 0.6 0.4	x_5 x_4 / \ / \ a d b x_3 / \ x_2 x_1 / \ / \ e f c g
replace x_4 and x_5 by x_6	x_6 1	x_6 / \ x_5 x_4 / \ / \ a d b x_3 / \ x_2 x_1 / \ / \ e f c g

This tree corresponds to the given code with the convention that left child means 1, and right child means 0 in the codeword.

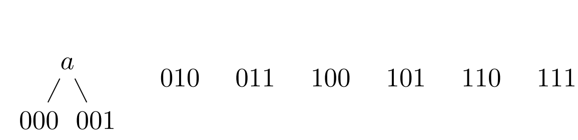
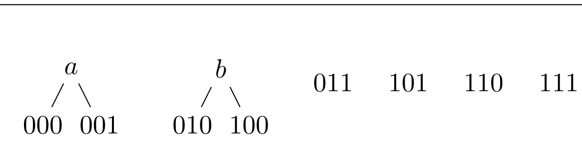
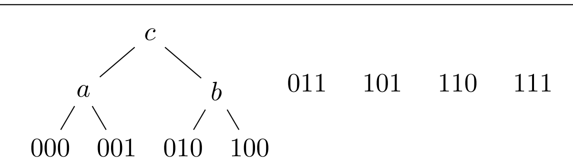
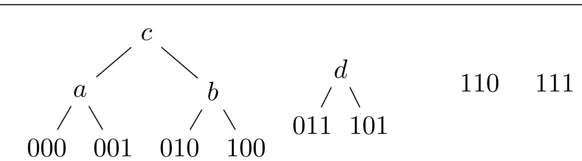
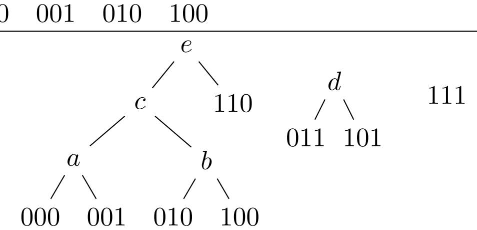
(d) Consider the frequencies $p_a = p_d = 0.3$, $p_b = 0.2$, $p_c = p_e = 0.04$, and $p_f = p_g = 0.06$. Constructing the Huffman code with these frequencies leads to the same codeword lengths as in part (b) because we only swapped the frequencies of the letters e and g whose codewords had the same length. Hence the codeword lengths of the given code ψ are the same as in a Huffman code with these letter frequencies, so ψ is optimal.

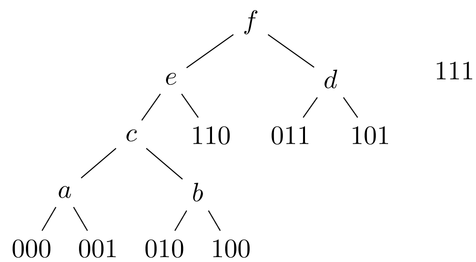
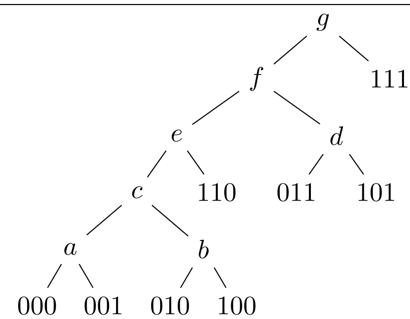
To see that ψ is not a Huffman code, recall that in a Huffman code the lowest frequency letters are always siblings. The letters with the lowest frequencies are c and e, but they are not siblings in the tree corresponding to the code ψ because their codewords differ before the last bit.

2. (a) Since the individual bits are independent, the probability of a codeword is simply the product of the probabilities of the individual bits. This gives

$$p_{000} = \frac{1}{4}\cdot\frac{1}{4}\cdot\frac{1}{4} = \frac{1}{64} \qquad p_{100} = \frac{3}{4}\cdot\frac{1}{4}\cdot\frac{1}{4} = \frac{3}{64}$$
$$p_{001} = \frac{1}{4}\cdot\frac{1}{4}\cdot\frac{3}{4} = \frac{3}{64} \qquad p_{101} = \frac{3}{4}\cdot\frac{1}{4}\cdot\frac{3}{4} = \frac{9}{64}$$
$$p_{010} = \frac{1}{4}\cdot\frac{3}{4}\cdot\frac{1}{4} = \frac{3}{64} \qquad p_{110} = \frac{3}{4}\cdot\frac{3}{4}\cdot\frac{1}{4} = \frac{9}{64}$$
$$p_{011} = \frac{1}{4}\cdot\frac{3}{4}\cdot\frac{3}{4} = \frac{9}{64} \qquad p_{110} = \frac{3}{4}\cdot\frac{3}{4}\cdot\frac{3}{4} = \frac{27}{64}$$

(b) As in problem 1(c), we inductively construct the Huffman tree:

	letter frequencies	current forest
replace 000 and 001 by a	a 4/64, 010 3/64, 011 9/64, 100 3/64, 101 9/64, 110 9/64, 111 27/64	
replace 010 and 100 by b	a 4/64, b 6/64, 011 9/64, 101 9/64, 110 9/64, 111 27/64	
replace a and b by c	c 10/64, 011 9/64, 101 9/64, 110 9/64, 111 27/64	
replace 011 and 101 by d	c 10/64, d 18/64, 110 9/64, 111 27/64	
replace c and 110 by e	e 19/64, d 18/64, 111 27/64	

replace d and e by f	f 111 37/64 27/64	
replace f and 111 by g	g 1	

With the convention that left child means 1, and right child means 0 in the codeword this gives the following code:

$$\begin{array}{llll} \psi(000) = 11111 & \psi(001) = 11110 & \psi(010) = 11101 & \psi(011) = 101 \\ \psi(100) = 11100 & \psi(101) = 100 & \psi(110) = 110 & \psi(111) = 0 \end{array}$$

(c) The average codeword length is given by

$$\sum_{x \in \mathbb{Z}_2^3} p_x \cdot \ell(\psi(x)) = \frac{1}{64} \cdot 5 + \frac{3}{64} \cdot 5 + \frac{3}{64} \cdot 5 + \frac{9}{64} \cdot 3 + \frac{3}{64} \cdot 5 + \frac{9}{64} \cdot 3 + \frac{9}{64} \cdot 3 + \frac{27}{64} \cdot 1 = \frac{79}{32} \approx 2.469$$

Compared to the original encoding where every codeword uses 3 bits, on average we save $\frac{17}{32}$ bits per code word (this is $\approx 17.7\%$).

3. (a) A codeword in Fitingof's code consists of the prefix which has fixed length $\lceil \log_2(14) \rceil = 4$ and the suffix, which has length $\lceil \log_2 \binom{14}{w} \rceil$, where w is the Hamming weight of the element of $\mathbb{Z}_2^{14}$.

The value of $\binom{14}{w}$, and thus also the codeword length is minimised when $w = 0$ or $w = 14$, and $\binom{14}{0} = \binom{14}{14} = 1$. The codeword for an element of Hamming weight 0 or 14 has $4 + \lceil \log_2 1 \rceil = 4$ bits.

The value of $\binom{14}{w}$, and thus also the codeword length is maximised when $w = 7$, and $\binom{14}{7} = 3432$. The codeword for an element of Hamming weight 7 has $4 + \lceil \log_2 3432 \rceil = 4 + 12 = 16$ bits.

(b) the codeword is shorter than 14 bits if and only if $\lceil \log_2 \binom{14}{w} \rceil < 10$, or equivalently $\binom{14}{w} \leq 2^9 = 512$. We see that $\binom{14}{3} = 364 < 2^9$ and $\binom{14}{4} = 1001 > 2^9$. Using the fact that $\binom{n}{k} = \binom{n}{n-k}$ and monotonicity of the binomial coefficients for $k \leq \frac{n}{2}$, we conclude that the codeword has fewer than 14 bits if and only if the hamming weight is either at most 3, or at least 11.

(c) We first determine the prefix. As noted in (a), the prefix length is $\lceil \log_2(14) \rceil = 4$. The Hamming weight of the string 00100000001100 is 3. Hence the prefix is 0011 (the binary representation of 3 using 4 bits).

Next we determine the suffix. Suffix length is $\lceil \log_2 \binom{14}{3} = 9$. The 1s in the bitstring are in positions 3, 11, and 12, hence we compute the ordinal number of this string as

$$\binom{14-12}{1} + \binom{14-11}{2} + \binom{14-3}{3} = 2 + 3 + 165 = 170.$$

The binary representation of 170 is 10101010, so the suffix id 010101010 (because suffix length is 9).

Finally, we obtain the codeword by concatenating prefix and suffix as 0011010101010.

(d) the first 4 bits must be the prefix of the first codeword. Since 0011 is 3 in decimal, this means that the codeword has Hammming weight 3, and therefore the suffix has length $\lceil \log_2 \binom{14}{3} = 9$.

Thus the suffix is 000001100, which translates to 12 in decimal. We need to solve

$$\binom{14-n_3}{1} + \binom{14-n_2}{2} + \binom{14-n_1}{3} = 12.$$

By Proposition 4.30 in the lecture notes, $14 - n_1$ is equal to the largest x for which $\binom{x}{3} \leq 12$. We compute $\binom{5}{3} = 10$ and $\binom{6}{3} = 10$ and conclude that $14 - n_1 = 5$, and thus $n_1 = 9$.

Next, $14 - n_2$ can be found as the maximal x for which $\binom{x}{2} \leq 12 - \binom{14-n_1}{3} = 2$. Since $\binom{2}{2} = 1$ and $\binom{3}{2} = 3$ we see that $14 - n_2 = 2$, and hence $n_2 = 12$.

Finally $14 - n_2$ can be found as the maximal x for which $\binom{x}{2} \leq 2 - \binom{14-n_2}{2} = 1$. Since $\binom{1}{1} = 1$ we conclude that $n_2 = 13$.

Hence the decoded string has 1s in positions 9, 12, and 13, and thus the first block decodes to 00000000100110.