Solution 1.
a) r is the smallest integer such that $2^r \geq n + r + 1$. b) r = the smallest integer such that $2^r \geq 10 + r + 1$ is 4 because $2^4 > 10 + 1 + 4 = 15$ but $2^3 < 10 + 1 + 3 = 14$. c) n is the largest integer $n \leq 2^r - r - 1$. d) $2^3 \geq 3 + 1 + n$, so $n \leq 8 - 4 = 4$.

Solution 2:.
a) The assignment is a code because different letters have been assigned different codewords.

b) A prefix code is a code in which no codeword is a proper prefix of another codeword.

c) We use Kraft's theorem to decide wether there is a prefix binary code for the given codewords lengths.

i) Because $2^{-2} + 2^{-1} + 2^{-2} + 2^{-133333} = 1 + 2^{-133333} > 1$ by Kraft's theorem there is no prefix binary code for which the codewords lengths are exactly: 2, 1, 2, 133333.

ii) Because $2^{-3} + 2^{-3} + 2^{-3} + 2^{-3} + 2^{-3} + 2^{-3} + 2^{-3} + 2^{-3} = 1$ by Kraft's theorem there is a prefix code whose codewords lengths are all 3: 000, 001, 010, 011, 100, 101, 110, 111. This code is unique.

iii) Because $2^{-3} + 2^{-5} + 2^4 + 2^{-1} < 1$ by Kraft's theorem there is a prefix code whose codewords lengths are 3, 5, 4, 1, for example, 100, 10110, 1010, 0. This code is not unique; for example changing 0 to 1 and 1 to 0 in this code we obtain another code satisfying all requirements.

Solution 3.
Write the letters in increasing order of their frequencies, say C, D, A, B, E, and use Huffman's procedure:

C (10) D (10) A (15) B (15) E (50)
CD (20) AB (30) E (50)
CDAB (50) E (50)
CDABE (100)

so the code is

Letter	Code
A	101
B	100
C	111
D	110
E	0

If we write the letters in increasing order of their frequencies as D, C, A, B, E, and use Huffman's procedure:

D (10) C (10) A (15) B (15) E (50)
DC(20) AB (30) E (50)
DCAB (50) E (50)
DCABE (100)

so the code is

Letter	Code
A	101
B	100
C	110
D	111
E	0

Solution 4.
1) If the file has n characters each stored as a 7-bit code you need $7n$ bits.

2) The file uses $26 + 10 + 10 = 46$ characters, so with a 6-bit code one can code $2^6 = 64 > 46$ characters.

3) A file with n characters will be coded by $6n$ bits, so the size reduction is (from $7n$ to $6n$) is about 14%.

Solution 5.
1) In a two-dimensional parity check a block of bits is organised in a table and parity is checked on both dimensions. First one calculates the parity bit for each data unit. Second one calculates the parity bit for each column and one creates a new row of bits – the parity bits for the whole block.

The two-dimensional parity is calculated as follows (see the added bits in bold):

0	1	0	0	0	0	0	0	**1**
0	0	0	0	1	1	1	0	**1**
0	0	0	0	1	0	1	0	**0**
0	1	1	0	0	0	0	0	**0**
1	1	1	1	1	1	0	0	**0**
1	1	1	1	1	1	0	1	**1**
1	1	1	1	1	1	1	0	**1**
1	1	1	1	0	0	0	0	**0**
0	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**0**

so the data plus parity bits transmitted are:

$$010000000\mathbf{1}, 000011100\mathbf{1}, 000010100\mathbf{0}, 011000000\mathbf{0}, 111111000\mathbf{0},$$
$$111111010\mathbf{1}, 111111100\mathbf{1}, 111100000\mathbf{0}, \mathbf{001010110}$$

3) Instead of 64 bits one sends 64 + 8 + 9 = 81 bits, an increase of about 21%.