ASCII Character Representation

0 NUL	16 DLE	32	48 0	64 @	80 P	96 `	112 p
1 SOH	17 DC1	33 !	49 1	65 A	81 Q	97 a	113 q
2 STX	18 DC2	34 "	50 2	66 B	82 R	98 b	114 r
3 ETX	19 DC3	35 #	51 3	67 C	83 S	99 c	115 s
4 EOT	20 DC4	36 $	52 4	68 D	84 T	100 d	116 t
5 ENQ	21 NAK	37 %	53 5	69 E	85 U	101 e	117 u
6 ACK	22 SYN	38 &	54 6	70 F	86 V	102 f	118 v
7 BEL	23 ETB	39 '	55 7	71 G	87 W	103 g	119 w
8 BS	24 CAN	40 (	56 8	72 H	88 X	104 h	120 x
9 HT	25 EM	41)	57 9	73 I	89 Y	105 i	121 y
10 LF	26 SUB	42 *	58 :	74 J	90 Z	106 j	122 z
11 VT	27 ESC	43 +	59 ;	75 K	91 [	107 k	123 {
12 FF	28 FS	44 ,	60 <	76 L	92 \	108 l	124 |
13 CR	29 GS	45 -	61 =	77 M	93]	109 m	125 }
14 SO	30 RS	46 .	62 >	78 N	94 ^	110 n	126 ~
15 SI	31 US	47 /	63 ?	79 O	95 _	111 o	127 DEL

Abbreviations

NUL	Null	DLE	Data link escape
SOH	Start of heading	DC1	Device control 1
STX	Start of text	DC2	Device control 2
ETX	End of text	DC3	Device control 3
EOT	End of transmission	DC4	Device control 4
ENQ	Enquiry	NAK	Negative acknowledge
ACK	Acknowledge	SYN	Synchronous idle
BEL	Bell (beep)	ETB	End of transmission block
BS	Backspace	CAN	Cancel
HT	Horizontal tab	EM	End of medium
LF	Line feed, new line	SUB	Substitute
VT	Vertical tab	ESC	Escape
FF	Form feed, new page	FS	File separator
CR	Carriage return	GS	Group separator
SO	Shift out	RS	Record separator
SI	Shift in	US	Unit separator
		DEL	Delete/Idle

1) The ASCII code for character 'A' is 65_{10}, 'B' is 66_{10}, ... and 'a' is 97_{10}, 'b' is 98_{10}, ...

a) What would be the 7-bit binary value used to represent 'A'?

b) What would be the 7-bit binary value used to represent 'a'?

c) How does an upper-case letter differ from its corresponding lower-case letter?

d) *Even parity* prepends a 0 or 1 so as to make the total number of 1's be even. What is the 8-bit ASCII value for"

'A':

'a':

e) What error(s) cannot be detected by even parity?

2 a) For the 8-bit data 01001011_2 develop the Hamming codeword for one-bit error detection and correction:

12	11	10	9	8	7	6	5	4	3	2	1
D_7	D_6	D_5	D_4	P_8	D_3	D_2	D_1	P_4	D_0	P_2	P_1
0	1	0	0		1	0	1		1		
4+8	1+2+8	2+8	1+8	8	1+2+4	2+4	1+4	4	1+2	2	1

Check bit P_1 looks at bit positions 1, 3, 5, 7, 9, and 11
Check bit P_2 looks at bit positions 2, 3, 6, 7, 10, and 11
Check bit P_4 looks at bit positions 4, 5, 6, 7, and 12
Check bit P_8 looks at bit positions 8, 9, 10, 11, and 12

b) If bit D_5 gets flipped (an error), then how would we be able to detect an error?

c) If bit D_5 gets flipped (an error), then how would we be able to know which bit to correct?

d) For the 8-bit data 11001001_2 develop the Hamming codeword for one-bit error detection and correction:

12	11	10	9	8	7	6	5	4	3	2	1
D_7	D_6	D_5	D_4	P_8	D_3	D_2	D_1	P_4	D_0	P_2	P_1
1	1	0	0		1	0	0		1		
4+8	1+2+8	2+8	1+8	8	1+2+4	2+4	1+4	4	1+2	2	1