Patipol Saechan
60130500219

1. Assume we are using the simple model for floating-point representation as given in this book (the representation uses a 14-bit format, 5 bits for the exponent with a bias of 15, a normalized mantissa of 8 bits, and a single sign bit for the number):

a) Show how the computer would represent the numbers 100.0 and 0.25 using this floating-point format.

100.0 = 64 + 32 + 4

= $0.1100100 * 2^7$

= 0 10110 11001000

0.25 = $\frac{1}{4}$

= $0.1*2^{-1}$

= 0 01110 10000000

b) Show how the computer would add the two floating-point numbers in part a by changing one of the numbers so they are both expressed using the same power of 2.

0 10110 11001000

0 10110 000000001 +

0 10110 11001000~~1~~

c) Show how the computer would represent the sum in part b using the given floating-point representation. What decimal value for the sum is the computer actually storing? Explain.

= 0 10110 11001000

= +/- Expo significand

= + 7 0.11001000

= $0.1100100 * 2^7$

= 11001_2

= 100

---

2. Show how each of the following floating point values would be stored using IEEE-754 single precision (be sure to indicate the sign bit, the exponent, and the significand fields):

a) 12.5 = $8 + 4 + \frac{1}{2}$

= $1100.1*2^0$

= $1.1001*2^3$

= 0 10000100 10010000000000000000000

b) −1.5 = $1+\frac{1}{2}$

= $1.1*2^{0}$

= 1 1000000 10000000000000000000000

c) 0.75 = $\frac{1}{2}+\frac{1}{4}$

= $0.11*2^{0}$

= $1.1*2^{-1}$

= 0 01111111 10000000000000000000000

d) 26.625 = $16+8+2+\frac{1}{2}+\frac{1}{8}$

= $11010.101*2^{0}$

= $1.1010101*2^{4}$

= 0 10000101 10101010000000000000000

3.

a) The ASCII code for the letter A is 1000001, and the ASCII code for the letter a is 1100001. Given that the ASCII code for the letter G is 1000111, with no help from any lookup table, what is the ASCII code for the letter g?

1100001

1000001 −

0100000

1000111

0100000 +

1100111

b) Given that the ASCII code for the letter Q is 1010001, with no help from any lookup table, what is the ASCII code for the letter q?

1010001

0100000 +

1110001

c) In general, if you were going to write a program to convert uppercase ASCII characters to lowercase, how would you do it?

ASCII of uppercase plus 32

---

4. Represent the following decimal numbers in Excess-127 representation:

a) 77 = 64+8+4+2+1

= $1001111*2^0$

= $1.001111*2^6$

= 0 10000110 00111100000000000000000

b) −42 = 32+8+2

= $101010*2^0$

= $1.0101*2^5$

= 1 10000101 01010000000000000000000

5.

a) Convert decimal 9126 to both BCD and ASCII codes. For ASCII, an odd parity bit is to be appended at the left.

1 2 3 4 5 6 7 8 9

0001 0010 0011 0100 0101 0110 0111 1000 1001

BCD 9126 = 1001 0001 0010 0110

= 1001000100100110

ASCII 9126 = 57 49 50 54

= 10111001001100010011001010110110

b) Write the expression “G. Boole” in ASCII using an eight-bit code. Include the period and the space. Treat the leftmost bit of each character as a parity bit. Each 8-bit code should have even parity.

G . space B o l e

01000111 00101110 10100000 01000010 01101111 01101100 01100101

“G. Boole” = 0100011100101110101000000100001001101111011011110110110001100101

Patipol Saechan
60130500219

6. Assume we wish to create a code using 3 information bits, 1 parity bit (appended to the end of the information), and odd parity. List all legal code words in this code.

_ _ _ x

2*2*2*1 = 8

0001 0010 0100 0111 1000 1011 1101 1110

---

7. Suppose we want an error-correcting code that will allow all single-bit errors to be corrected for memory words of length 10.

a) How many parity bits are necessary?

$2^k-1=m+k$

$2^k-1 \geq 10+k$; m=10

K=4

b) Assuming we are using the Hamming algorithm presented in this chapter to design our errorcorrecting code, find the code word to represent the 10-bit information word: 1001100110.

	1	0	0	1	1	0	0	x	1	1	0	x	0	x	x	
P_1	✓		✓		✓		✓		✓		✓		✓		✗	0
P_2		✓	✓		✓	✓			✓	✓			✓	✗		0
P_3	✓	✓	✓	✓					✓	✓	✓	✗				0
P_4	✓	✓	✓	✓	✓	✓	✓	✗								1

Hamming code is 1000110011100000

Patipol Saechan
60130500219

8. Suppose we are working with an error-correcting code that will allow all single-bit errors to be corrected for memory words of length 7. We have already calculated that we need 4 check bits, and the length of all code words will be 11. Code words are created according to the Hamming Algorithm presented in the text. We now receive the following code word: 1 0 1 0 1 0 1 1 1 1 0

Assuming even parity, is this a legal code word? If not, according to our error-correcting code, where is the error?

	1	0	1	0	1	0	1	1	1	1	0	
P_1	✓		✓		✓		✓		✓		✓	1
P_2	✓	✓			✓	✓			✓	✓		0
P_3					✓	✓	✓	✓				1
P_4	✓	✓	✓	✓								0

This code isn't have even parity bit.

Error is bit number 0101= 5

---

9. Using the CRC polynomial 1011, compute the CRC code word for the information word, 1011001.

$$
\begin{array}{r}
1000000 \\
1011\overline{)1011001000} \\
\underline{1011} \\
0000001000 \\
\underline{0000001011} \\
0000000011
\end{array}
$$

CRC code is 1011001000+0011= 1011001011

Patipol Saechan
60130500219

10. Using the CRC polynomial 1101, compute the CRC code word for the information word, 01011101. Check the division performed at the receiver.

```
        01100001
1101)0101110100
     0000
     01011
     01101
     001101
     001101
     0000001000
     0000001101
     0000000101
```

CRC code is 01011101000+101= 01011101101

```
        01100001
1101)0101110110
     0000
     01011
     01101
     001101
     001101
     0000001101
     0000001101
     0000000000
```

No error