Solution I

1-1 Decimal, Binary, Octal and Hexadecimal Numbers from $(16)_{10}$ to $(31)_{10}$

Dec	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Bin	10000	10001	10010	10011	10100	10101	10110	10111	11000	11001	11010	11011	11100	11101	11110	11111
Oct	20	21	22	23	24	25	26	27	30	31	32	33	34	35	36	37
Hex	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F

1-2

$12K = 12 \times 2^{10} = 12{,}288$ Bits
$128M = 128 \times 2^{20} = 134{,}217{,}728$ Bits
$2G = 2 \times 2^{30} = 2{,}147{,}483{,}648$ Bits

1-3

8 Bits $\Rightarrow 2^8 - 1 = 255$
32 Bits $\Rightarrow 2^{32} - 1 = 4{,}294{,}967{,}295$

1-4

$(111010)_2 = 2^1 + 2^3 + 2^4 + 2^5 = 58$
$(10101111.101)_2 = 2^7 + 2^5 + 2^3 + 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-3} = 175.625$
$(110110110)_2 = 2^8 + 2^7 + 2^5 + 2^4 + 2^2 + 2^1 = 438$

1-5

1940 = 111 1001 0100
138 = 1000 1010
1056 = 100 0010 0000
1995 = 111 1100 1011

1-6

$(12021)_3 = 141$
$(4321)_5 = 586$
$(A98)_{12} = 1556$

1-7

Decimal	Binary	Octal	Hexadecimal
369.3125	101110001.0101	561.24	171.5
189.625	10111101.101	275.5	BD.A
214.625	11010110.101	326.5	D6.A
62407.625	1111001111000111.101	171707.5	F3C7.A

1-8

$(715 + 367 = 1304)_8$
$(715 - 367 = 326)_8$
$(715 \times 367 = 336313)_8$

$(15F + A7 = 206)_{16}$
$(15F - A7 = B8)_{16}$
$(15F \times A7 = E4F9)_{16}$

$(110101 + 110110 = 1101011)_2$
$(110101 - 110110 = -1)_2$
$(110101 \times 110110 = 101100101110)_2$

1-9

$7562.45 = (16612.3463)_8$
$1938.257 = (792.41CA)_{16}$
$175.175 = (10101111.001011)_2$

1-10 For the first method, there will be $2 \times \lceil \log_2 N \rceil$ additions and multiplication.
The number of operations in the above algorithm is determined by the number of bits the binary representation of N occupies. If N requires n bits in binary, $n - 1$ additions and $n - 1$ multiplication are required. With N in decimal $n = \lceil \log_2 N \rceil$.
Look at the following example:
$N = 6 = (110)_2$ N occupies 3 bits.
$\log_2 6 = 2.585$ $\lceil \log_2 6 \rceil = 3$
Number of additions + Number of Multiplication = 4
In general, for any number with $n = 3$ the following expression can be applied:
$((A_2 \times r) + A_1) \times r + A_0$
The expression clearly indicates that two multiplication and two additions, $n - 1$ and $n - 1$, respectively, are needed.
In the method described in the book the number of additions remains $n - 1$, however, the number of multiplication is significantly larger. This is best shown by example:
$A_2 \times r^2 + A_1 \times r + A_0 = A_2 \times r \times r + A_1 \times r + A_0$
The number of multiplies is thus equal to 2 + 1 + 0, for any n, this sum will be equal to $n (n - 1) / 2$
Number of additions + Number of Multiplication = $(n - 1) + n (n - 1) / 2 = (n - 1) (n + 2) / 2$
It can be seen that the first algorithm requires fewer operations for n greater than or equal to 3.

1-11

$(764.7)_8 = (111\ 110\ 100.111)_2 = (1F4.E)_{16}$
$(F6D.C)_{16} = (1111\ 0110\ 1101.1100)_2 = (7555.6)_8$
$(147.5)_8 = (001\ 100\ 111.101)_2 = (1213.22)_4$

1-18

$(0011\ 1001\ 0111\ 1000)_{BCD} = 3978 = (111110001010)_2$
$(0100\ 0110\ 1001.0111\ 0101)_{BCD} = 469.75 = (111010101.11)_2$

1-19

John B. Doe

0100 1010	0110 1111	0110 1000	0110 1110	0010 0000
0100 0010	0010 1110	0010 0000	0100 0100	0110 1111
0110 0101				

1100 1010	0110 1111	1110 1000	1110 1110	1010 0000
0100 0010	0010 1110	1010 0000	0100 0100	0110 1111
0110 0101				

1-20

John Doe

1-21

100100111
0010 1001 0101

011 0010	011 1001	011 0101
$(0032)_{16}$	$(0039)_{16}$	$(0035)_{16}$

1-22

2^{32} = 4,294,967,296 Integers
32 Bits × (1 Digit) / (4 Bits) = 8 Digits = 10^8 = 100,000,000 Integers
32 Bits × (1 Digit) / (8 Bits) = 4 Digits = 10^4 = 10,000 Integers

1-23 BCD Digits with Odd and Even Parity

	0	1	2	3	4	5	6	7	8	9
Odd	1 0000	0 0001	0 0010	1 0011	0 0100	1 0101	1 0110	0 0111	0 1000	1 1001
Even	0 0000	1 0001	1 0010	0 0011	1 0100	0 0101	0 0110	1 0111	1 1000	0 1001

1-24

Answer: B_6

1-25

John B. Doe
FEFF 004A 006F 0068 006E 0020 0042 002E 0020 0044 006F 0065

or

FFFE 4A00 6F00 6800 6E00 2000 4200 2E00 2000 4400 6F00 6500

1-26

1 ÷ 4 = ¼ is a FRACTION.