Solutions to Exercises from Chapter 01

Contents

1 Exercises on the computer representation of numbers

1.1 Exercise 01

Transform the following numbers from base 2 to base 10:

1. 1000.101
2. 111.11

SOLUTION

The bits of a binary number (base 2) multiply powers of 2. Thus:

1. $1000.101 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 8 + 0.5 + 0.125 = 8.625$

2. $111.11 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 4 + 2 + 1 + 0.5 + 0.25 = 7.75$

1.2 Exercise 02

Transform the following numbers from base 10 to base 2:

1. 32.3125
2. 5.375

SOLUTION

We can divide each number into its integer and decimal part. For the integer part we keep dividing by two and consider the remainder; for the decimal part, we multiply by 2 and use 0 if the result is smaller than 1 and 1 if it is greater or equal than 1.

1. Let us focus on the integer part, first. Using division by 2 consecutively we obtain:

$$\begin{array}{rcrcrl} 32 & \div & 2 & = & 16 & \text{remainder } 0 \\ 16 & \div & 2 & = & 8 & \text{remainder } 0 \\ 8 & \div & 2 & = & 4 & \text{remainder } 0 \\ 4 & \div & 2 & = & 2 & \text{remainder } 0 \\ 2 & \div & 2 & = & 1 & \text{remainder } 0 \\ 1 & \div & 2 & = & 0 & \text{remainder } 1 \\ 0 & & & & & \text{STOP} \end{array}$$

The integer part of the number, 32, is represented by 100000 in base 2. Next, let us consider 0.3125 (the decimal part of the number), and follow a series of multiplications by 2:

$$\begin{array}{rcrcll} 0.3125 & \times & 2 & = & 0.625 & \text{take integer : } 0 \quad \text{value for next line : } 0.625 \\ 0.625 & \times & 2 & = & 1.25 & \text{take integer : } 1 \quad \text{value for next line : } 0.25 \\ 0.25 & \times & 2 & = & 0.5 & \text{take integer : } 0 \quad \text{value for next line : } 0.5 \\ 0.5 & \times & 2 & = & 1 & \text{take integer : } 1 \quad \text{value for next line : STOP} \end{array}$$

The decimal part is therefore represented by 0101. In conclusion:

$$32.3125 \ \rightarrow \ 100000.0101.$$

2. Let us focus on the integer part, first. Using division by 2 consecutively we obtain:

$$\begin{array}{rcrcrl} 5 & \div & 2 & = & 2 & \text{remainder } 1 \\ 2 & \div & 2 & = & 1 & \text{remainder } 0 \\ 1 & \div & 2 & = & 0 & \text{remainder } 1 \\ 0 & & & & & \text{STOP} \end{array}$$

The integer part of the number, 5, is represented by 101 in base 2. Next, let us consider 0.375 (the decimal part of the number), and follow a series of multiplications by 2:

$$\begin{array}{rcrcll} 0.375 & \times & 2 & = & 0.75 & \text{take integer : } 0 \quad \text{value for next line : } 0.75 \\ 0.75 & \times & 2 & = & 1.5 & \text{take integer : } 1 \quad \text{value for next line : } 0.5 \\ 0.5 & \times & 2 & = & 1 & \text{take integer : } 1 \quad \text{value for next line : STOP} \end{array}$$

The decimal part is therefore represented by 011. In conclusion:

$$5.375 \ \rightarrow \ 101.011.$$

1.3 Exercise 03

Find the base 10 expression of the three consecutive numbers, $01101001, 01101010, 01101011$, written in the toy 8-bit IEEE system.

SOLUTION

Let us start from the first number. The initial 0 means that the number is positive. The next three bits define the exponent. 110 is the integer 6. Considering the offset 3, this corresponds to $2^{6-3} = 2^3$. The following four bits have to be interpreted as values following 1 and the floating point:

$$1001 \ \rightarrow \ 1.1001.$$

This number is multiplied by 2^3. Therefore:

$$01101001 \ \rightarrow \ 1.1001 \times 2^3 \ \rightarrow \ 1100.1 \ \rightarrow \ 12.5.$$

Thus, 01101001 corresponds to 12.5. The next, adjacent number is 01101010, and we have:

$$01101010 \ \rightarrow \ 1.1010 \times 2^3 \ \rightarrow \ 1101.0 \ \rightarrow \ 13.$$

Finally, for the next (and last) adjacent number we have:

$$01101011 \ \rightarrow \ 1.1011 \times 2^3 \ \rightarrow \ 1101.1 \ \rightarrow \ 13.5.$$

1.4 Exercise 04

The smallest normal positive number in the IEEE 8-bit toy system is 0.25. It is possible to represent exactly a handful of numbers smaller than 0.25, but greater than 0 within the IEEE system, the so-called subnormal numbers (see main text). List all 15 subnormal numbers representable with the IEEE 8-bit system.

SOLUTION

Subnormal numbers are of the form

$$0\ 000\ xxxx,$$

where the four bits in the significand cannot all be zero at the same time (this situation is identified as the number 0). These values are then multiplied by the smallest power of 2 usabe, which is 2^{-2}. We have, thus

$$\begin{array}{llll} 0 & 000 & 0001 & \rightarrow\ 0.015625 \\ 0 & 000 & 0010 & \rightarrow\ 0.03125 \\ 0 & 000 & 0011 & \rightarrow\ 0.046875 \\ 0 & 000 & 0100 & \rightarrow\ 0.0625 \\ 0 & 000 & 0101 & \rightarrow\ 0.078125 \\ 0 & 000 & 0110 & \rightarrow\ 0.09375 \\ 0 & 000 & 0111 & \rightarrow\ 0.109375 \\ 0 & 000 & 1000 & \rightarrow\ 0.125 \\ 0 & 000 & 1001 & \rightarrow\ 0.140625 \\ 0 & 000 & 1010 & \rightarrow\ 0.15625 \\ 0 & 000 & 1011 & \rightarrow\ 0.171875 \\ 0 & 000 & 1100 & \rightarrow\ 0.1875 \\ 0 & 000 & 1101 & \rightarrow\ 0.203125 \\ 0 & 000 & 1110 & \rightarrow\ 0.21875 \\ 0 & 000 & 1111 & \rightarrow\ 0.234375 \end{array}$$

There are, indeed, 15 subnormal numbers.

1.5 Exercise 05

Calculate the smallest positive normal number and the largest positive subnormal number in the IEEE 32-bit system.

SOLUTION

The smallest positive number is given by the formula

$$\beta^{e_{\min}+1},$$

where $\beta = 2$ and $[e_{\min}, e_{\max}] = [-127, 128]$ for the IEEE 32-bit system. The smallest positive normal number is thus

$$2^{-127+1} = 2^{-126} \approx 1.175494350 \times 10^{-38}.$$

To calculate the largest positive subnormal number, we could proceed similarly to what done with the IEEE 8-bit toy system (see previous exercise). But this would absorb too much time. Rather, and more conveniently, we can simply subtract an appropriate power of 2 from the smallest normal number, and obtain the same result. Such a value is

$$2^{e_{\min}+1} \times 2^{-m},$$

where m is the number of bits in the significand. For the 32-bit system we have $e_{\min} = -127$ and $m = 23$ and therefore the appropriate power of 2 is

$$2^{-126} \times 2^{-23} = 2^{-149} \approx 1.401298464324817 \times 10^{-45}.$$

This number is, as expected, very small and therefore the largest subnormal number is very close to the largest normal number. This is even more the case for the IEEE 64-bit system.

1.6 Exercise 06

Verify that there are $n = 2^m$ normal numbers between any two consecutive powers of 2, in the IEEE system, where the two consecutive numbers are represented as

$$2^p, 2^{p+1}, \qquad e_{\min+1} \leq p \leq e_{\max} - 1.$$

In particular, find out how many normal numbers exist between 2 and 4 in the IEEE 32-bit and 64-bit systems.

SOLUTION

The number of normal numbers between any two consecutive powers of 2 is determined by the number of bits in the significand, and it is independent from the specific value of p. This is due to the fact that each bit of the significand can only take two values, 0 and 1. Therefore, between any two consecutive powers of 2 there will be 2^m normal numbers. For the 32-bit system there will be $2^{23} = 8388608$ (millions) numbers, while for the 64-bit system, where $m = 52$, there will be $2^{52} = 4503599627370496$ (million of billions) numbers. In both cases we are excluding the largest power of 2, i.e. 2^{p+1}.

1.7 Exercise 07

Represent $\sqrt{2}$ and $\sqrt{3}$ in the IEEE 8-bit toy system. For both numbers, calculate the absolute and relative errors, due to round off.

SOLUTION

Let us start with $\sqrt{2} \approx 1.414213562$. First, the number is transformed into a binary number. The integer is 1, and this corresponds to 1 in base 2 too. For the decimal part we have:

0.4144213562	×	2	=	0.8288427124	take integer :	0	value for next line :	0.8288427124
0.8288427124	×	2	=	1.6576854248	take integer :	1	value for next line :	0.6576854248
0.6576854248	×	2	=	1.3153708496	take integer :	1	value for next line :	0.3153708496
0.3153708496	×	2	=	0.6307416992	take integer :	0	value for next line :	0.6307416992
0.6307416992	×	2	=	1.2614833984	take integer :	1	value for next line :	0.2614833984
0.2614833984	×	2	=	0.5229667968	take integer :	0	value for next line :	0.5229667968
0.5229667968	×	2	=	1.0459335936	take integer :	1	value for next line :	0.0459335936

which yields

$$\sqrt{2} \approx 1.0110101.$$

The algorithm for conversionn does not stop in this case because an irrational number has an infinite number of digits even in base 10. It is important, though, to go beyond the number m of the significand, in order to apply the rules for rounding correctly. The fourth bit after the floating point is rounded, in this case, to 1. The result is the following number in the 8-bit representation:

$$\sqrt{2} \ \rightarrow \ 1.0111 \times 2^0 \ \rightarrow \ 0\ 011\ 0111 = 1.4375$$

Therefore, $\sqrt{2}$ is represented by 1.4375, in the 8-bit system. The absolute error due to this representation (round off error) is

$$E_a = |x - \tilde{x}| \approx |1.4144 - 1.4375| = 0.0231.$$

The relative error is

$$E_r = |x - \tilde{x}|/|x| \approx 0.0231/1.4144 \approx 0.0163.$$

This is, as expected, smaller than the machine precision ϵ_{mach} which, for the 8-bit system, is $2^{-m} = 2^{-4} = 0.0625$.

1.8 Exercise 08

Find the sum and difference of the two base 2 numbers, $110.011, 110.010$, once they have been represented in the IEEE 8-bit toy system. Work out the absolute and relative errors of both the addition and subtraction.

SOLUTION

The two numbers are

$$110.011 = 6.375, \quad 110.010 = 6.25.$$

Therefore, the correct results for sum and difference are

$$110.011 + 110.010 = 6.375 + 6.25 = 12.625, \quad 110.011 - 110.010 = 6.375 - 6.25 = 0.125.$$

These operations can be carried out without passing through the base 10 system, and we will proceed this way after having transformed preliminarly the numbers in the 8-bit system.

The first number cannot be represented exactly:

$$110.011 \ \rightarrow \ 1.1010 \times 2^2 \quad \text{corresponding to } 6.5$$

The second number can be represented exactly:

$$110.010 \ \rightarrow \ 1.1001 \times 2^2 \quad \text{corresponding to } 6.25.$$

We expect, therefore, the sum to give 12.75 and the difference to give 0.25. Indeed,

$$\begin{array}{rcll} 1.1010 & \times & 2^2 & + \\ 1.1001 & \times & 2^2 & = \\ \hline 11.0011 & \times & 2^2. & \end{array}$$

This number is equal to 1100.11, which is 12.75 in base 10, as expected. For the difference we have:

$$\begin{array}{rcll} 1.1010 & \times & 2^2 & - \\ 1.1001 & \times & 2^2 & = \\ \hline 0.0001 & \times & 2^2. & \end{array}$$

This number is equal to 0.01, which is 0.25 in base 10, also as expected.

The absolute errors for both results are:

$$E_a = |12.625 - 12.75| = 0.125, \quad E_a = |0.125 - 0.25| = 0.125.$$

The corresponding relative errors are:

$$E_r = E_a/|12.625| \approx 0.00990, \quad E_r = E_a/|0.125| = 1.$$

The relative error is significantly high for subtractions (here 100%), when the two numbers are similar in magnitude, as it is the case in this exercise.

1.9 Exercise 09

Calculate the absolute and relative error when $\sin(x)$ is replaced by the truncated expansion

$$\sin(x) \approx x - x^3/6,$$

for $x = 0.1$ and $x = \pi/6$ (all values in radians).

SOLUTION

If $y = \sin(x)$ and $\tilde{y} = x - x^3/6$, the absolute and relative errors are

$$E_a = |y - \tilde{y}| = |\sin(x) - x + x^3/6|, \quad E_r = E_a/|y| = |\sin(x) - x + x^3/6|/|\sin(x)|.$$

For $x = 0.1$ the values are

$$E_a = |\sin(0.1) - 0.1 + 0.1^3/6| \approx 0.000000083$$

and

$$E_r = E_a/|\sin(0.1)| = 0.000000835$$

Such a small value for E_r means that calculations with $x - x^3/3$ replacing $\sin(x)$ will be accurate when x is close to 0.1 radians. For $x = \pi/6$ (corresponding to 30 degrees), the truncation does not yield a similar precision, but it is still advantageous:

$$E_a = |\sin(\pi/6) - \pi/6 + (\pi/6)^3/6| \approx 0.000325821$$

and

$$E_r = E_a/|\sin(\pi/6)| = E_a/0.5 \approx 0.000651641$$