Exercise 6: Basics of computer science, part I (Solution)

Data and Digital Storage

The logic and mathematics of the digital world are essential tools in computer science. Since in the most basic way computers work with binary digits (0s and 1s), a basic grasp of data representation and arithmetics in binary base in extremely important. In order to thoroughly understand binary bases, an understanding of general number bases and arithmetics using different number bases is needed.

Problem 1: Data Representation

1. Why do we use base-2 for data storage in computers?

> **Answer:**
> Computers are electronic machines, and it is much easier to represent values as ON/OFF signals, especially if low voltages are required. This way keeps the circuits simple and allows for a greater control over possible errors (think about the difference between, i.e., $0, 1, 2, \ldots, 9$ Volts as representing 10 different values, vs. only having two possible values).

2. How many Bits are in a Byte? How many in a Word?

> **Answer:**
> Today, the standard is 8 bits per Byte. A word, on the other hand, can have a varying amount of bits, depending on architecture: 8 and 16 bits/word were common in the past, and today 32 and 64 bits/word are the de-facto standards.

3. How many Bytes are in the following: 1KiB, 1MiB, 1GiB?

> **Answer:**
> One kibibyte (tradionally: kilobyte) has 2^{10} bytes, or 1024 bytes. A Mebibyte (tradionally: megabyte) contains 1024 kibibytes, and therefore $1024^2 = 1,048,576$ bytes. A gibibyte (tradionally: gigabyte) is 1024 mebibytes, and therefore $1024^3 = 1,073,741,824$ bytes.

4. What is the minimum amount of bits needed for a complete representation of the English alphabet, which has 26 characters?

> **Answer:**
> The closest power of 2 that is bigger than 26 is 32: $32 = 2^5$. Therefore, at least 5 bits are needed to completely represent all of the English alphabet.

5. A book contains 1337 pages, with 4000 characters per page (including spaces). How much data is stored in the book, if we use the ASCII format to represent all characters (and assuming no compression)?

> **Answer:**
> There are $1337 \times 4000 = 5,348,000$ characters. As ASCII is usually encoded at 8 bits/character, the book would be $8 \times 5,384,000$ bits long, or simply $5,384,000$ Bytes, which are 5.136 MB.

Problem 2: Number Bases

1. Convert the following binary numbers to base-10: 10, 101, 101111.

> **Answer:**
>
> $$\begin{array}{|c|c|}\hline 1 & 0 \\ \hline \multicolumn{1}{c}{2^1} & \multicolumn{1}{c}{2^0}\end{array} = 1 \times 2^1 + 0 \times 2^0 = 2$$

1	0	1
2^2	2^1	2^0

$= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5$

1	0	1	1	1	1
2^5	2^4	2^3	2^2	2^1	2^0

$= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 47$

2. Convert the following base-10 numbers to base-2: 5, 16, 109.

Answer:

$5_{10} \to$ base-2

Result	Quotient	Remainder
5	2	1
2	1	0
1	0	1

Answer: $5_{10} = 101_2$

$16_{10} \to$ base-2

Result	Quotient	Remainder
16	8	0
8	4	0
4	2	0
2	1	0
1	0	1

Answer: $16_{10} = 10000_2$

$109_{10} \to$ base-2

Result	Quotient	Remainder
109	54	1
54	27	0
27	13	1
13	6	1
6	3	0
3	1	1
1	0	1

Answer: $109_{10} = 1101101_2$

3. Convert the same numbers to base-3 and base-16.

Answer:

$5_{10} \to$ base-3

Result	Quotient	Remainder
5	1	2
1	0	1

Answer: $5_{10} = 12_3$

$16_{10} \to$ base-3

Result	Quotient	Remainder
16	5	1
5	1	2
1	0	1

Answer: $16_{10} = 121_3$

$109_{10} \to$ base-3

Result	Quotient	Remainder
109	36	1
36	12	0
12	4	0
4	1	1
1	0	1

Answer: $109_{10} = 11001_3$

4. How many bits are needed to represent two hexadecimal digits? How many different numbers can be represented with two hexadecimal digits?

Answer:
two hexadecimal digits can represent $16^2 = 256$ different values. Since $2^8 = 256$, 8 binary digits are needed to represent 2 hexadecimal digits.

5. Convert the following 4-digit binary numbers to hexadecimal numbers: 0000, 0110, 1100, 1111.

Answer:
$0000_2 = 0_{16}$
$0110_2 = 6_{16}$
$1100_2 = \text{C}_{16}$
$1111 = \text{F}_{16}$

6. Convert the following 2-digit hexadecimal numbers to binary numbers: 04, 1E, A7, FF.

Answer:
Each single digit hexadecimal number has a 4-bit representation:

Hex	Bin	Hex	Bin
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

Therefore, each N-digit hexadecimal number can be converted to a $4N$-bit number by simply converting each digit separately. Thus (notice that the leading zeros are discarded):

$$\begin{aligned}04_{16} &= 0000\ 0100 = 100_2\\ 1\text{E}_{16} &= 0001\ 1110 = 11110_2\\ \text{A}7_{16} &= 1010\ 0111 = 10100111_2\\ \text{FF}_{16} &= 1111\ 1111 = 11111111_2\end{aligned}$$

A common notation for hexadecimal numbers in many programming languages is **0x** preceding the number. For example, the above numbers would be written as **0x04**, **0x1E**, **0xA7** and **0xFF**.

7. YouTube uses Base64 to generate unique ID numbers for each video on their servers [1]. To represent Base64 all 10 digits are used (0-9) plus all lowercase English letters (a-z) plus all uppercase English letters (A-Z) plus two more signs: '-' (minus) and '_' (underscore). Since YouTube uses 11 digits per video ID - how many unique IDs can this method yield?

Answer:
Since each of the 11 digits can have any value out of 64 (important: the order doesn't matter and characters can repeat) - the total number of possible combinations is $64^{11} = 73,786,976,294,838,206,464$.

[1] Watch `www.youtube.com/watch?v=gocwRvLhDf8` for more information. Also note the video's ID and watch more of Tom Scott's videos!