DigiPen
INSTITUTE OF TECHNOLOGY

Chapter 12 | Numbering Systems

Java AP

Encoding is the process of converting or transforming input from one format to another so that it can be correctly interpreted.

English Language

A book written using the English language consists of chapters that consist of paragraphs that consist of statements that consist of words that are constructed using a character or a symbol set. The character or symbol set consist of alphabet letters and some special characters such as the space, exclamation mark, period, comma, etc…

Basically, a language is a code. In other words, using the alphabet and the concept of using a combination of characters to encode a word is the process of coding the words. **For example:** the word "hello" consists of four characters: 'h', 'e', 'l', and 'o'.

In order to converse, read, or write using the English language, once must study the language itself (Vocabulary, grammar etc…).

Morse code

Morse code is a way of communication and information transmission of encoded characters between human beings, by using "dots" and "dashes" to represent letters, numbers, punctuation and special characters.

Example:

This example shows that we can convert numbers and letters to codes and vice versa. By using a series of dots and dashes we made sure that each letter and number has a code different from all the others.

The simplest representation of letters in Morse code is by using a single dot (letter E) or a single dash (letter T). A combination of two dots or dashes allows us to represent four letters; a combination of three dots or dashes gives us eight more letters and so on.

Number of Dots and Dashes	Combinations
1	$2^1 = 2$
2	$2^2 = 4$
3	$2^3 = 8$
4	$2^4 = 16$
:	:

General Formula: Combinations = $2^{\text{number of dots and dashes}}$

Morse code is said to be a binary code because it deals only with dots and dashes only.

Numbers

Just as we needed to encode words, we also need to encode numbers to be able to communicate values.
How old are you?
How tall are you?
How far is it?
Etc…

The answer to the previous questions requires numbers.

Our civilization was based on buying and selling goods, by doing business between cultures. For example a farmer having ten cows he would have represented them with a drawing of a cow followed by 10 scratch marks. A better way to represent this was by introducing the numbering system that is of two types:

1. **Non positional:** The Roman numbering system
2. **Positional:** The Hindu-Arabic numbering system

Non positional numbering system

The Roman Numbering System:

It is still used today as chapter numbers, clocks, dates on monuments, movies copyright notice…

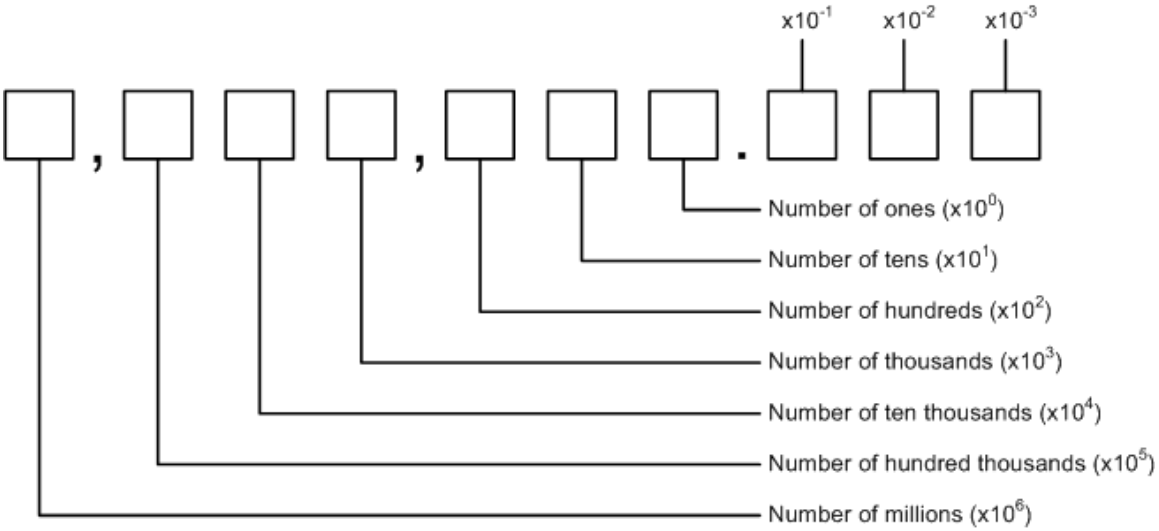
- I = 1 V = 5 X = 10 L=50
  C = 100 (Centum) D = 500 M = 1000 (Mille)

Difficult to **read** yet better than counting sticks or pebbles.
For example, the number III means 3, VI is 6, XXXI is 31, CDXLVIII is 448 and MCMXCVIII is 1998.
Easy to **add (+)**, difficult to **multiply (*) and divide (/).**

Positional numbering system

The Hindu-Arabic Numbering System:

- It was introduced to the world by Arab mathematicians. In particular by Muhammad Ibn-Mussa Al Khawarizmi (of Persian origins), through his book on Algebra (AD 825). A translation to Latin of this book (AD 1120) helped the spread of the Hindu-Arabic numbering system throughout Europe.
- Special unique shapes or symbols: 0 1 2 3 4 5 6 7 8 9
- The Hindu-Arabic number system was different from previous number systems in three ways:
  - **Positional**. A weight is given to each **digit** in the number depending on the **position** of the digit in the number. The digit '1' in the number 130 and the digit '1' in the number 12 have different meaning. In the first it indicates how many **hundreds** are in 130; whereas it indicates how many **tens** are in 12.



  - No symbol for the number 10.
  - Introduced the 'Zero'. The Zero supported the **positional** numbering system.

- Advantages of the Hindu-Arabic system:
  - Can represent any number using only a combination of up to 10 symbols.
  - Easy to **read** any number no matter how complex it is.
  - Easy to **add (+), subtract (-), multiply (*) and divide (/).**

Positional numbering systems are classified according to the total number of unique digits the numbering system uses. In other words, the base or radix of a system indicates how many unique symbols the numbering system uses. For example, Decimal system is a base 10 numbering system because it uses 10 unique symbols.

Please note, that it is very important to distinguish the difference between a digit and a number. The digit has no value, while the number has a value. For example, ‘12’ as a number has a value of 12 units, while it consists of two digits ‘1’ and ‘2’ respectively. Consequently, in base 10, a digit is a character between 0 and 9.

Decimal numbering system

As human beings, we intuitively count to ten because we have ten fingers and ten toes. Because of this, our numbering system is based on the number 10; it's a **base 10** numbering system.

Suppose we have a number 1384.29, we can write its components like this:

$$1384.29 = 1 * 10^3 + 3 * 10^2 + 8 * 10^1 + 4 * 10^0 + 2 * 10^{-1} + 9 * 10^{-2}$$

- Any positive number can be represented as a sum of powers of 10. Note that any number to the power 0 is equal to 1.

- Operations (+, -, / and *) done on the decimal numbering system somehow follow a procedure, for example the addition table:

+	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10	11
3	3	4	5	6	7	8	9	10	11	12
4	4	5	6	7	8	9	10	11	12	13
5	5	6	7	8	9	10	11	12	13	14
6	6	7	8	9	10	11	12	13	14	15
7	7	8	9	10	11	12	13	14	15	16
8	8	9	10	11	12	13	14	15	16	17
9	9	10	11	12	13	14	15	16	17	18

Table 1: Addition in decimal

The addition table showed us how easy and simple is the addition for decimal numbers especially that it follows a pattern. As for the multiplication table:

*	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9
2	0	2	4	6	8	10	12	14	16	18
3	0	3	6	9	12	15	18	21	24	27
4	0	4	8	12	16	20	24	28	32	36
5	0	5	10	15	20	25	30	35	40	45
6	0	6	12	18	24	30	36	42	48	54
7	0	7	14	21	28	35	42	49	56	63
8	0	8	16	24	32	40	48	56	64	72
9	0	9	18	27	36	45	54	63	72	81

Table 2: Multiplication in decimal

This numbering system is more natural for humans that any other numbering system but what about other beings that have less than ten fingers, birds for example have eight fingers. Since they have eight fingers they won't need to count to ten so their numbering system is called octal numbering system or **base 8**.

Other positional numbering systems

Octal numbering system

- The octal numbering system consists of eight digits, so both digits 8 and 9 are not used in this numbering system.
- The numbers are: 0, 1, 2, 3, 4, 5, 6, 7.
- Number 10 in octal is the number that is after 7. If we continue counting we will have 11_8, 12_8, 13_8, 14_8, 15_8, 16_8, 17_8, 20_8, 21_8…
- To differentiate between decimal, octal and other numbering systems we need to use the subscript 10 for the decimal numbers (1234_{10}), the subscript 8 for the octal numbers (1234_8) and so on…
- Suppose we have a number 3725, we can write its components like this:

$$3725 = 3 * 8^3 + 7 * 8^2 + 2 * 8^1 + 5 * 8^0$$

- We can add and multiply octal numbers the same way we add and multiply decimal numbers. For example:

$$\begin{array}{r} 5_8 \\ +\ 7_8 \\ \hline 14_8 \end{array} \qquad \begin{array}{r} 135_8 \\ +\ 643_8 \\ \hline 1000_8 \end{array}$$

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Table 3: Addition in octal

*	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

Table 4: Multiplication in octal

How about a numbering system that is higher than the decimal, does it exist?

Hexadecimal numbering system

- The hexadecimal numbering system consists of 16 symbols (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F); therefore it is also called base 16.

- Any positive number can be represented as a sum of powers of 16, for example:

$$1A5_{16} = 1 * 16^2 + A * 16^1 + 5 * 16^0$$

- The first 20 Hexadecimal numbers starting from 0 are:

- 0_{16}, 1_{16}, 2_{16}, 3_{16}, 4_{16}, 5_{16}, 6_{16}, 7_{16}, 8_{16}, 9_{16}, A_{16}, B_{16}, C_{16}, D_{16}, E_{16}, F_{16}, 10_{16}, 11_{16}, 12_{16}, and 13_{16}.

- This table shows the meaning of the hexadecimal numbering system in the decimal one.

Hex	Dec
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

- Operations on hexadecimal numbers are the same as the operations on decimal numbers including A, B, C, D, E and F digits, for example:

$$\begin{array}{r} 7_{16} \\ +\ 8_{16} \\ \hline F_{16} \end{array} \qquad \begin{array}{r} 1_{16} \\ +\ A_{16} \\ \hline B_{16} \end{array}$$

All possible positional numbering systems

Following the same concept as the previous numbering systems, we can go to higher bases 12, 16, 32, 64… or even go to lower basis like 4, 2. However the lowest we can go is the binary numbering system that is represented with only 2 digits.

Binary numbering system

Binary digital systems form the basis of just about all hardware systems in existence today.
The binary numbering system consists of 2 symbols: 0 and 1. Therefore, the base is 2.
In binary we count like this:
0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010 …

Any positive number can be represented as a sum of powers of 2, for example:

Example1: $1011_2 = 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0$

Example2: $100.11_2 = 1*2^2 + 0*2^1 + 0*2^0 + 1*2^{-1} + 1*2^{-2}$

Operations on binary numbers are the same as the operations on decimal numbers.

Here is a decimal and binary table:

Decimal	Binary			
0	*0*	*0*	*0*	*0*
1	*0*	*0*	*0*	*1*
2	*0*	*0*	*1*	*0*
3	*0*	*0*	*1*	*1*
4	*0*	*1*	*0*	*0*
5	*0*	*1*	*0*	*1*
6	*0*	*1*	*1*	*0*
7	*0*	*1*	*1*	*1*
8	*1*	*0*	*0*	*0*
9	*1*	*0*	*0*	*1*
10	*1*	*0*	*1*	*0*
11	*1*	*0*	*1*	*1*
12	*1*	*1*	*0*	*0*
13	*1*	*1*	*0*	*1*
14	*1*	*1*	*1*	*0*
15	*1*	*1*	*1*	*1*
	MSB			**LSB**

MSB: Most Significant Bit
LSB: Least Significant Bit

In this table we can notice how the digits, in the binary column, alternate going down the column:

- The rightmost digit (LSB) alternates between 0 and 1.
- The next digit, from the right, alternates between two 0s and two 1s.

And you can deduce the alternation for the rest of the digits. Now with this method we can write the next sixteen binary numbers by just repeating the first sixteen and putting a 1 in front.

Special (useful) values:

- $0000\ 0010_2 = 2^1 = 2$
- $0000\ 0100_2 = 2^2 = 4$
- $0000\ 1000_2 = 2^3 = 8$
- $0001\ 0000_2 = 2^4 = 16$
- $0010\ 0000_2 = 2^5 = 32$
- $0100\ 0000_2 = 2^6 = 64$
- $1000\ 0000_2 = 2^7 = 128$
- $10000\ 0000_2 = 2^8 = 256$

A set of 8 bits or one byte can represent 256 numbers from 0 to 255.

Base 2 and computers

The computer system is based on electrical wires. The electrical wire can have two states: current is present or current is NOT present. Consequently, binary numbers system is the natural numbering system to use with a computer. In addition, other electrical devices can be used to represent binary numbers. Such as, light bulbs or switches. In the next chapter, the basic concepts of electricity are covered in order to demystify how it is used inside the computer systems.

Addition in base 2 numbers

Arithmetic calculation in all positional numbering systems follows the same principles, so the addition table looks like this:

Operands	Sum
0 + 0	0
0 + 1	1
1 + 0	1
1 + 1	**0 (carry 1)**

Example 1: **$01_2 + 01_2 = 10_2$**

Binary Addition		
Carry	1	
	0	1
+	0	1
Sum	1	0

Example 2: **$01_2 + 01_2 + 01_2 = 11_2$**

Binary Addition		
Carry	1	
	0	1
	0	1
+	0	1
Sum	1	1

Example 3: **$1011101+1111001=11010110_2$**

Binary Addition								
Carry	1	1	1	1			1	
		1	0	1	1	1	0	1
	+	1	1	1	1	0	0	1
Sum	1	1	0	1	0	1	1	0

Conversion

Converting from binary to decimal

The position of a digit in a binary number determines its value.
Going from right to the left, the decimal value of the first digit is 2^0 times the digit value.
The value of the second digit is 2^1 times the digit value.
The value of the third digit is 2^2 times the digit value.
The value of the n^{th} digit is 2^{n-1} times the digit value.
In order to convert a binary number to decimal, the values of all the operations must be added.

Example1: **111_2 to Decimal**

$111_2 = (2^2*1) + (2^1*1) + (2^0*1)$
$= 4 + 2 + 1$
$= 7_{10}$

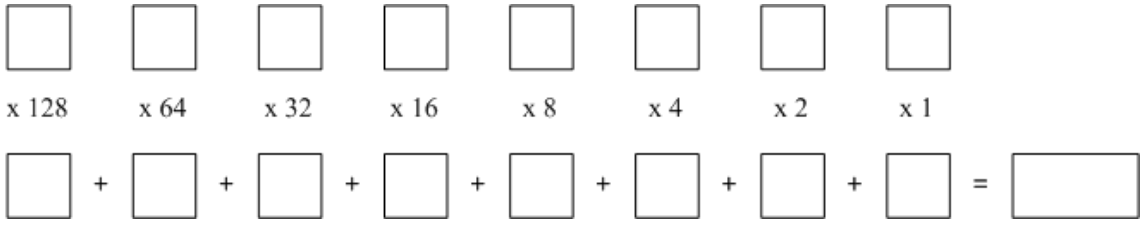
Example2: **1001_2 to Decimal**

$1001_2 = (2^3*1) + (2^2*0) + (2^1*0) + (2^0*1)$
$= 8 + 0 + 0 + 1$
$= 9_{10}$

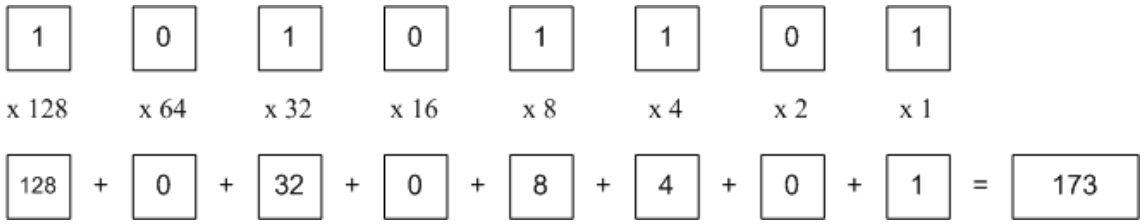
Example3: **1010_2 to Decimal**

$1010_2 = (2^3*1) + (2^2*0) + (2^1*1) + (2^0*0)$
$= 8 + 0 + 2 + 0$
$= 10_{10}$

Another way to do a binary to decimal conversion is by using this template:



The upper part of the template is for writing binary numbers as for the lower part, is the result of every binary digit multiplied by its corresponding value. The results of all the multiplications are added together to get the final result in decimal. Here is an example of converting 10101101_2 to decimal:



Converting from decimal to binary

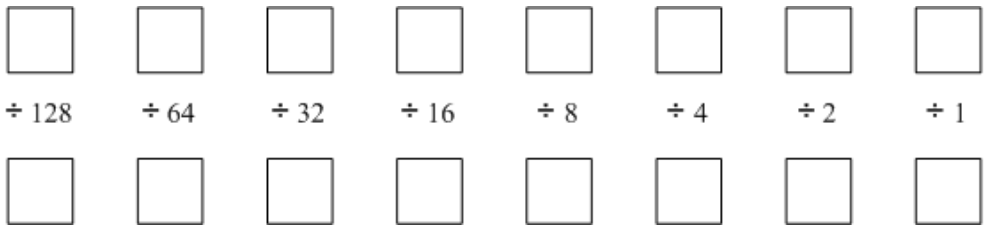
- Decimal numbers have 10 digits.
- Binary numbers have 2 digits.
- The number must be converted from 10 digits to 2 digits representation. How?
    - Divide the decimal number by two.
    - Save the remainder.
    - If the result of the division is divisible by two, repeat the process.
- The binary number is formed by writing the remainders off all the divisions starting from the last.

Example1: **7_{10} to Binary**
7/2 = 3 Remainder = 1
3/2 = 1 Remainder = 1
1/2 = 0 Remainder = 1 (last remainder)
Decimal 7_{10} = 111_2 Binary

Example2: **9_{10} to Binary**
9/2 = 4 Remainder = 1
4/2 = 2 Remainder = 0
2/2 = 1 Remainder = 0
1/2 = 0 Remainder = 1 (last remainder)
Decimal 9_{10} = 1001_2 Binary

Example3: **10_{10} to Binary**
10/2 = 5 Remainder = 0
5/2 = 2 Remainder = 1
2/2 = 1 Remainder = 0
1/2 = 0 Remainder = 1 (last remainder)
Decimal 10_{10} = 1010_2 binary

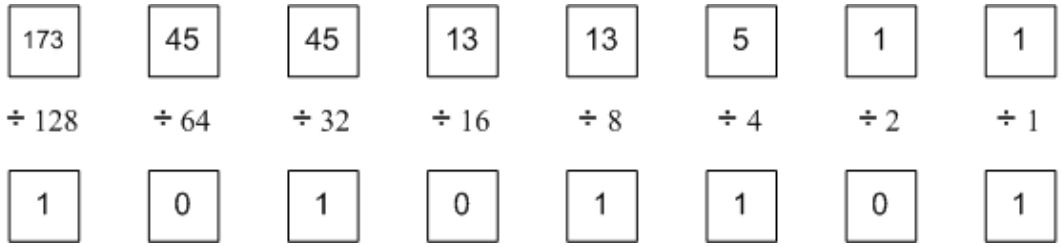
Also a template is available to convert from decimal to binary:



In this template the upper part is the decimal part and the lower one is binary. Here how it works:

- Start by the upper left square and write in it the decimal number that needs to be converted to binary.
- Divide the number by 128.
- Write the quotient in the lower left square. As for the remainder, write it in the next upper square.
- Repeat the same steps for the rest of the columns.
- The result is the lower part

Here is an example of converting 173_{10} to binary using the template:



Conversion from Hexadecimal to Binary

Each digit of a hexadecimal number can be converted to a **4bit** binary number **(16=24)**.

Example 1: **$FE5_{16}$ to Binary**

Hex to Binary: $FE5_{16}$		
F	***E***	***5***
1 1 1 1	1 1 1 0	0 1 0 1

Conversion from Binary to Hexadecimal

- In this case the conversion follows a reverse process than the one described in the previous paragraph. The binary number is converted **4 bit** at a time starting from the LSB (Least Significant Bit).

Example 1: **$111\ 1101\ 1001\ 1111_2$ to Hex**

Binary to Hex: $0111\ 1101\ 1001\ 1111_2$			
0 1 1 1	***1101***	***1001***	***1111***
7	D	***9***	***F***

Converting from hexadecimal to decimal

- The position of a digit in a hexadecimal number determines its value.
- Going from right to the left, the decimal value of the first digit is 16^0 times the digit value.
- The value of the second digit is 16^1 times the digit value.
- The value of the third digit is 16^2 times the digit value.
- The value of the n^{th} digit is 16^{n-1} times the digit value.
- In order to convert a hexadecimal number to decimal, the values of all the operations must be added.

Example1: **7_{16} to Decimal**

$$7_{16} = (16^0 * 7)$$
$$= 7$$
$$= 7_{10}$$

Example2: **10_{16} to Decimal**

$$10_{16} = (16^1 * 1) + (16^0 * 0)$$
$$= 16 + 0$$
$$= 16_{10}$$

Example3: **$A3C_{16}$ to Decimal**

$$A3C_{16} = (16^2 * 10) + (16^1 * 3) + (16^0 * 12)$$
$$= 2560 + 48 + 12$$
$$= 2620_{10}$$

Converting from decimal to hexadecimal

- Hexadecimal numbers have 16 digits.
- Binary numbers have 2 digits.
- The number must be converted from 16 digits to 2 digits representation. How?
  - Divide the decimal number by 16.
  - Save the remainder.
  - If the result of the division is divisible by 16, repeat the process.
- The hexadecimal number is formed by writing the remainders off all the divisions starting from the last.

***Example1*: 7_{10} to Hexadecimal.**
7/16 = 0 Remainder = 7 (last remainder)
Decimal 7_{10} = 7_{16} Hexadecimal

***Example2*: 16_{10} to Hexadecimal**
16/16 = 1 Remainder = 0
1/16 = 0 Remainder = 1 (last remainder)
Decimal 16_{10} = 10_{16} Hexadecimal

***Example3*: 2620_{10} to Hexadecimal**
2620/16 = 163 Remainder = C
163/16 = 10 Remainder = 3
10/16 = 0 Remainder = A (last remainder)
Decimal 2620_{10} = $A3C_{16}$ Hexadecimal