Number System

In Mathematics, Number System is a set of numbers together with one or more operations. Such as multiplication, subtraction etc.

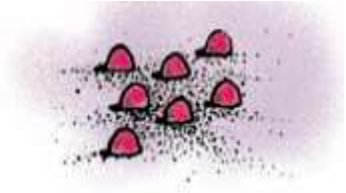
There are various types of number systems such as decimal number systems, binary number systems, hexadecimal number system and many more.

History of Number Systems

Archaeologists and anthropologists study ancient civilizations. They have helped us to understand how people long ago counted and recorded numbers. Their findings suggest that the first attempts at counting were to use a **tally.**

For example, in ancient times people used items to represent numbers:

knots on a rope showed the rows of corn planted



pebbles on the sand showed the number of traps set for fish

notches cut on a branch showed the number of new lambs born

Number System

Why Learn The Number Systems?

When humans are speaking to one another, they speak in a particular language. This language

is made of words and letters.

Although we type words and letters in the computer, the computer does not understand the words and letters. Rather, those words and letters are translated into numbers. Computers “talk” and understand in numbers.

As a computer programmer or an IT professional, we need to understand the three most common number systems that are frequently used in computers.

Those number systems are:

- Decimal
- Hexadecimal
- Binary

If you have an understanding of the number systems,
you will understand the following topics in computer
science:

- We will understand how to calculate network addresses.
- We will understand how to read memory address locations in core dumps.
- We will understand the color definitions in web and application programming.
- We will understand many other concepts.

Modern Number Systems

Since childhood, we have learned to do our computations using the numbers 0 - 9, the digits of the decimal number system.

In fact, we are so accustomed to working with decimal Numbers that we hardly think about their use. We balance our checkbooks, pay monthly bills, and even solve algebra homework with the aid of the decimal number system.

Modern Number Systems

Considering the widespread use of decimal system, why should anyone bother to study the other number system?

The answer is found in something that is almost as widespread as decimal numbers:

computers

While it is fine for us to use ten digits for our computations, computers do not have this luxury.

Every computer processor is made of millions of tiny switches that can be turned off or on. Since these switches only have two states, it makes sense for a computer to perform its computations with a number system that only has two digits: the binary number system.

These digits (0 and 1) are called bits and correspond to the off/on positions of the switches in the computer processor. With only these two digits, a computer can perform all the arithmetic that we can with ten digits.

The study of number systems is useful to the student of computing due to the fact that number systems other than the familiar decimal (base 10) number system are used in the computer field.

Digital computers internally use the binary (base 2) number system to represent data and perform arithmetic calculations. The binary number system is very efficient for computers, but not for humans. Representing even relatively small numbers with the binary system requires working with long strings of ones and zeroes.

The Decimal Number System

The decimal number system is the system that is commonly known with people because it is most frequently used in arithmetic and in everyday life. The decimal number system is also known as the *base-10 number system*, *because* the base of any number system is determined by the number of digits in the system. decimal is a base-10 system since it uses ten digits

For example, the first position (the furthest right) is represented as 10^0. The second position (one from furthest right) is represented as 10^1, and so forth.

The prefix “deci-” stands for 10

The decimal number system is a Base 10 number system:

There are 10 symbols that represent quantities:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Each place value in a decimal number is a power of 10.

The decimal number system that we are all familiar with is a positional number system.

You know that in the decimal number system, the successive positions to the left of the decimal point represent units (ones) , tens, hundreds, thousands, etc. Put another way, each position represents a specific power of base 10.

Decimal Number System

10^3 10^2 10^1 10^0

1000 100 10 1

1 4 9 2

1 x 1000 =	1000
4 x 100 =	400
9 x 10 =	90
2 x 1 = +	2
	1492

The number 1492 is actually a representation of decimal positions. The first position at the furthest right of the number represents 10^0. The next position represents 10^1. The next position represents 10^2. As we move further left of the number, each power is incremented by 1.

- In order to determine what the number actually is in the position, take the number that appears in the position, and multiply it by 10^x, where x represents the power.

- Add all the numbers together from each position to find out what the number is.

Background Information

Any number to the 0 (zero) power is 1.

– $4^0 = 1$ $16^0 = 1$ $1,482^0 = 1$.

Any number to the 1st power is the number itself.

– $10^1 = 10$ $49^1 = 49$ $827^1 = 827$

Binary System

The prefix “bi-” stands for 2

The binary number system is a Base 2 number system:

There are 2 symbols that represent quantities: 0, 1

Each place value in a binary number is a power of 2.

Why binary?

The original computers were designed to be high-speed calculators.

The designers needed to use the electronic components available at the time.

The designers realized they could use a simple coding system--the binary system-- to represent their numbers

Bits and Bytes

A binary digit is a single numeral in a binary number.

Each 1 and 0 in the number below is a binary digit:

1 0 0 1 0 1 0 1

The term “binary digit” is commonly called a “bit.”

Eight bits grouped together is called a “byte.”

The same principles of positional number systems we applied to the decimal number system can be applied to the binary number system.

However, the base of the binary number system is two, so each position of the binary number represents a successive power of two. From right to left, the successive positions of the binary number are weighted 1, 2, 4, 8, 16, 32, 64, etc. A list of the first several powers of 2 follows:

$2^0 = 1$ $2^1 = 2$ $2^2 = 4$ $2^3 = 8$
$2^4 = 16$ $2^5 = 32$

Binary Number System

1 0 1 1

2^3 2^2 2^1 2^0

8 4 2 1

1 x 8 = 8

0 x 4 = 0

1 x 2 = 2

1 x 1 = + 1

11

The number 1011 is a binary number. The first position at the furthest right of the number represents 2^0. The next position represents 2^1. The next position represents 2^2.As we move further left of the number, each power is incremented by 1.

In order to determine what the number actually is in the position, take the number that appears in the position, and multiply it by 2^x, where x represents the power. Note that in the binary number system, the only two numbers that can appear in each position is either 0 or 1.

Add all the numbers together from each position to find out what the decimal number is.

Hexadecimal Numbers

The prefix “hexa-” stands for 6 and the prefix “deci-’ stands for 10. The hexadecimal number system is a Base 16 number system:

There are 16 symbols that represent quantities:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Each place value in a hexadecimal number is a power of 16.

Why Hexadecimal

The hexadecimal number system is used as an intermediary system in computers, such as a representation of memory addresses or a representation of colors.

The hexadecimal provides us with a shorthand method of working with binary numbers. Each group of four binary digits can be represented by a single hexadecimal digit

viewing the data in a "shorthand" representation (such as hex) is less tedious than viewing the data in binary representation.

Hexadecimal Numbers

Dec	Bin	Hex	Dec	Bin	Hex
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

Hexadecimal Number

2 F A 4

16^3 16^2 16^1 16^0

4096 256 16 1

2 x 4096 =	8192
15 x 256 =	3840 (convert “F” to 15)
10 x 16 =	160 (convert “A” to 10)
4 x 1 =+	4
	12,196

- The number 17AF is a hexadecimal number. The first position at the furthest right of the number represents 16^0. The next position represents 16^1. The next position represents $16^{2.}$ As we move further left of the number, each power is incremented by 1.

- In order to determine what the number actually is in the position, take the number that appears in the position, and multiply it by 16x, where x represents the power. Note that in the hexadecimal number system, letters A through F represent numbers 10 through 15, respectively.

- Add all the numbers together from each position to find out what the decimal number is.

Converting Binary-to-Hexadecimal & Hexadecimal-to-Binary

Converting a binary number to its hexadecimal equivalent or vice-versa is a simple matter. Four binary digits are equivalent to one hexadecimal digit .

To convert from binary to hexadecimal, divide the binary number into groups of 4 digits starting on the right of the binary number. If the leftmost group has less than 4 bits, put in the necessary number of leading zeroes on the left. For each group of four bits, write the corresponding single hex digit.

Example 1: 1101001101110111^{2} = ?^{16}

1101	0011	0111	0111
D	3	7	7

To convert from hexadecimal to binary, write the corresponding group of four binary digits for each hex digit.

Example 1: 1BE9^{16} = ?^{2}

1	B	E	9
0001	1011	1110	1001

The Octal Number System

- The same principles of positional number systems we applied to the decimal and binary number systems can be applied to the octal number system. However, the base of the octal number system is eight, so each position of the octal number represents a successive power of eight. From right to left, the successive positions of the octal number are weighted 1, 8, 64, 512, etc. A list of the first several powers of 8 follows:

$8^0 = 1$ $8^1 = 8$ $8^2 = 64$ $8^3 = 512$

$8^4 = 4096$

Converting an Octal Number to a Decimal Number

- To determine the value of an octal number (367^8, for example), we can expand the number using the positional weights as follows:

$7 \text{ x } 8^0 =$	$7 \text{ x } 1 =$	7
$6 \text{ x } 8^1 =$	$6 \text{ x } 8 =$	48
$3 \text{ x } 8^2 =$	$3 \text{ x } 64 =$	192

		247^{10}

Converting a Decimal Number to an Octal Number

- To convert a decimal number to its octal equivalent, the remainder method (the same method used in converting a decimal number to its binary equivalent) can be used. To review, the remainder method involves the following four steps:

(1) Divide the decimal number by the base (in the case of octal, divide by 8).

(2) Indicate the remainder to the right.

(3) Continue dividing into each quotient (and indicating the remainder) until the divide operation produces a zero quotient.

(4) The base 8 number is the numeric remainder reading from the last division to the first (if you start at the bottom, the answer will read from top to bottom).

Converting a Decimal Number to an Octal Number

Example 1: Convert the decimal number 465^{10} to its octal equivalent:

The answer, is 721, so $465^{10} = 721^{8}$.

Converting Binary-to-Octal or Octal-to-Binary

- Converting a binary number to its octal equivalent or vice-versa is a simple matter. Three binary digits are equivalent to one octal digit .

- To convert from binary to octal, divide the binary number into groups of 3 digits starting on the right of the binary number. If the leftmost group has less than 3 bits, put in the necessary number of leading zeroes on the left. For each group of three bits, write the corresponding single octal digit.

- Example 1: 1101 001101110111^{2} = ?^{8}

Answer: Bin:	001	101	001	101	110	111
Oct:	1	5	1	5	6	7

To convert from octal to binary, write the corresponding group of three binary digits for each octal digit

Example 1: 1764^{8} = ?^{2}

Answer: Oct:	1	7	6	4
Bin:	001	111	110	100

Decimal Number System:

Decimal Addition:

Adding two number corresponds to add the values of their digits at respective positions.

$$\begin{array}{rr} & 987 \\ + & 657 \\ \hline & 1644 \\ \hline \end{array}$$

Binary Number System

Binary Addition

Adding two binary numbers together is easy, keeping in mind the following four addition rules:

(1) **0 + 0 = 0**

(2) **0 + 1 = 1**

(3) **1 + 0 = 1**

(4) **1 + 1 = 10**

Binary Number System

(4) **1 + 1 = 10**

Note in the last example that it was necessary to "carry the 1“. In the decimal system, we used up all the digits after the tenth counting number, 9.

The same method is used in both systems to come up with the next number: place a zero in the "ones" position and start over again with one in the next position on the left. In the decimal system, this gives ten, or 10.

In binary, it gives 10^2, which is read "one-zero, base two."

Consider the following binary addition problems and note where it is necessary to carry the 1:

$$\begin{array}{r} 1110 \\ +\ 1010 \\ \hline 11010 \end{array} \qquad \begin{array}{r} 11111 \\ +\quad 10101 \\ \hline 110100 \end{array} \qquad \begin{array}{r} 110101 \\ +\quad 011101 \\ \hline 1010010 \end{array}$$

Subtraction Using Complements

Subtraction in any number system can be accomplished through the use of complements. A complement is a number that is used to represent the negative of a given number.

When two numbers are to be subtracted, the subtrahend* can either be subtracted directly from the minuend or, the complement of the subtrahend can be added to the minuend to obtain the difference.

Subtraction Using Complements

When the latter method is used, the addition will produce a high-order (leftmost) one in the result (a "carry"), which must be dropped.

This is how the computer performs subtraction: it is very efficient for the computer to use the same "add" circuitry to do both addition and subtraction; thus, when the computer "subtracts", it is really adding the complement of the subtrahend to the minuend.

Subtraction Using Complements

To understand complements, consider a mechanical register, such as a car mileage indicator, being rotated backwards. A five-digit register approaching and passing through zero would read as follows:

00005

00004

00003

00002

00001

00000

99999

99998

99997

It should be clear that the number 99998 corresponds to -2. Furthermore, if we add

$$\begin{array}{r} 00005 \\ \underline{+\ 99998} \\ 1\ 00003 \end{array}$$

and ignore the carry to the left, we have effectively formed the operation of subtraction: 5 - 2 = 3.

Subtraction Using Complements

In the example above, subtraction with the use of complements was accomplished as follows:

(1) We were dealing with a five-digit subtrahend that had a value of 00002. First, each digit of the subtrahend was subtracted from 9 (this preliminary value is called the *nine's complement of the subtrahend):*

$$\begin{array}{ccccc} 9 & 9 & 9 & 9 & 9 \\ -0 & -0 & -0 & -0 & -2 \\ \hline 9 & 9 & 9 & 9 & 7 \end{array}$$

Subtraction Using Complements

(2) Next, 1 was added to the nine's complement of the subtrahend (99997) giving the ten's complement of subtrahend (99998):

$$\begin{array}{ccccc} 9 & 9 & 9 & 9 & 7 \\ & & & + & 1 \\ \hline 9 & 9 & 9 & 9 & 8 \end{array}$$

(3) The ten's complement of the subtrahend was added to the minuend giving 100003. The leading (carried) 1 was dropped, effectively performing the subtraction of 00005 - 00002 = 00003.

$$\begin{array}{cccccc}
 & 0 & 0 & 0 & 0 & 5 \\
+ & 9 & 9 & 9 & 9 & 8 \\
\hline
1 & 0 & 0 & 0 & 0 & 3
\end{array}$$

The answer can be checked by making sure that 2 + 3 = 5.

TRY THIS: Solve the following subtraction problems using the complement method:

(a) 5086 - 2993 =

(b) 8391 - 255 =

Binary Subtraction

We will use the complement method to perform subtraction in binary. As mentioned in the previous section, the use of complemented binary numbers makes it possible for the computer to add or subtract numbers using only circuitry for addition .

The computer performs the subtraction of A - B by adding A + (two's complement of B) and then dropping the carried 1.

Subtraction Using Complements

The steps for subtracting two binary numbers are as follows:

(1) Compute the one's complement of the subtrahend by subtracting each digit of the subtrahend by 1. A shortcut for doing this is to simply reverse each digit of the subtrahend - the 1's become 0's and the 0's become 1's.

(2) Add 1 to the one's complement of the subtrahend to get the two's complement of the subtrahend.

(3) Add the two's complement of the subtrahend to the minuend and drop the high-order 1. This is your difference.

Subtraction Using Complements

Example 1: Compute 11010101_2 - 1001011_2

(1) Compute the one's complement of 1001011_2 by subtracting each digit from 1 (note that a leading zero was added to the 7-digit subtrahend to make it the same size as the 8-digit minuend):

$$\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -0 & -1 & -0 & -0 & -1 & -0 & -1 & -1 \\ \hline 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{array}$$

(Note that the one's complement of the subtrahend causes each of the original digits to be reversed.)

Subtraction Using Complements

(2) Add 1 to the one's complement of the subtrahend, giving the two's complement of the subtrahend:

$$\begin{array}{cccccccc} 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ & & & & & & & +1 \\ \hline 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{array}$$

Subtraction Using Complements

(3) Add the two's complement of the subtrahend to the minuend and drop the high-order 1, giving the difference:

  11010101

+ 10110101

110001010

So $11010101^2 - 1001011^2 = 10001010^2$.

The answer can be checked by making sure that $1001011^2 + 10001010^2 = 11010101^2$.

Hexadecimal Addition :

- One consideration is that if the result of an addition is between 10 and 15, the corresponding letter A through F must be written in the result:

$$\begin{array}{cccc} & 1 & 9 & 5 \\ + & 3 & 1 & 9 \\ \hline & 4 & \mathrm{A} & \mathrm{E} \end{array}$$

- In the example above, 5 + 9 = 14, so an "E" was written in that position; 9 + 1 = 10, so an "A" was written in that position.

Hexadecimal Addition :

- A second consideration is that if either of the addends contains a letter A through F, convert the letter to its decimal equivalent (either by memory or by writing it down) and then proceed with the addition:

$$\begin{array}{cccc} & 3 & \underset{10}{A} & 2 \\ + & 4 & 1 & \underset{12}{C} \\ \hline & 7 & B & E \end{array}$$

Hexadecimal Addition :

- A third consideration is that if the result of an addition is greater than 15, you must subtract 16 from the result of that addition, put down the difference of that subtraction for that position, and carry a 1 over to the next position, as shown below:

		1	
	D	E	B
	13	14	11
+	1	0	E
			14
		11+14=25	
		25-16=9	
	E	F	9

Hexadecimal Addition :

- In the example above, when B^{16} (11^{10}) was added to E^{16} (14^{10}), the result was 25^{10}. Since 25^{10} is greater than 15^{10}, we subtracted 16^{10} from the 25^{10} to get 9^{10}. We put the 9 down and carried the 1 over to the next position.

Hexadecimal Subtraction

- We will use the complement method to perform hexadecimal subtraction. The steps for subtracting two hexadecimal numbers are as follows:

1. Compute the 15's complement of the subtrahend by subtracting each digit of the subtrahend by 15.
2. Add 1 to the 15's complement of the subtrahend to get the 16's complement of the subtrahend.
3. Add the 16's complement of the subtrahend to the minuend and drop the high-order 1. This is your difference.

Hexadecimal Subtraction

Example 1: Compute ABED^{16} - $\mathrm{1FAD}^{16}$

(1) Compute the 15's complement of $\mathrm{1FAD}^{16}$ by subtracting each digit from 15:

$$\begin{array}{cccc} 15 & 15 & 15 & 15 \\ -1 & -5 & -\mathrm{A} & -\mathrm{D} \\ \hline \mathrm{E} & 0 & 5 & 2 \end{array}$$

Hexadecimal Subtraction

(2) Add 1 to the 15's complement of the subtrahend, giving the 16's complement of the subtrahend:

$$\begin{array}{ccccc} & E & 0 & 5 & 2 \\ + & & & & 1 \\ \hline & E & 0 & 5 & 3 \end{array}$$

(3) Add the 16's complement of the subtrahend to the minuend and drop the high-order 1, giving the difference:

Hexadecimal Subtraction

$$\begin{array}{cccccc}
 & & & 1 & 1 & \\
 & & A & B & E & D \\
+ & & E & 0 & 5 & 3 \\
\hline
 & & 24-16=8 & & 20-16=4 & 16-16=0 \\
 & \cancel{1} & 8 & C & 4 & 0
\end{array}$$

So $ABED^{16}$ - $1FAD^{16}$ = $8C40^{16}$

The answer can be checked by making sure that

$1FAD^{16}$ + $8C40^{16}$ = $ABED^{16.}$

Octal Addition

- Octal addition is performed just like decimal addition, except that if a column of two addends produces a sum greater than 7, you must subtract 8 from the result, put down that result, and carry the 1.

- Example 2: Add 7652^8 + 4574^8 (carries required):

	1	1		
	7	6	5	2
+	4	5	7	4
	12 - 8 = 4	12 - 8 = 4	12 - 8 = 4	
1	4	4	4	6

Octal Subtraction

- We will use the complement method to perform octal subtraction. The steps for subtracting two octal numbers are as follows:
- (1) Compute the seven's complement of the subtrahend by subtracting each digit of the subtrahend by 7.
- (2) Add 1 to the seven's complement of the subtrahend to get the eight's complement of the subtrahend.
- (3) Add the eight's complement of the subtrahend to the minuend and drop the high-order 1. This is your difference.

Octal Subtraction

Example 1: Compute 7526^8 - 3142^8

(1) Compute the seven's complement of 3142^8 by subtracting each digit from 7:

7 7 7 7

- 3 - 1 - 4 - 2

4 6 3 5

(2) Add 1 to the seven's complement of the subtrahend, giving the eight's complement of the subtrahend:

4 6 3 5

+ 1

4 6 3 6

Octal Subtraction

(3) Add the eight's complement of the subtrahend to the minuend and drop the high-order 1, giving the difference:

	1		1	
	7	5	2	6
+	4	6	3	6
	12 - 8 = 4	11 - 8 = 3		12 - 8 = 4
1	4	3	6	4

So $7526^8 - 3142^8 = 4364^8$