CSCI 112
Introduction to computer Science -I

Instructor: Santanu Banerjee

Number Systems

Number Systems

- Binary (0,1)
- Octal (0-7)
- Decimal (0-9)
- Hexadecimal (0-9, A-F)

Decimal Numbers

Base or Radix : 10

$$4053 = 3 \times 1 + 5 \times 10 + 0 \times 10^2 + 4 \times 10^3$$

1s (10^0)

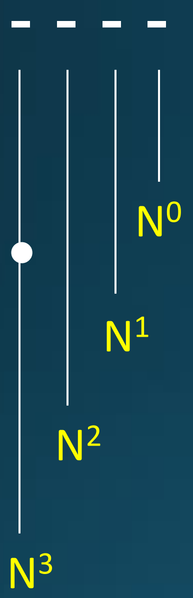
10s (10^1)

100s (10^2)

1000s (10^3)

General Rule

Base or Radix : N



If positions are 0,1,2,3…

Value at position p : N^{p-1}

Binary Numbers

Base or radix : 2

$10111 = 1 \times 1 + 1 \times 2 + 1 \times 2^2 + 0 \times 2^3 + 1 \times 2^4$

1s

2s

4s

8s

16s

$10111_2 = 23_{10}$

Numbers are : 0 and 1

Called *bits* (binary digits)

Octal Numbers

Base or radix : 8

$10607 = 7 \times 1 + 0 \times 8 + 6 \times 8^2 + 0 \times 8^3 + 1 \times 8^4$

1s

8s

64s

512s

4096s

$10607_8 = 4487_{10}$

Numbers are : 0, 1,2,. . . ,7

Hexadecimal Numbers

Base or radix : 16

$10607 = 7 \times 1 + 0 \times 16 + 6 \times 16^2 + 0 \times 16^3 + 1 \times 16^4$

1s

16s

256s

4096s

65536s

$10607_{16} = 67079_{10}$

Numbers are : 0, 1,2,. . . ,9 and A,..,F

A is 10(decimal), B is 11(decimal)

. . . . .F is 15(decimal)

Hexadecimal to Decimal

Base 16

$5CB = 11 \times 1 + 12 \times 16 + 5 \times 16^2$

1s

16s

256s

$5CB_{16} = 1483_{10}$

Converting decimal to base-X

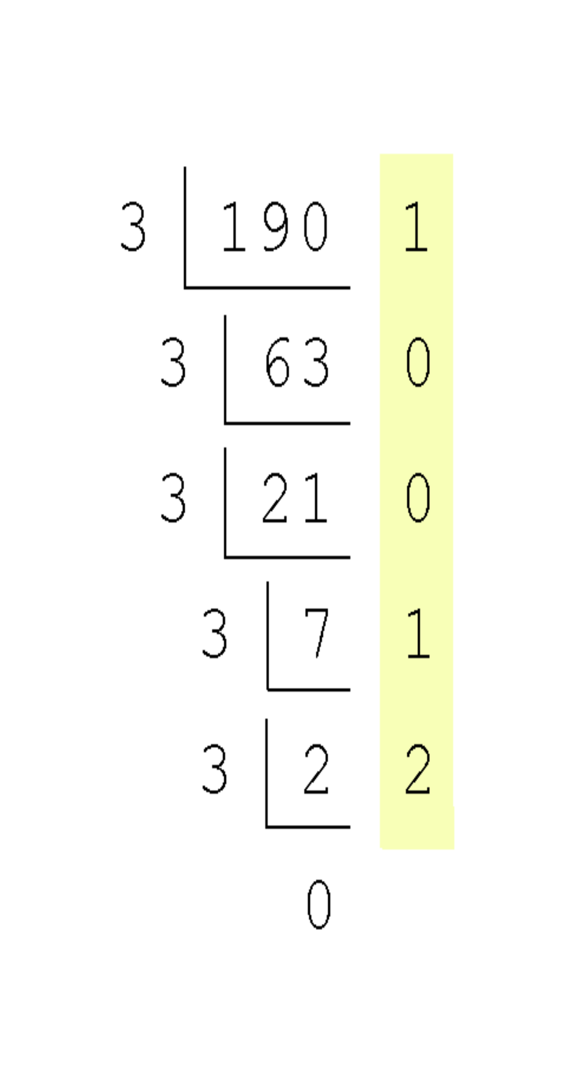
- Algorithm

repeat

Divide *DecimalNumber* by X;
Get *Quotient* and *Remainder*;
Remainder (in base-X) is the next digit (right to left);
DecimalNumber := *Quotient*;

until *DecimalNumber* = 0;

Example: $190_{10} = 21001_{3}$



Why do we convert?

- The binary numbering system is the most important radix system for digital computers.
- However, it is difficult to read long strings of binary numbers -- and even a modestly-sized decimal number becomes a very long binary number.
  - For example: $11010100011011_2 = 13595_{10}$
- For compactness and ease of reading, binary values are usually expressed using the hexadecimal, or base-16, numbering system.

Conversion table

Decimal	Binary	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Conversion(B←→H)

Binary to Hexadecimal

- Group binary digits in sets of 4 bits(add zeroes to the left, as needed)
- Convert each set of 4 binary digits into a single hexadecimal digit
- Example: 10100011 => 1010|0011 => A|3 => A3

Hexadecimal to Binary

- Convert each hexadecimal digit into 4 bits(add zeroes to the left, as needed)
- Concatenate the sets of 4 binary digits into a single set of binary digits(preserve the order)
- Example: 2C => 2|C => 0010|1100 => 00101100

Conversion(B←→O)

Binary to Octal

- Group binary digits in sets of 3 bits(add zeroes to the left, as needed)
- Convert each set of 3 binary digits into a single octal digit
- Example: 10100011 => 010|100|011 => 2|4|3 => 243

Octal to Binary

- Convert each octal digit into 3 bits(add zeroes to the left, as needed)
- Concatenate the sets of 3 binary digits into a single set of binary digits(preserve the order)
- Example: 327 => 3|2|7 => 011|010|111 => 011010111

Conversion(H←→O)

Hexadecimal to Octal

- Use either of

(i)Hexadecimal to Binary and then Binary to Octal

(iI)Hexadecimal to Decimal and then Decimal to Octal

Octal to Hexadecimal

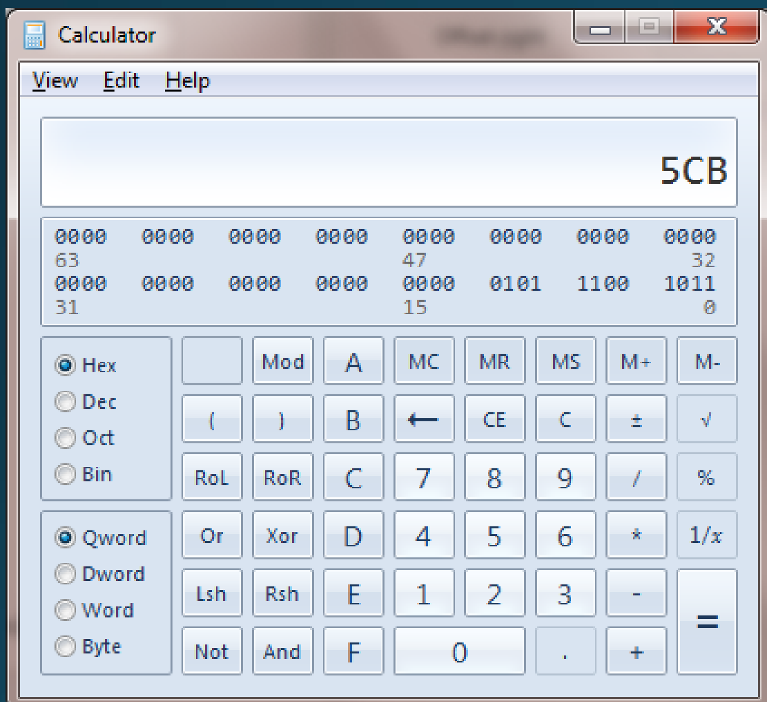
- Use either of

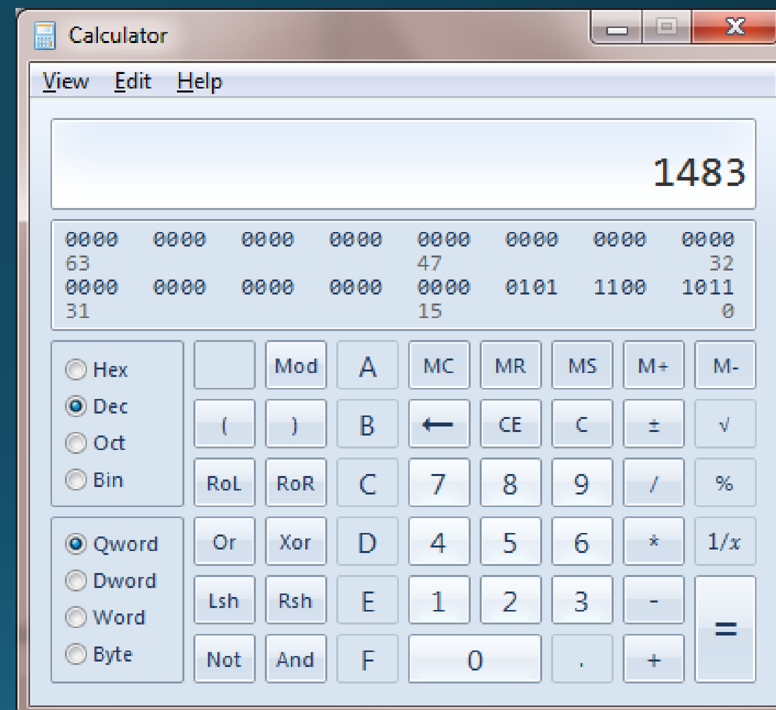
(i) Octal to Binary and then Binary to Hexadecimal

(iI) Octal to Decimal and then Decimal to Hexadecimal

Converting between bases

- You can use a programmer calculator





- Or use online tools
- Or by yourself

Higher order units for bits

- *Byte*
  - is a group of eight bits.
  - is the smallest possible *addressable* unit of computer storage.
- The term, "addressable," means that a particular byte can be retrieved by specifying its location in memory.

Higher order units for bits

- A *word* is a contiguous group of bytes.
  - Word sizes : 16, 32, or 64 bits depending on the architecture
  - In a word-addressable system, a word is the smallest addressable unit of storage.
- A group of four bits is called a *nibble*.
  - Bytes, therefore, consist of two nibbles:
    - a high-order nibble
    - a low-order nibble

Prefix/levels for byte units

- Like metric system, each level is 2^{10} times higher or lower than the next level.

File Storage Capacity by Powers of Two (Base 2)										
	bit	byte	Kilobyte	Megabyte	Gigabyte	Terabyte	Petabyte	Exabyte	Zettabyte	Yottabyte
bit	2^0	2^3	2^13	2^23	2^33	2^43	2^53	2^63	2^73	2^83
byte	2^3	2^0	2^10	2^20	2^30	2^40	2^50	2^60	2^70	2^80
Kilobyte	2^13	2^10	2^0	2^10	2^20	2^30	2^40	2^50	2^60	2^70
Megabyte	2^23	2^20	2^10	2^0	2^10	2^20	2^30	2^40	2^50	2^60
Gigabyte	2^33	2^30	2^20	2^10	2^0	2^10	2^20	2^30	2^40	2^50
Terabyte	2^43	2^40	2^30	2^20	2^10	2^0	2^10	2^20	2^30	2^40
Petabyte	2^53	2^50	2^40	2^30	2^20	2^10	2^0	2^10	2^20	2^30
Exabyte	2^63	2^60	2^50	2^40	2^30	2^20	2^10	2^0	2^10	2^20
Zettabyte	2^73	2^70	2^60	2^50	2^40	2^30	2^20	2^10	2^0	2^10
Yottabyte	2^83	2^80	2^70	2^60	2^50	2^40	2^30	2^20	2^10	2^0