Number System

Lecture 2

What is Data

Data is simply

any **numbers**, **letters** or **symbols** that can be entered into a computer system.

Data values **don't have any meaning** unless we put them into **context**

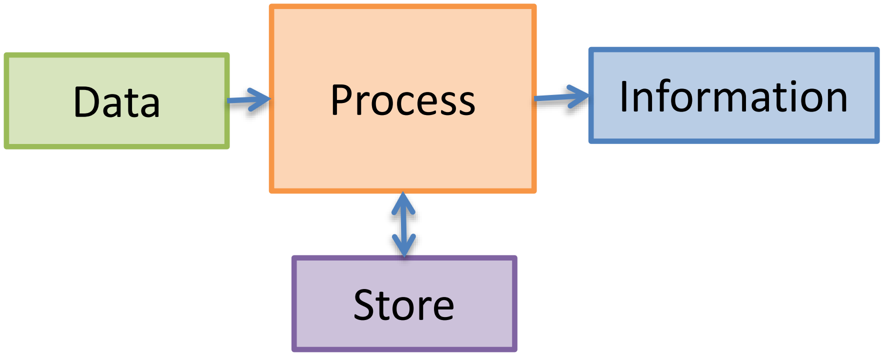
Information = Data + Context

What is a System

1. Something feeds into the system (the **input**)
2. The system does something with the input (the **process**)
3. The process gives a result (the **output**)

What is a Computer?

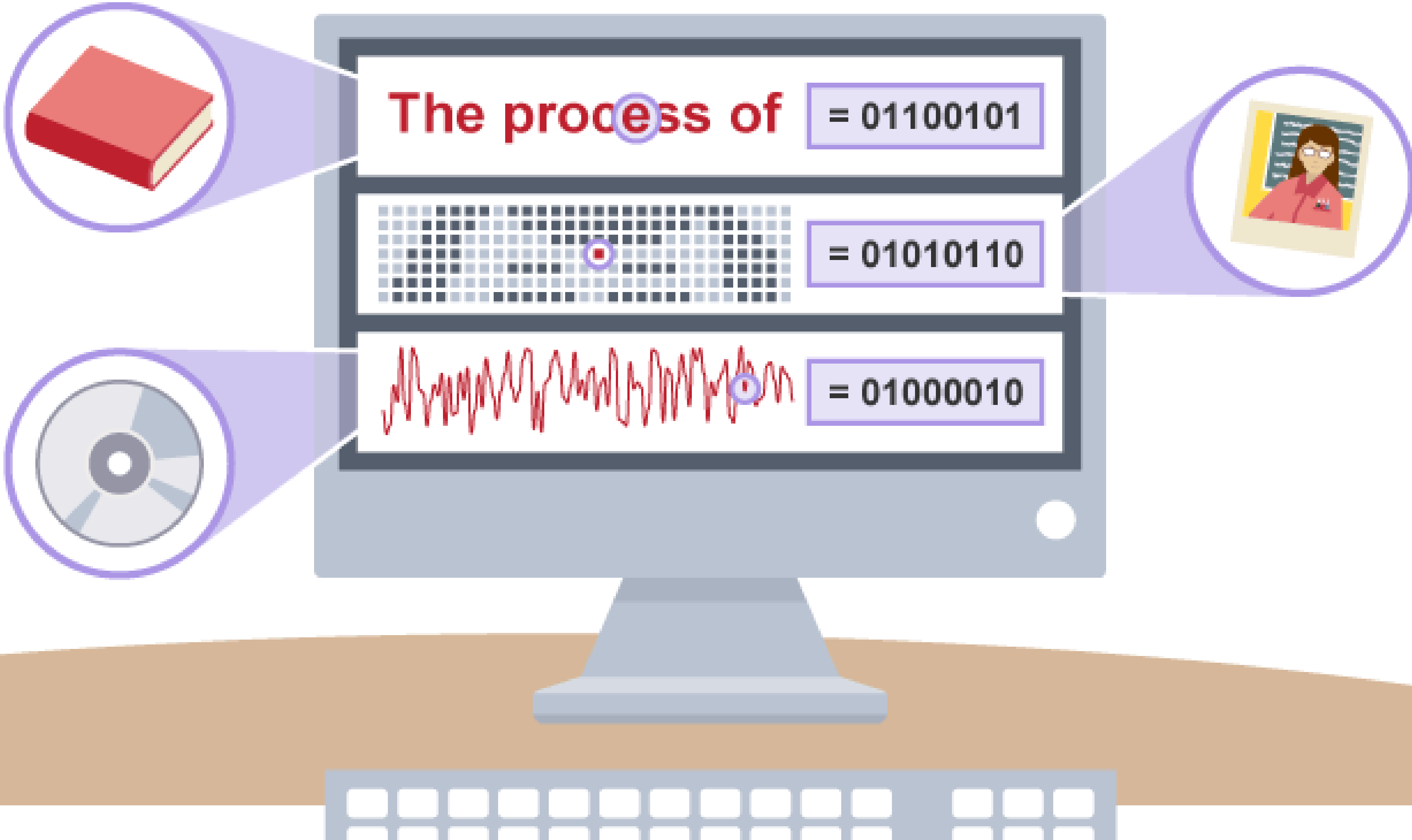
- A computer is a device that stores and processes information according to a set of instructions.



Activity

Data Name	Data Type	Example Data
Name	Text	"Bob Gripper"
Height	Real	1.85
Date of Birth	Date	19 May 1980
Phone No.	Alphanumeric	92-42-11128128
Pay Rate	Currency	£35.75
Tax Rate	Percentage	15%

Computers Process Data

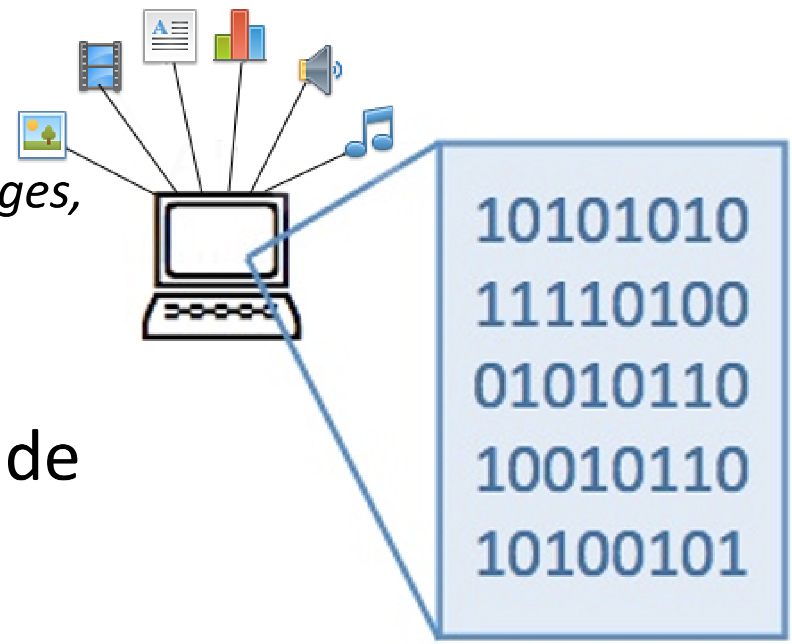


Computers Process Data

- Computers are used to process all types of information in a broad spectrum fields.
  - **Numeric data** consisting of Integers and real numbers are used in programs calculating payroll. We typically perform **arithmetic operations** on numeric data.
  - **Strings** of **alphabets** and numbers (**Alphanumeric Data**) are processed in customer record keeping systems.
  - **Multimedia** content including images, sound and text are frequently used in a large collection of application areas.
  - **Signals** representing various types of information like temperature, pressure, presence or absence of objects etc. are processed by computers in Robotics, IoT, monitoring and control applications.

How is Data Actually Stored in Computer

Everything that is stored and processed inside a computer *(all data, information, instructions, files, images, etc.)* is stored as **Numbers**

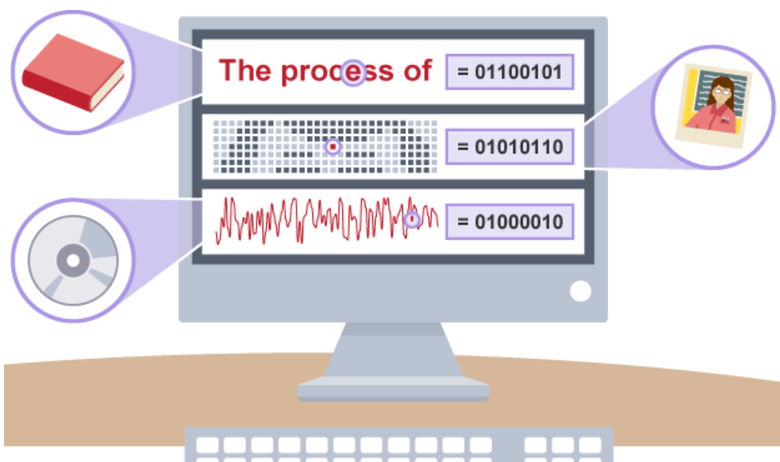


Digital computers have been made such that all data and instructions(program) for processing must be stored in computers memory before processing.

How to store *text* and *pictures* as numbers?

- The solution is to use **numeric codes**:
  - Different **letters** in a text document are given different numeric codes
  - Different **pixels** (colored dots) in an image are given different numeric codes
  - Different **sounds** in a music file are given different numeric codes

Everything is numbers!



What can be represented using a Bit

- Computers use binary - the digits 0 and 1 - to store data.
- A binary digit, or bit, is the smallest unit of data in computing.

- Single Bit can be used to represent two different quantities
  - ON means TRUE and OFF means FALSE
  - ON means number 79 and OFF means number -23
  - ON means 23.5 and OFF means 39.25
  - ON means **RED COLOR** and OFF means **BLUE COLOR**

- Most commonly ON means 1 and OFF means 0 and therefore **Bit** is also known as **Bi**nary Digi**t** **(Bit)**

Memory Measuring Units
(As viewed by computer scientists)

- Bits can be grouped together to make them easier to work with. A group of 8 bits is called a byte.

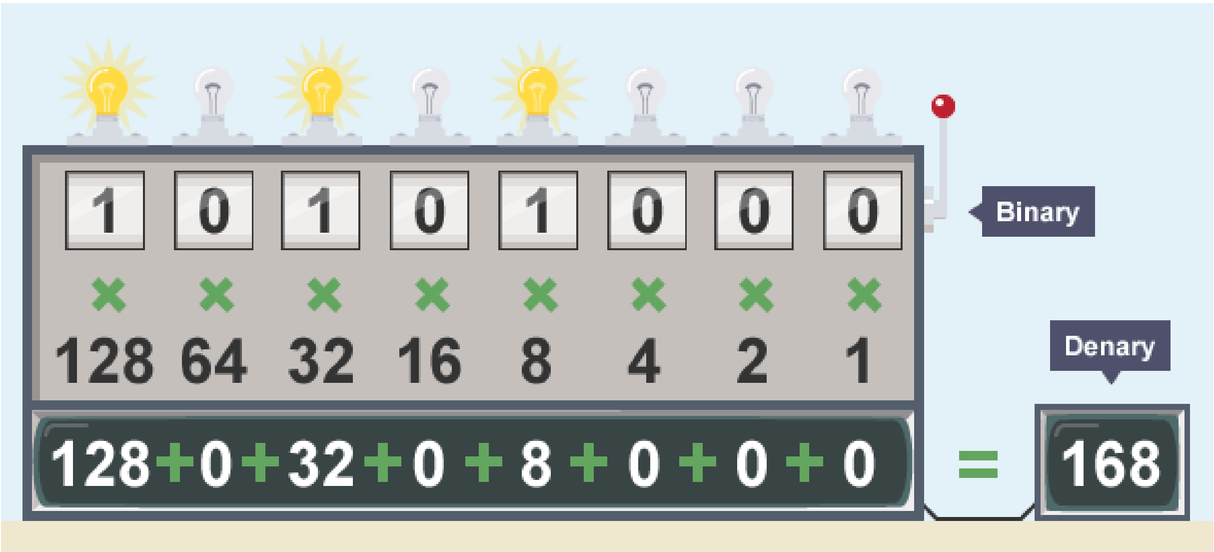
UNIT	ABBREVIATION	STORAGE
Bit	B	Binary Digit, Single 1 or 0
Nibble	-	4 bits
Byte/Octet	B	8 bits
Kilobyte	KB	1024 bytes
Megabyte	MB	1024 KB
Gigabyte	GB	1024 MB
Terabyte	TB	1024 GB
Petabyte	PB	1024 TB
Exabyte	EB	1024 PB
Zettabyte	ZB	1024 EB
Yottabyte	YB	1024 ZB

Storage units (www.byte-notes.com

Most computers can process millions of bits every second. A hard drive's storage capacity is measured in gigabytes or terabytes. RAM is often measured in megabytes or gigabytes.

Bytes as Numbers

- We can view each byte as a binary number. For Example the following Binary number (10101000) represents the quantity ONE Hundred and SIXTY EIGHT



- Can you see the similarity between Binary and Decimal numbers?

Exercise # 1

- What Quantities are Represented by the following 8-bit binary numbers.

Bit#	7	6	5	4	3	2	1	0	Ans
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Place Value	128	64	32	16	8	4	2	1	
a.	0	1	1	1	0	1	0	1	117
b.	1	0	1	1	1	0	1	1	187
c.	0	1	0	0	0	1	0	0	68
d.	1	0	1	0	0	0	0	0	160
e.	0	0	0	0	1	1	1	1	15

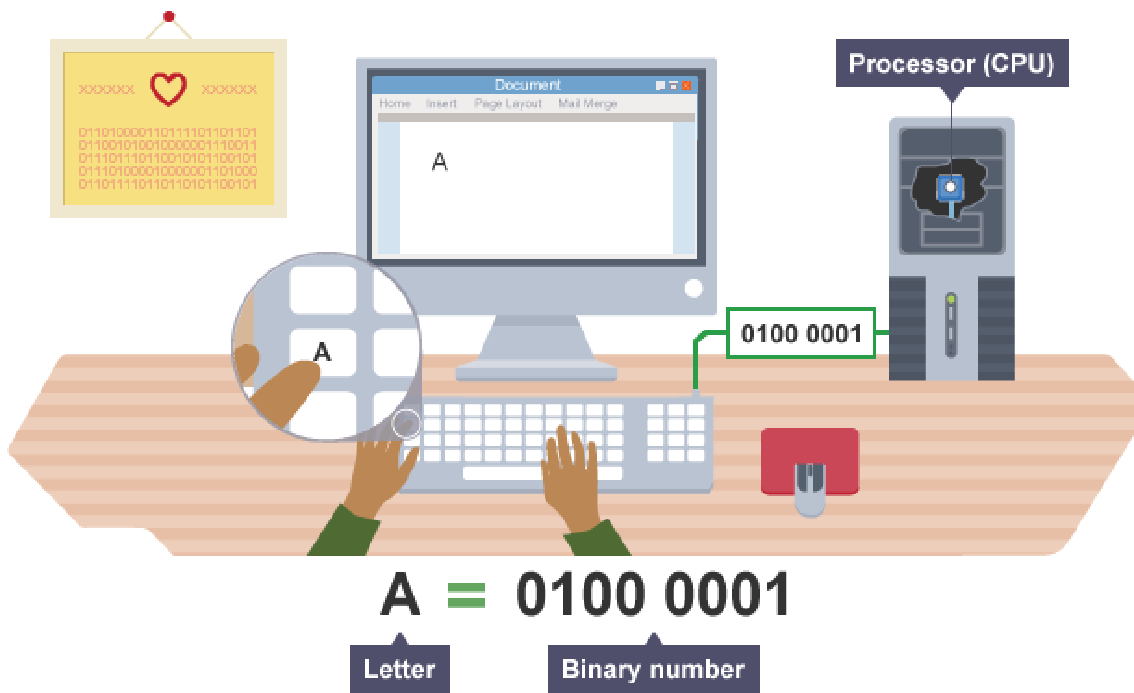
Exercise # 2

If a Byte is assumed to represent a number, as described earlier, then what is the range (minimum and maximum values) of numbers that can be stored in a 8-bit Byte?

- Min : 0
- Max : 128+64+32+16+8+4+2+1 = 255

How can we represent a character?

- IDEA.
  - Assign numeric codes to characters and represent each character in a Byte using it’s numeric code.
  - Can we assign numeric codes of our choice to each character?. What might be a problem with this approach?



How can we represent a character?

- **IDEA**
- Create a Standard coding scheme so that information can be easily shared between devices from different vendors.
- Standard Codes
  - ASCII (American Standard Code for Information Interchange)
  - Unicode
  - Unicode Transformation Format(UTF) UTF-8, UTF-16
  - ANSI Character Set

ASCII Character Encoding

Letter | Number | Punctuation | Symbol | Other | undefined

ASCII (1977/1986)

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	NUL 0000 0	SOH 0001 1	STX 0002 2	ETX 0003 3	EOT 0004 4	ENQ 0005 5	ACK 0006 6	BEL 0007 7	BS 0008 8	HT 0009 9	LF 000A 10	VT 000B 11	FF 000C 12	CR 000D 13	SO 000E 14	SI 000F 15
1_	DLE 0010 16	DC1 0011 17	DC2 0012 18	DC3 0013 19	DC4 0014 20	NAK 0015 21	SYN 0016 22	ETB 0017 23	CAN 0018 24	EM 0019 25	SUB 001A 26	ESC 001B 27	FS 001C 28	GS 001D 29	RS 001E 30	US 001F 31
2_	SP 0020 32	! 0021 33	" 0022 34	# 0023 35	$ 0024 36	% 0025 37	& 0026 38	' 0027 39	(0028 40	) 0029 41	* 002A 42	+ 002B 43	, 002C 44	- 002D 45	. 002E 46	/ 002F 47
3_	0 0030 48	1 0031 49	2 0032 50	3 0033 51	4 0034 52	5 0035 53	6 0036 54	7 0037 55	8 0038 56	9 0039 57	: 003A 58	; 003B 59	< 003C 60	= 003D 61	> 003E 62	? 003F 63
4_	@ 0040 64	A 0041 65	B 0042 66	C 0043 67	D 0044 68	E 0045 69	F 0046 70	G 0047 71	H 0048 72	I 0049 73	J 004A 74	K 004B 75	L 004C 76	M 004D 77	N 004E 78	O 004F 79
5_	P 0050 80	Q 0051 81	R 0052 82	S 0053 83	T 0054 84	U 0055 85	V 0056 86	W 0057 87	X 0058 88	Y 0059 89	Z 005A 90	[005B 91	\ 005C 92	] 005D 93	^ 005E 94	_ 005F 95
6_	` 0060 96	a 0061 97	b 0062 98	c 0063 99	d 0064 100	e 0065 101	f 0066 102	g 0067 103	h 0068 104	i 0069 105	j 006A 106	k 006B 107	l 006C 108	m 006D 109	n 006E 110	o 006F 111
7_	p 0070 112	q 0071 113	r 0072 114	s 0073 115	t 0074 116	u 0075 117	v 0076 118	w 0077 119	x 0078 120	y 0079 121	z 007A 122	{ 007B 123	| 007C 124	} 007D 125	~ 007E 126	DEL 007F 127

Exercise # 4

- The following 24 values represents a message consisting of 24 characters stored in RAM.
- If the message has been written using 8-Bit Extended ASCII codes then decipher the message

87 104 97 84 32 105 83 32 89 111 117 82 32 70 105 82 83 84 32 78 65 77 69 63

WhaT iS YouR FiRST NAME?

Exercise # 5

- Use ASCII encoding to give answer to the question in the previous Exercise?

ASCII to Binary 0s and 1s

Converting the text “hope” into binary

Characters:	h	o	p	e
ASCII Values:	104	111	112	101
Binary Values:	01101000	01101111	01110000	01100101
Bits:	8	8	8	8

Representing Non-Negative(Unsigned) Integer Values

- Idea No 1.
  - Each integer is a sequence of characters and hence we can use character encoding to represent each quantity as a sequence of characters.
  - 371 = 51 55 49 (ASCII)

Exercise # 5

- Represent the following integer quantities as sequence of bytes encoded using ASCII characters.

Integer	ASCII REPRESENTATION
20456	50 48 52 53 54
196	49 57 54
1024	49 48 50 52
32	51 50
100015	49 48 48 48 49 53

Representing Non-Negative(Unsigned) Integer Values

- Idea No 2.
  - Integer quantities can be represented using the idea of place value using binary number system. That is each bit has a place value and total value stored is sum of all the place values included in the number.

Representing Non-Negative(Unsigned) Integer Values

- Problem
  - A byte has only eight bits and hence we can not represent quantities bigger than 255 in a byte. For processing integer quantities this is an unacceptably low value.

- Solution
  - Use 2 or more bytes to store an integer quantity

Byte 1								Byte 0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Exercise # 6

- Use 2-Bytes to represent each of the following quantity

Integer	2 Byte Representation
20456	01001111 11101000
196	00000000 11000100
1024	00000100 00000000
32	00000000 0100000
100015	?

What is the Maximum unsigned integer value that can be represented using 2 Bytes? $(2^{16} - 1) = 65535$

Exercise # 7

- Use 4-Bytes to represent each of the following quantity

Integer	4 Byte Representation
20456	00000000 00000000 01001111 11101000
196	00000000 00000000 00000000 11000100
1024	00000000 00000000 00000100 00000000
32	00000000 00000000 00000000 0100000
100015	00000000 00000001 10000110 10101111

- What is the Maximum unsigned integer value that can be represented using 4 Bytes? $(2^{32} - 1) = 4294967295$

Representing Signed Integer Values

- Problem
  - How can we represent Signed (Both negative and positive) numbers?

- Solution
  - FIX ONE OF THE BIT FOR REPRESENTING SIGN (Sign-Magnitude method)
    - $(01101101)_2 = +(109)_{10}$
    - $(11101101)_2 = -(109)_{10}$
    - $(00101011)_2 = +(43)_{10}$
    - $(10101011)_2 = -(43)_{10}$

Exercise

- Represent each of the following quantity in 1-bytes using sign-magnitude method.

Integer	1 Byte Representation
65536	??
-64	11000000
64	01000000
-110	10010010

- What is the Maximum signed integer value that can be represented using 1 Byte? $(2^{8-1} -1) = 127$ For 2 Bytes $(2^{16-1} -1) = 32767$

Representing Signed Integer Values

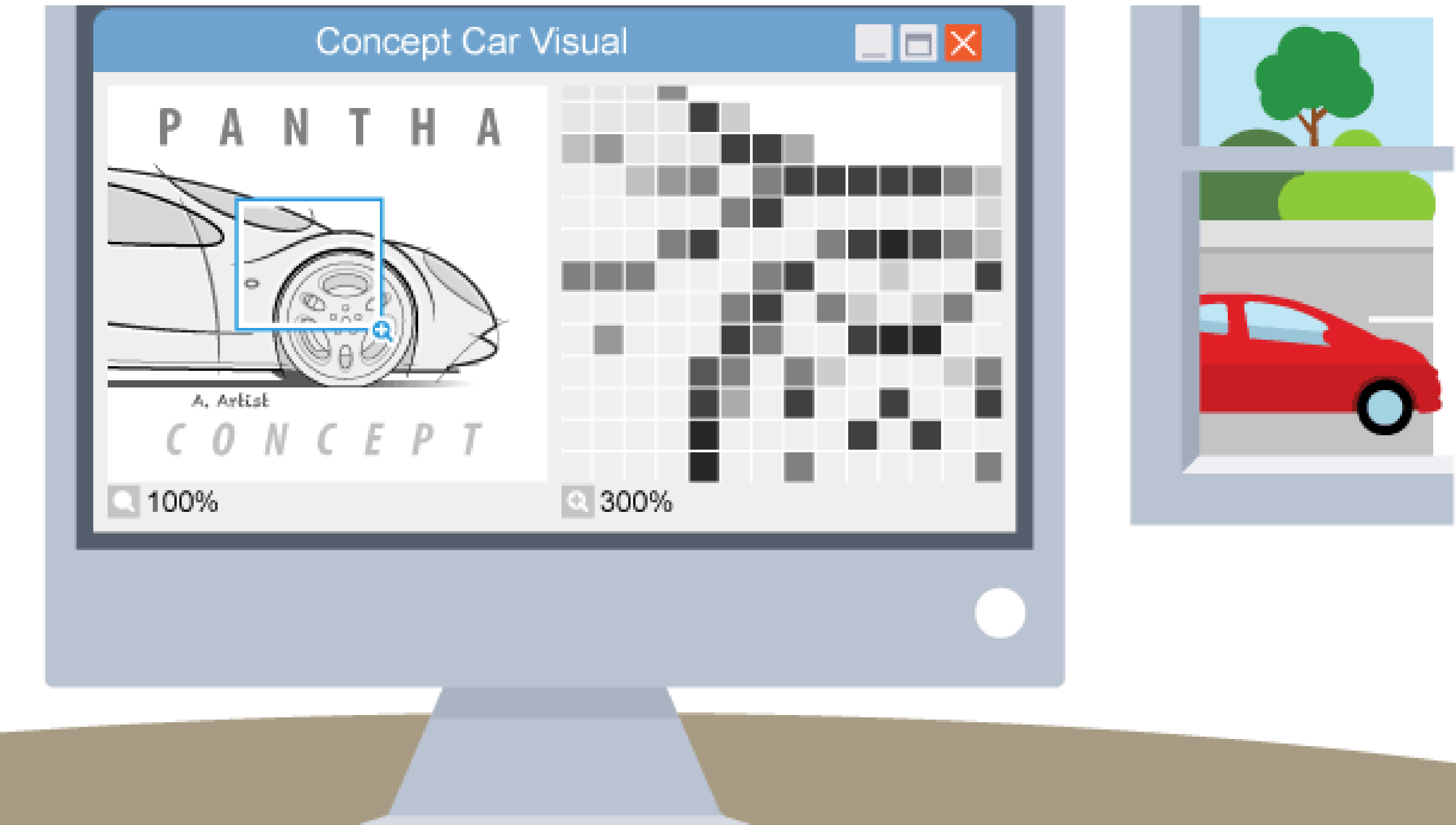
- Problem
  – How can we represent Signed (Both negative and positive) numbers?
- Use 2’s Complement Representation
  – **1’s complement** of a binary number is another binary number obtained by toggling all bits in it, i.e., transforming the 0 bit to 1 and the 1 bit to 0
    - 1's complement of "0111" is "1000"
    - 1's complement of "1100" is "0011"
  – **2’s complement** of a binary number is 1 added to the 1’s complement of the binary number.
    - 2's complement of "0111" is "1001"
      2's complement of "1100" is "0100"

Exercise

- Represent each of the following quantity in 2's Complement method.

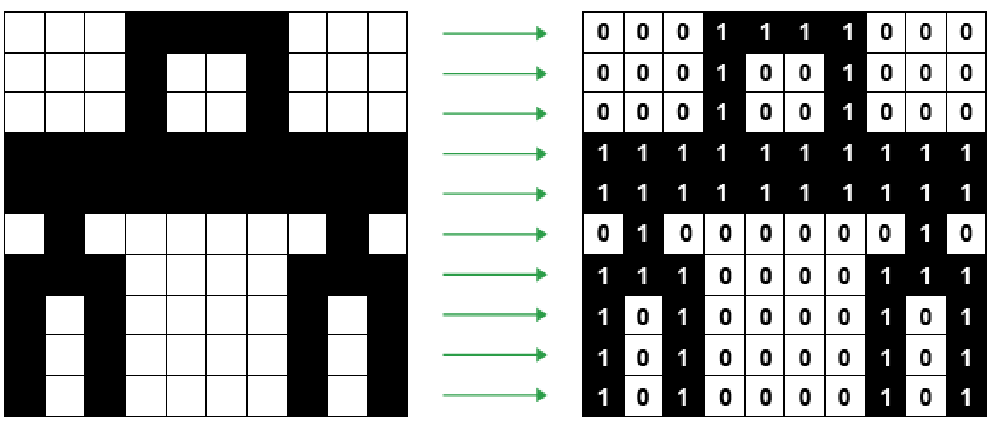
Integer	1 Byte Representation
12	00001100
-12	11110100
-53	11001011
30	00011110
-110	10010010

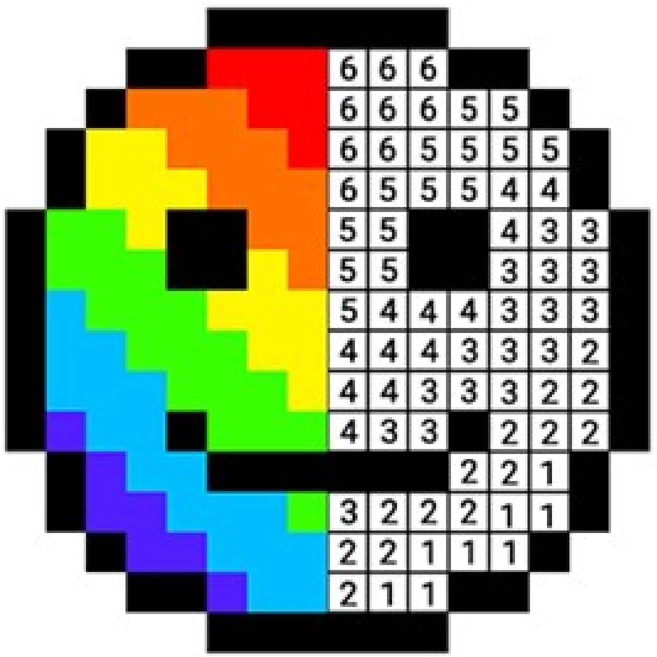
Image Representation



Representing Images in Binary

- Graphics on a screen are made up of tiny blocks called pixels.
- Each color of an image is stored as a binary number.



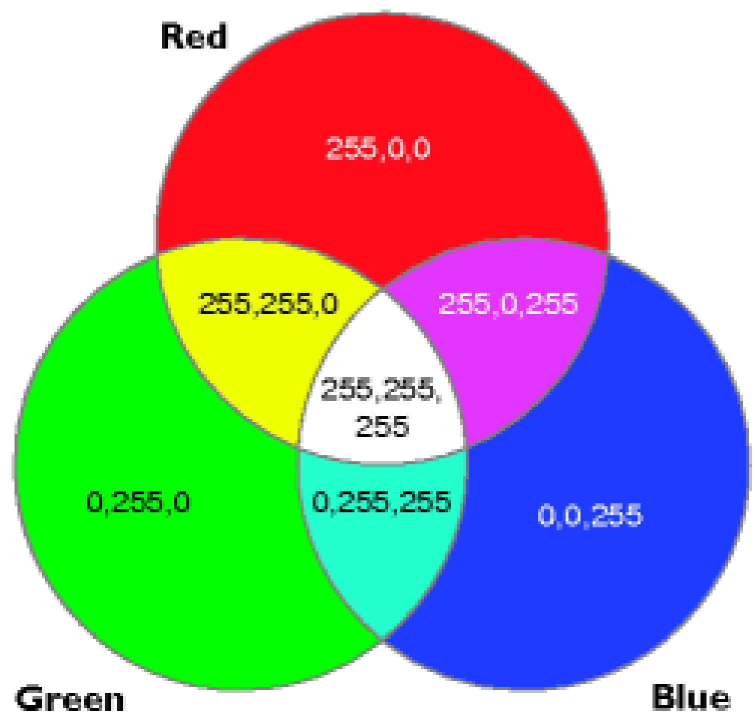


RGB

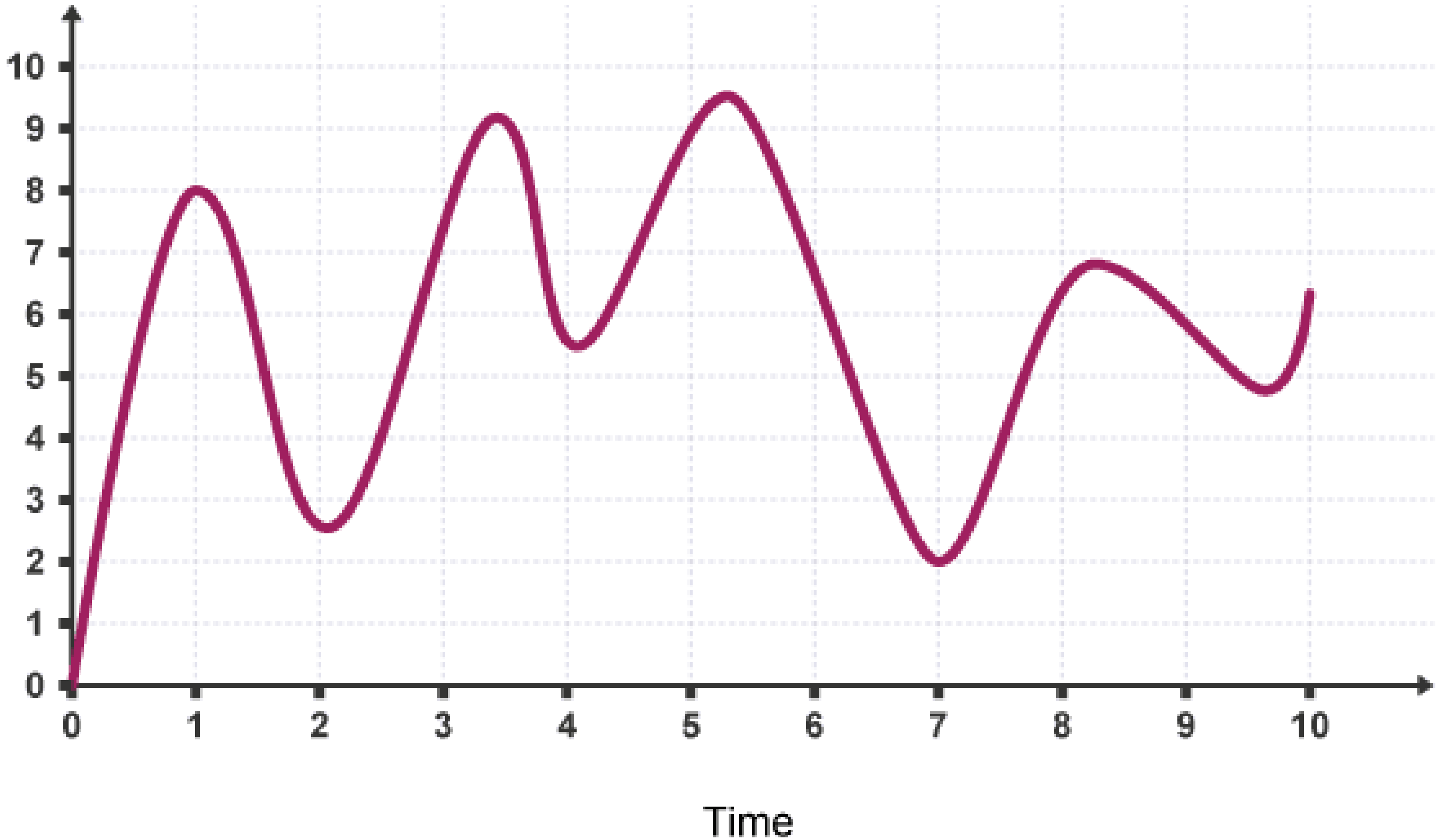
- 1 bit per pixel (0 or 1): two possible colors
- 2 bits per pixel (00 to 11): four possible colors
- 3 bits per pixel (000 to 111): eight possible colors
- 4 bits per pixel (0000 – 1111): 16 possible colors

0-255	0-255	0-255

16 Million Distinct Colors

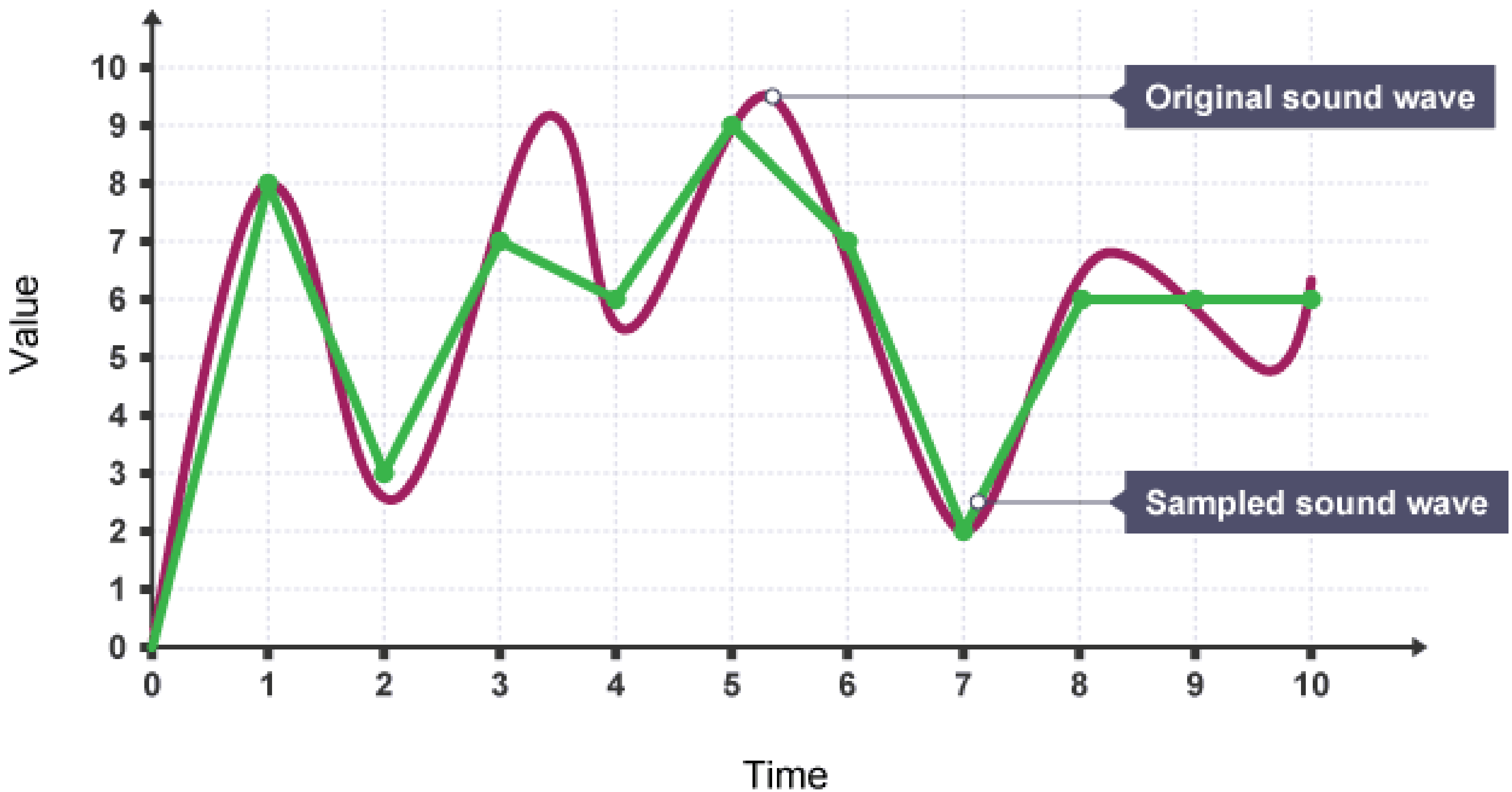


Sound Representation



Sound Representation

Time sample	1	2	3	4	5	6	7	8	9	10
Denary	8	3	7	6	9	7	2	6	6	6
Binary	1000	0011	0111	0110	1001	0111	0010	0100	0110	0110



Number Systems

Decimal, Binary, Hexa-Decimal

Common Number Systems

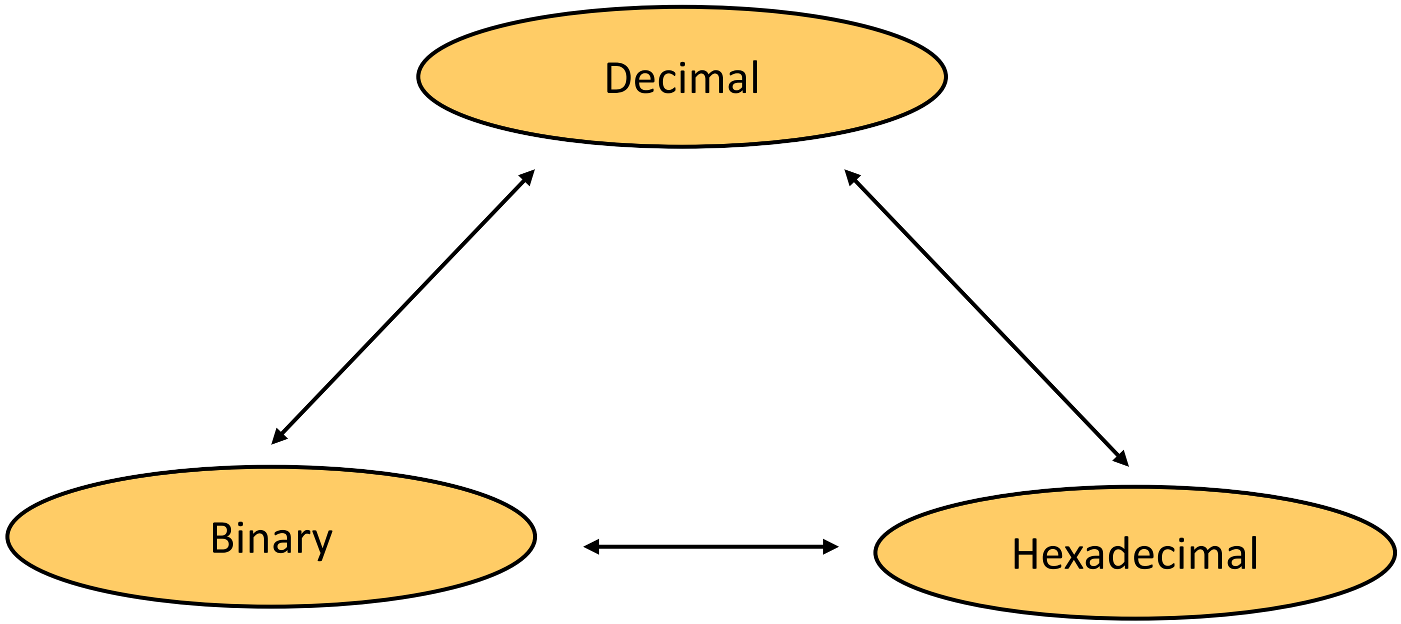
System	Base	Symbols	Used by humans?	Used in computers?
Decimal	10	0, 1, … 9	Yes	No
Binary	2	0, 1	No	Yes
Hexa-decimal	16	0, 1, … 9, A, B, … F	No	No

Conversion Chart

Decimal	Binary	Hex
00	0000	0
01	0001	1
02	0010	2
03	0011	3
04	0100	4
05	0101	5
06	0110	6
07	0111	7
08	1000	8
09	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Conversion Among Bases

- The possibilities:



Quick Example

$$25_{10} = 11001_2 = 19_{16}$$

Base

Conversions

- Decimal to Binary
- Decimal to Hexadecimal

- Binary to Decimal
- Binary to Hexadecimal

- Hexadecimal to Decimal
- Hexadecimal to Binary

Decimal to Binary

- Technique
  - Divide by two, keep track of the remainder

$125_{10} = ?_2$

2	125	
2	62	1
2	31	0
2	15	1
2	7	1
2	3	1
2	1	1
	0	1

$125_{10} = 1111101_2$

Conversions

- Decimal to Binary
- Decimal to Hexadecimal

- Binary to Decimal
- Binary to Hexadecimal

- Hexadecimal to Decimal
- Hexadecimal to Binary

Decimal to Hexadecimal

- Numbers having base 16
- Possible digits are

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

- Technique
  - Divide by 16
  - Keep track of the remainder

Example: Decimal to Hexadecimal

$1234_{10} = ?_{16}$

16	1234	
16	77	2
16	4	13 = D
	0	4

$1234_{10} = 4D2_{16}$

Conversions

- Decimal to Binary
- Decimal to Hexadecimal

- Binary to Decimal
- Binary to Hexadecimal

- Hexadecimal to Decimal
- Hexadecimal to Binary

Binary to Decimal

- Technique
  - Multiply each bit by 2^n, where n is the "weight" of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Example: Binary to Decimal

Bit "0"

$$101011_2 \Rightarrow \begin{array}{lcl} 1 \times 2^0 &=& 1 \\ 1 \times 2^1 &=& 2 \\ 0 \times 2^2 &=& 0 \\ 1 \times 2^3 &=& 8 \\ 0 \times 2^4 &=& 0 \\ 1 \times 2^5 &=& 32 \\ \hline & & 43_{10} \end{array}$$

2^5	2^4	2^3	2^2	2^1	2^0	
32	16	8	4	2	1	
1	0	1	0	1	1	32+8+2+1 = 43

Conversions

- Decimal to Binary
- Decimal to Hexadecimal

- Binary to Decimal
- Binary to Hexadecimal

- Hexadecimal to Decimal
- Hexadecimal to Binary

Binary to Hexadecimal

- Technique
  - Group bits in fours, starting on right
  - Convert to hexadecimal digits

$1010111011_2 = ?_{16}$

```
10  1011  1011
↓    ↓     ↓
2    B     B
```

$1010111011_2 = 2BB_{16}$

Conversions

- Decimal to Binary
- Decimal to Hexadecimal

- Binary to Decimal
- Binary to Hexadecimal

- Hexadecimal to Decimal
- Hexadecimal to Binary

Hexadecimal to Decimal

- Technique
  - Multiply each bit by 16^n, where n is the “weight” of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Example

$ABC_{16} = ?_{10}$

$$
\begin{array}{lrrrrrr}
ABC_{16} \Rightarrow & C \times 16^0 & = & 12 \times & 1 & = & 12 \\
 & B \times 16^1 & = & 11 \times & 16 & = & 176 \\
 & A \times 16^2 & = & 10 \times & 256 & = & 2560 \\
\hline
 & & & & & & 2748_{10}
\end{array}
$$

16^2	16^1	16^0	
256	16	1	
A	B	C	
256X10	16X11	12X1	2560+176+12=2748

Conversions

- Decimal to Binary
- Decimal to Hexadecimal

- Binary to Decimal
- Binary to Hexadecimal

- Hexadecimal to Decimal
- Hexadecimal to Binary

Hexadecimal to Binary

- Technique
  - Convert each hexadecimal digit to a 4-bit equivalent binary representation

$10AF_{16} = ?_2$

1	0	A	F
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
0001	0000	1010	1111

$10AF_{16} = 0001000010101111_2$

Exercise

Decimal	Binary	Hexa-decimal
33		
	1110101	
		1AF

Answer

Decimal	Binary	Hexa-decimal
33	100001	21
117	1110101	75
431	110101111	1AF

Activity

	Information in File 1	Coded Information in File 2
1	Your NUCES ID	Coded using ASCII Codes (one eight bit code for each ASCII character)
2	Your Full Name	Name Coded using ASCII Codes (Do remember to code the blank space as well)
3	Father's Name	Coded using ASCII Codes
4	Marks in F.Sc.	Coded using 16 bit unsigned code
5	Your favorite Color from the following list. i) Pure Red, ii) Pure Green iii) Pure Blue	Coded as 24 bit RGB color space using 8-bit for each of the R, G and B components

Activity

Some information has been coded in the ten bytes given above. Decode the information in each of the following cases

- Case 1: if these bytes contain characters coded using extended-ASCII or ANSI coding standard
- Case 2: if the bytes contain 5 integer values each stored using two byte unsigned integer representation.
- Case 3: if the bytes contain 5 integer values each stored using two byte signed integers represented using sign-magnitude method.

10010011 01001101 01000101 01010011 01010011 01000001 01000111 01000101 10010100 00101110

Homework

- How to represent Real Numbers?
  - Fixed point representation.
  - Floating Point Representation.

Lets Play a Game

- https://games.penjee.com/binary-numbers-game/

Recommended

- https://www.youtube.com/watch?v=1GSjbWt0c9M
- https://www.khanacademy.org/computing/computer-science/how-computers-work2/v/khan-academy-and-codeorg-introducing-how-computers-work
- https://www.youtube.com/watch?v=ptzGI9VaZmQ


SET INDEX 1,472.69
High -7.46 Low 1,465.66 -49.78 Vol(K) 8,274,659
FILE
SMS
DATA Report
www

Thank You