DATA REPRESENTATION

KNOWLEDGE ORGANISER: 1/5

Fulwood academy

TOPIC TERMINOLOGY

Tier 2 Vocab:	Tier 3 Vocab:
Logic, gates, input, output, data, unit, file size, calculate, storage, binary, system, convert.	Bit, Byte, Kilobyte, Megabyte, Gigabyte, terabyte, petabyte, binary, pixels, CPU denary, hexadecimal.

CALL AND RESPONSE

I SAY...	YOU SAY...
1 is ... 0 is ...	ON OFF
Boolean order of Precedence is...	NOT AND OR
The smallest unit of data is...	1 or 0 (bit)
One nibble is...	4 bits
One byte is...	8 bits
One's Complement is...	Flip the digits
Two's Complement is...	Add 1

BINARY NUMBERS

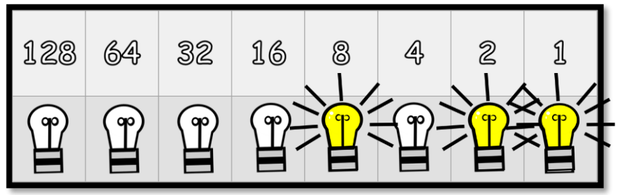
1's used to represent "ON"
0's used to represent "OFF"
BINARY IS A BASE 2 NUMBER SYSTEM

DID YOU KNOW!

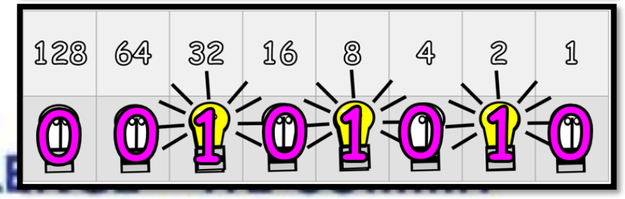
Computers can only use binary to communicate. This means that to send a signal from one part of the computer to another, they have to send either a number 1 or 0.

Binary is just lots of 1s and 0s – they can be as small as 1 digit long to an infinite amount of 1s and 0s

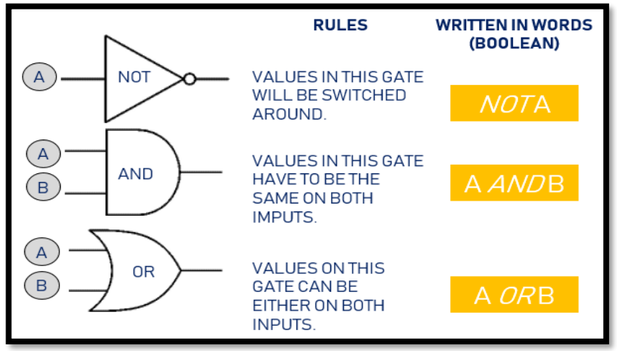
BINARY NUMBERS EXPLAINED IN PICTURES



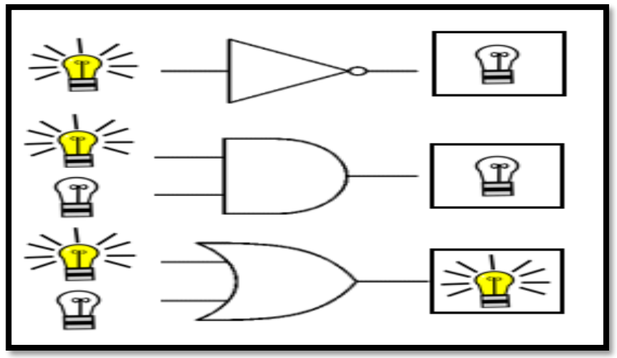
Computers don't use lightbulbs to represent numbers. They use a series of 0's (offs) 1's (ons). When using the binary calculator. Wherever you have a lightbulb that is switched ON, that must be replaced with a 1. Lightbulbs that are switched OFF, must be replaced with a 0.



LOGIC GATES



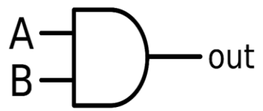
LOGIC GATE RULES EXPLAINED IN PICTURES



LOGIC GATE TRUTH TABLES

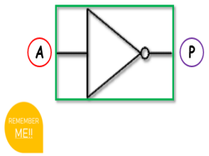
REMEMBER ME!!

The output of a logic gate can also be summarised in the form of a table, called a 'Truth Table'. A truth table shows all the possible combinations a gate can have depending on its number of inputs.



AND	
Inputs	Output
B	A	Q
0	0	
0	1	
1	0	
1	1	

BOOLEAN LOGIC



REMEMBER ME!!

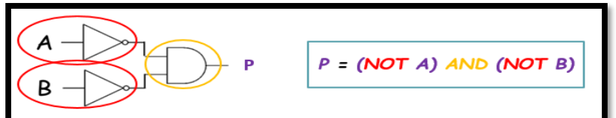
For a single input **NOT gate**, the output P is ONLY true when the input is "**NOT**" true, the output is the inverse or complement of the input giving the **Boolean Expression** of: (P = NOT A)

For a two input **AND gate**, the output P is ONLY true when the input A "**AND**" B are true, giving the **Boolean Expression** of: (P = A AND B)

Look at the first example:

We always start with the **output =** when writing an expression. In this case output is labelled ***P. Then we follow the order of precedence. (See Below) And normally writ them out in alphabetical order.***

We also **use brackets** – **brackets** are normally where a gate is directly connected to an input (AB) but they specify which operation should be performed first.



Order of Precedence

1. NOT
2. AND
3. OR

KWE>2022-23

KNOWLEDGE ORGANISER: 2/5

UNITS OF DATA

REMEMBER ME!!

8 BITS: 01011011 = 1 BYTE (ENOUGH SPACE FOR A SINGLE LETTER OR SYMBOL)

TERABYTE
GIGABYTE
MEGABYTE
KILOBYTE

4 BITS = 1 NIBBLE

1 BIT

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1

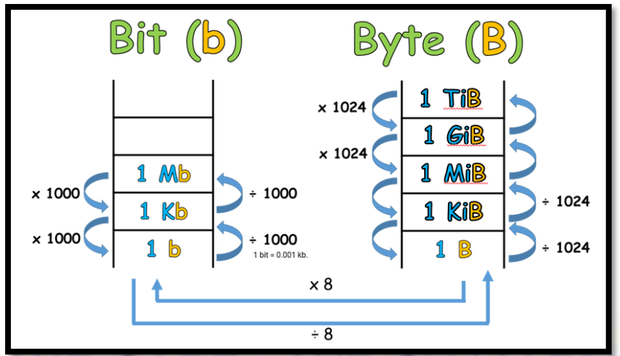
The more bits we have available the bigger the number we can convert from denary to binary (in one go)

BINARY PIZZA

We can use the pizza analogy to recall:

1 Slice is a bit
4 slices is a nibble
8 slices is a byte

BITS AND BYTES



BINARY MATH

The different techniques of calculating binary digits. The CPU and the ALU register must calculate binary in many way in order to decode and execute instructions.

ADDITION

A + B	SUM	CARRY
0 + 0	0	0
0 + 1	1	0
1 + 0	1	0
1 + 1	0	1
1 + 1 + c1	1	1

ADDITION EXAMPLE

	8	4	2	1		
	1	1				
	0	1	1	0	6	
	0	1	1	0	6	+
	1	1	0	0	12	

BINARY MATH 2

REMEMBER ME!!

Computers can't count backwards and therefore a computer has to convert positive numbers into negative numbers to work out subtractions.

REMEMBER ME!!

Binary numbers can be represented in two ways...
i.e. signed and unsigned.

- Signed numbers use a sign flag to distinguish between negative values and positive values
- Unsigned numbers store only positive numbers but not negative numbers.

Signed

Notice that the most significant bit (MSB) – the left most bit, now represents a negative number.
We use Signed numbers within Two's complement.

MSB LSB

-128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0

NEGATIVE NUMBERS

REMEMBER ME!!

Binary: 1's Complement

1's Complement is obtained by inverting all the bits in the binary representation of the number (Swapping 0's for 1's and vice versa). This allows the binary number to behave like a **negative** number when used with **2's complement**.

REMEMBER ME!!

Binary: 2's Complement

i. **Step 1: Find 1's Complement**
ii. **Step 2: Add 1 to the 1's Complement (First column only)**

BINARY MATH 3

SUBTRACTION EXAMPLE PT1

15(-13)

1 Original (Number)

8	4	2	1
1	1	0	1

13

2 1's Complement

8	4	2	1
0	0	1	0

3 2's Complement

-8	4	2	1	
0	0	1	0	
			1	+
0	0	1	1	

SUBTRACTION EXAMPLE PT2

	8	4	2	1		
	1	1	1			
+15	1	1	1	1		
-13	0	0	1	1		+
	0	0	1	0	2	

KNOWLEDGE ORGANISER: 3/5

HEXADECIMAL

REMEMBER ME!!

Hexadecimal:
Hexadecimal (or hex) is a base 16 system used to simplify how binary is represented. A hex digit can be any of the following 16 digits: 0 1 2 3 4 5 6 7 8 9 A B C D E F.
Each hex digit reflects a 4-bit binary sequence.

Using hexadecimal:
Hex codes are used in many areas of computing to simplify binary codes. It is important to note that computers do not use hexadecimal – it is used by humans to shorten binary to a more easily understandable form to reduce human error.

HEX REPRESENTION

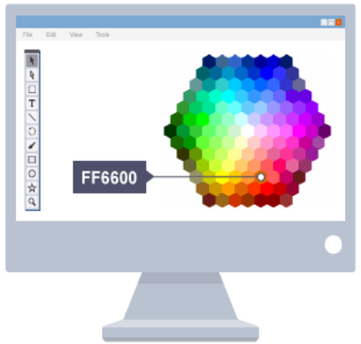
Denary	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

USING HEXADECIMAL

Colours

Hex can be used to represent **colours** on web pages and image-editing programs using the format **#RRGGBB** (RR = reds, GG = greens, BB = blues). The # symbol indicates that the number has been written in hex format.

This system uses two hex digits for each colour, eg #FF6600.



1 1 1 1 = F (R)
1 1 1 1 = F
0 1 1 0 = 6 (G)
0 1 1 0 = 6
0 0 0 0 = 0 (B)
0 0 0 0 = 0

As one hex digit represents 4 **bits**, two hex digits together make 8 bits (1 **byte**). The values for each colour run between 00 and FF. In binary, 00 is 0000 0000 and FF is 1111 1111. That provides 256 possible values for each of the three colours.

CONVERSION HEX / BINARY / DENARY

Denary: 74
= (Binary): 0 1 0 0 1 0 1 0
Conversion: Binary value split into groups of 4-digits (Starting from the right)

0100 | 1010

=
4 — 10 (A in Hex)

= (Hex): 4A

CONVERSION HEX / BINARY / DENARY

Insert Binary below							
Binary =							
128	64	32	16	8	4	2	1
0	0	0	0	1	0	1	0

Insert Hexadecimal Number below							
HEX =					HEX =		
8	4	2	1		8	4	2	1
0	0	0	0		1	0	1	0
0					A			
#0A								

COLOUR REPRESENTAION USING HEX

R G B colour values are used in all web browsers to represent colours on screen.

An R G B colour value is specified with:

R G B (Red, Green, Blue).

Each parameter (Red, Green, and Blue) defines the intensity of the colour as a denary number between 0 and 255.

	R	G	B
Denary	0	0	255
Binary	00000000	00000000	11111111
Hexadecimal	0 0	0 0	F F

EXAMPLE

Remember each number should be converted separately in HEX so 87 is 8 and 7

#873260

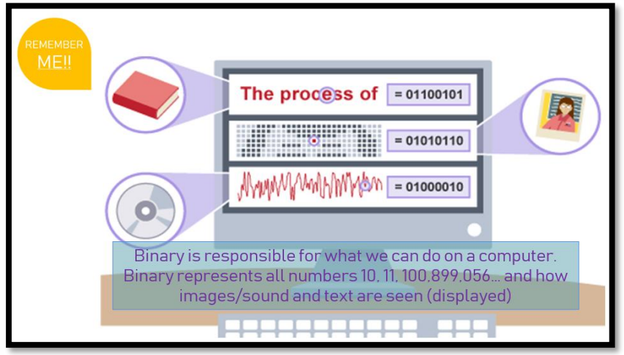
RGB	HEX	BIN	
R	8	1000	RED Binary
	7	0111	10000111
G	3	0011	GREEN Binary
	2	0010	0011 0010
B	6	?	BLUE Binary
	0	?	?

DATA REPRESENTATION

KNOWLEDGE ORGANISER: 4/5

Fulwood academy

CHARACTER, IMAGE AND SOUND REPRESENTATION



CHARACTERS USE ASCII

Computers use a character set called ASCII (American standard code information interchange)

REMEMBER ME!!

A Character Set IS NOT A FONT!

Den	Hex	Char	Den	Hex	Char	Den	Hex	Char	Den	Hex	Char
0	0		32	20	[space]	64	40	@	96	60	`
1	1		33	21	!	65	41	A	97	61	a
2	2		34	22	"	66	42	B	98	62	b
3	3		35	23	#	67	43	C	99	63	c
4	4		36	24	$	68	44	D	100	64	d
5	5		37	25	%	69	45	E	101	65	e
6	6		38	26	&	70	46	F	102	66	f
7	7		39	27	'	71	47	G	103	67	g
8	8		40	28	(	72	48	H	104	68	h
9	9		41	29	)	73	49	I	105	69	i
10	A		42	2A	*	74	4A	J	106	6A	j
11	B		43	2B	+	75	4B	K	107	6B	k
12	C		44	2C	,	76	4C	L	108	6C	l
13	D		45	2D	-	77	4D	M	109	6D	m
14	E		46	2E	.	78	4E	N	110	6E	n
15	F		47	2F	/	79	4F	O	111	6F	o
16	10		48	30	0	80	50	P	112	70	p
17	11		49	31	1	81	51	Q	113	71	q
18	12		50	32	2	82	52	R	114	72	r
19	13		51	33	3	83	53	S	115	73	s
20	14		52	34	4	84	54	T	116	74	t
21	15		53	35	5	85	55	U	117	75	u
22	16		54	36	6	86	56	V	118	76	v
23	17		55	37	7	87	57	W	119	77	w
24	18		56	38	8	88	58	X	120	78	x
25	19		57	39	9	89	59	Y	121	79	y
26	1A		58	3A	:	90	5A	Z	122	7A	z
27	1B		59	3B	;	91	5B	[	123	7B	{
28	1C		60	3C	<	92	5C	\	124	7C	|
29	1D		61	3D	=	93	5D	]	125	7D	}
30	1E		62	3E	>	94	5E	^	126	7E	~
31	1F		63	3F	?	95	5F	_	127	7F	[DEL]

CHARACTERS (ASCII)	
DENARY	CHARACTER IN BINARY
065	C = 0100 0011
079	O = 0100 1111
077	M = 0100 1101
080	P = 0101 0000
085	U = 0101 0101
084	T = 0101 0100
069	E = 0100 0101
082	R = 0101 0010
033	! = 00100 001

IMAGE REPRESENTATION

Just like numbers and characters - Images also need to be converted into binary data so that the CPU can process/understand it.
To make them viewable on a screen, digital images are made up of pixels.
Each pixel in an image is made up from binary data/numbers.

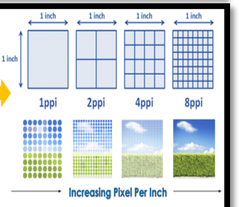
EXAMPLE:

If we say that 1 is black (or on) and 0 is white (or off), then a simple black and white digital image can be created using 1 bit of Binary data.

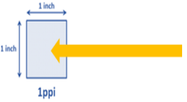
0	0	0	0	0	0	0	0	0
0	0	1	1	0	1	1	0	0
0	1	1	1	0	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	0	0
0	0	0	1	1	1	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0

Quality of images can be improved by:

Increasing pixels per inch



Colour depth (bit depth) available



The more binary numbers we can use in each pixel allows us to use more colours – thus improving the quality of the image

QUALITY OF IMAGES – BIT DEPTH

Bit Depth:

REMEMBER ME!!

How many colours can be created with different amounts binary bits...

How many colours do you think we can make with 3bits (2^3) or 4 bits (2^4)?

REMEMBER BINARY IS BASE 2?

2^1 (2*1) 1 0

2^2 (2*2) 00 01 10 11

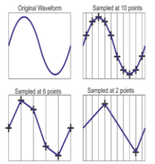
3 bits per pixel (000 to 111): eight possible colours
4 bits per pixel (0000 – 1111): 16 possible colours
...
16 bits per pixel (0000 0000 0000 0000 – 1111 1111 1111 1111): over 65 000 possible colours

SOUND REPRESENTATION

Sound is made up of waves and is measured in

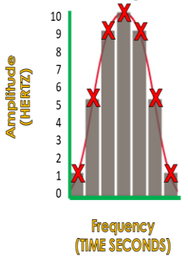
Amplitude

Frequency



Sound
Sound Waves

Every time a sample is taken from the sound wave, the amplitude is converted into Binary



0001 0101 1001 1010 1001 0101 0001

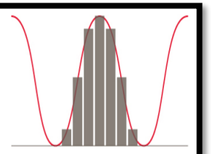
Each Sample here is 4bits

QUALITY OF SOUND – BIT DEPTH

Bit Depth:

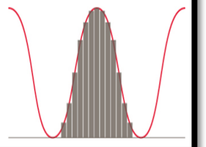
REMEMBER ME!!

Sound quality works similarly to how images improve their quality.



How do you think it is similar?

The more bits available to represent each sound sample taken – will produce more range to be used in digital sounds.
EXAMPLE BIT DEPTH : Use of 3bits (2^3) or 4 bits (2^4)



CALL AND RESPONSE

I SAY...	YOU SAY...
Characters, Images and sound all contain...	Binary Data
Images are made up from...	Pixels
Each pixel contains...	Binary data
Each sound sample contains...	Binary data
ASCII stands for...	American Standard Code Information Interchange
Bit depth is...	The number of binary bits used
Bit depth is calculated ...	2 power of n (2n)
ASCII is ...	NOT a font type, it is a character set

WE CARE • WE CHALLENGE • WE COMMIT

KNOWLEDGE ORGANISER: 5/5

META DATA

Metadata is the information stored within a file which helps the computer recreate the file when you want to view it on the screen.

For example: Metadata for an image usually includes, the image's file format (PSD, PNG, JPG), height and width of the image, the colour depth, and the resolution.

Without metadata, your device would not be able to (decode the instructions) to display the image on the screen as intended.

DATA COMPRESSION

REMEMBER ME!!

11111100000000010001000000111000111110000

Take the above binary number. Data like this can be compressed to make it smaller in size. In computing, data compression can be used to make file sizes smaller.

Smaller file sizes mean: Less storage space used and/or quicker download and upload times.

There are two types of compression
LOSSY AND LOSSLESS

BINARY DATA

REMEMBER ME!!

(b) Bit

Mb (Megabit)

Data SPEEDS are usually described in terms of BITS:

- 1,000 bits = 1 Kilobit
- 1,000 Kb = 1 Megabit
- 1,000 Mb = 1 Gigabit
- 1,000 Gb = 1 Terabit

(B) Byte

MB (MiBibyte)

Data STORAGE is usually described in terms of BYTES:

- 8 bits = 1 byte
- 1,024 Bytes = 1 Kilobyte
- 1,024 KB = 1 Megabyte
- 1,024 MB = 1 Gigabyte

CONVERTING DATA

In exams we are often asked to convert data from bits to bytes (vice versa). To do this, we need to use the conversion ladder (next image). We expected to write out the expression to perform the calculation.

EXAMPLE

REMEMBER ME!!

00111100 01000010 10000001 10100101 10100101

Take the above binary number. It means nothing... but if we know the height = 5 and the width = 8 and the number of bits per pixel is 1, then we can rearrange them to get:

0	0	1	1	1	1	0	0
0	1	0	0	0	0	1	0
1	0	0	0	0	0	0	1
1	0	1	0	0	1	0	1
1	0	1	0	0	1	0	1

COMPRESSION METHODS

CALCULATING BINARY DATA / FILE SIZES

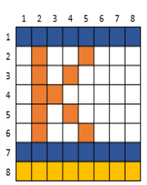
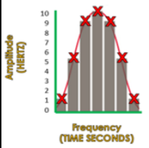


Image file size (bits) = Pixel H * Pixel W * Bit Depth (ANSWER IS IN BITS)

META DATA: HEIGHT 8px WIDTH 8px BIT DEPTH 2 bits

8 * 8 * 2 = 128 bits (FILE SIZE)

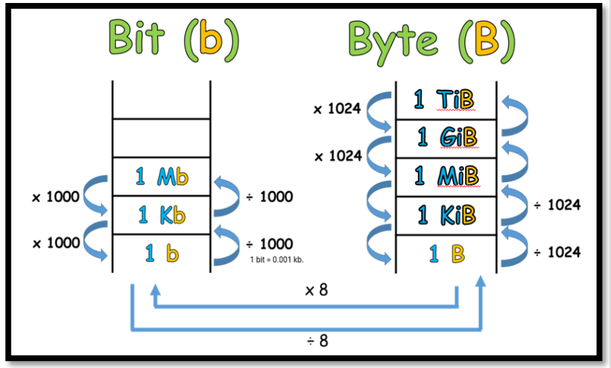


Sound file size = sample rate (Hz) x duration (seconds) x bit depth (bits)

META DATA: SAMPLE RATE 12000Hz DURATION 60Sec BIT DEPTH 4bits

12000 * 60 * 4 = 128 bits (FILE SIZE)

CONSTRUCTING EXPRESSIONS



Metadata ensures that computing devices can interpret what the binary data is intended to look/sound like. This is also important when compressing and extracting data within a data file.

REMEMBER ME!!

One compression type is RUN LENGTH ENCODING
It is a LOSSLESS compression type

The below binary data can be compressed in this way:

11111100000000010001000000111000111110000 (39bits)
618011301160313041 40 (20bits)

NOTE – RLE works best when there are repetitive values with a number like 1010101010 compressing it would make it larger this is known as negative compression

EXAMPLE: Calculate META DATA: HEIGHT 720px WIDTH 1080px BIT DEPTH 24 bits

- Step 1: HEIGHT * WIDTH
- Step 2: * BIT RATE.
- Step 3: Write out the expression

720 * 1080 * 24 = 18,662,400bits

(Answer always in bits)

- We want to convert from bits to MebiBytes.
- Step 1: 720 * 1080 * 24
- Step 2: From Bits to Bytes we divide by 8.
- Step 3: We need to go up the Byte ladder twice. So, we divide our number by 1024 power of 2, to give us MiB.

720 * 1080 * 24 / 8 / 1024 * 1024