OCR GCSE Computing – 2.1.4
(REPRESENTING DATA) – REVISION SHEET

REPRESENTING DATA IN COMPUTERS

Data can be stored in different format such as NUMBERS, TEXT, SOUNDS, IMAGES and PROGRAM COMMANDS

ELECTRIC CIRCUITS and BINARY

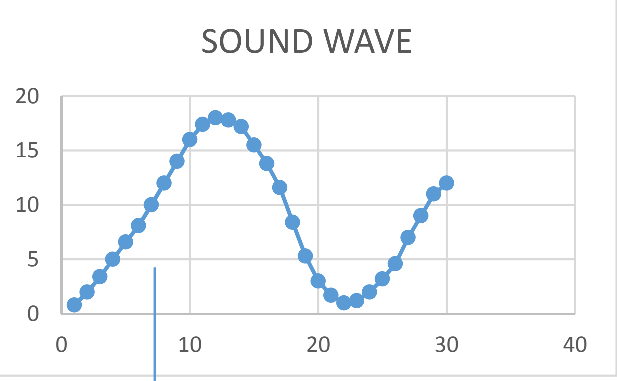
Computers consist of circuits that either allow electricity to flow (1) or not (0).

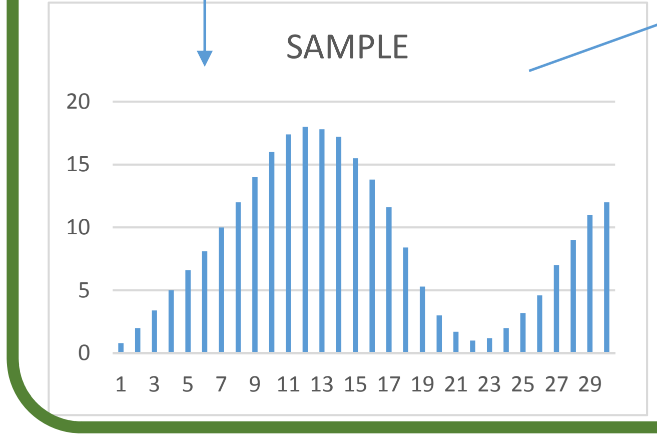
Single 1 or 0	***=***	***Binary Digit (BIT)***
4 BITS	***=***	***1 NIBBLE***
8 BITS	***=***	***1 BYTE***
1024 BYTES	***=***	***1 KILOBYTE***
1024 KILOBYTES	***=***	***1 MEGABYTE***
1024 MEGABYTES	***=***	***1 GIGABYTE***
1024 GIGABYTES	***=***	***1 TERABYTE***

TIP – LEARN BYTE / KILO / MEGA / GIGA / TERA and then remember that we multiply by 1024

REPRESENTING SOUND

SOUND is CONTINUALLY CHANGING so is SAMPLED at intervals called SAMPLE INTERVAL. Number of BITS used to store value is SAMPLE RESOULTION – The higher each of these is the BETTER QUALITY the sound is



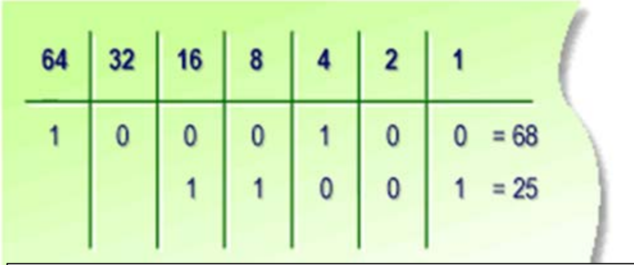


1	0.8
2	2
3	3.4
4	5
5	6.6
6	8.1
7	10
8	12
9	14
10	16
11	17.4
12	18

CONVERSIONS

Understand HOW to convert from

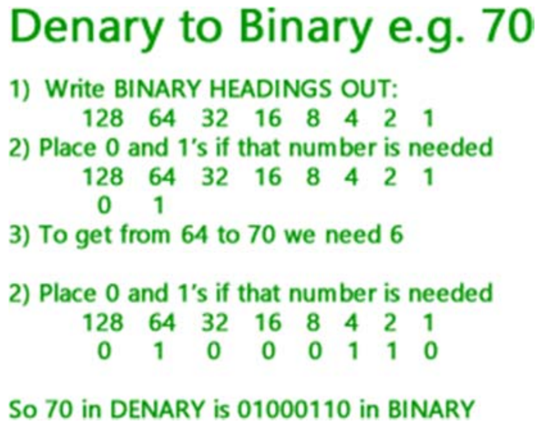
BINARY ⟺ DENARY (Normal eg 68)



64	32	16	8	4	2	1	
1	0	0	0	1	0	0	= 68
		1	1	0	0	1	= 25

1) Write out binary headings 128/64/32 etc

2) Add up the value of each DIGIT



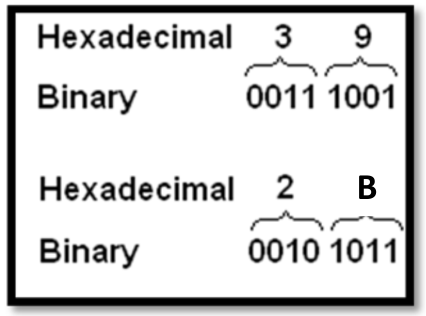
Denary to Binary e.g. 70

1) Write BINARY HEADINGS OUT:
128 64 32 16 8 4 2 1
2) Place 0 and 1's if that number is needed
128 64 32 16 8 4 2 1
0 1
3) To get from 64 to 70 we need 6

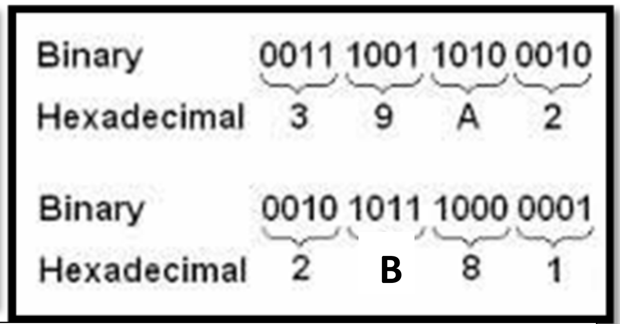
2) Place 0 and 1's if that number is needed
128 64 32 16 8 4 2 1
0 1 0 0 0 1 1 0

So 70 in DENARY is 01000110 in BINARY

BINARY ⟺ HEXADECIMAL



Hexadecimal	3	9
Binary	0011	1001
Hexadecimal	2	B
Binary	0010	1011



Binary	0011	1001	1010	0010
Hexadecimal	3	9	A	2
Binary	0010	1011	1000	0001
Hexadecimal	2	B	8	1

NIBBLE = 4 BINARY DIGITS

HEXADECMIAL uses 0,1,2,3,4,5,6,7,8,9, A, B, C, D,E, F where A=10, B=11 etc

DENARY to HEXADECIMAL
CONVERT DENARY to BINARY and then convert this to HEXADECIMAL

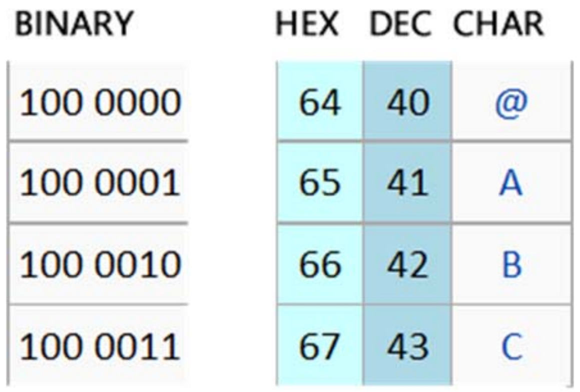
e.g. 70 = 0100 0110 in Binary which is 4 6 in HEXADECIMAL

HEXADECIMAL to DENARY
CONVERT HEXADECIMAL to BINARY and then BINARY to DECIMAL

e.g. 7E Binary = 0111 1110 in Binary which is 64 + 32 + 16 + 8 + 4 + 2 = 126 in denary

REPRESENTING CHARACTERS

KEY PRESSED -> Code transmitted to computer ->Code is stored as BINARY
PC's use ASCII codes
The characters that can be represented is called a character set

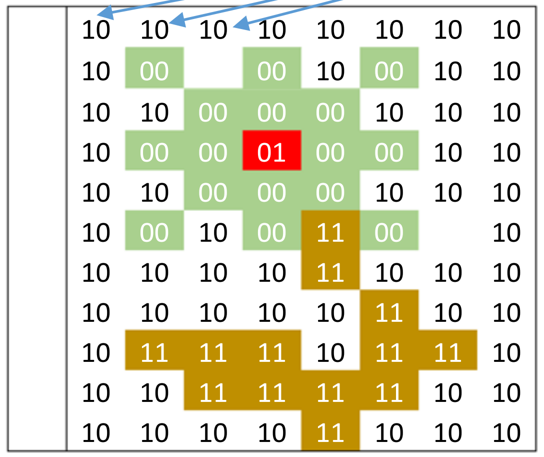


BINARY	HEX	DEC	CHAR
100 0000	64	40	@
100 0001	65	41	A
100 0010	66	42	B
100 0011	67	43	C

If we use 8 bits i.e. 0000 0000 to 11111111 we can represent up to 255 characters as 1111 1111 represents 255 in denary

REPRESENTING IMAGES

Making an IMAGE file:
Image uses four colours – To store we would save 10 10 10 etc…



	10	10	10	10	10	10	10	10
	10	00		00	10	00	10	10
	10	10	00	00	00	10	10	10
	10	00	00	01	00	00	10	10
	10	10	00	00	00	10	10	10
	10	00	10	00	11	00		10
	10	10	10	10	11	10	10	10
	10	10	10	10	10	11	10	10
	10	11	11	11	10	11	11	10
	10	10	11	11	11	11	10	10
	10	10	10	10	11	10	10	10

11	10	01	00
🟫	⬜	🟥	🟩

Using TWO BITS per pixel we can use 4 colours…

Using FOUR BITS per pixel we can use 15 colours

Number of bits used for each pixel is the colour depth

If the image is stored as 10 10 10 10 10 10 10 10 10 00…. Then to make the image the computer needs to know the COLOUR DEPTH, THE IMAGE HEIGHT and the IMAGE WIDTH