KS3 Knowledge Organiser

Binary, Denary, Sound, Images

The Binary Number System

In the number system that us humans use, the number 10 means 'ten' because the digits mean '1 ten and 0 ones'.

10's	1's
1	0

...the number 33 means 'thirty three' because the digits mean '3 tens and 3 ones'.

10's	1's
3	3

...the number 528 means 'five hundred and twenty eight' because the digits mean '5 hundreds, 2 tens and 8 ones'.

100's	10's	1's
5	2	8

Humans developed this 'base 10' number system millions of years ago because they learnt to count using their hands, which have 10 digits. Computers, being electronic, are simply made up of switches (which can be in only one of two states (on/off)) and as a result cannot count like us! As a result, computers use a different number system – the binary number system

The columns in the binary system, from right to left are 1, 2, 4, doubling as we go

...in binary the digits 10 means 'the number two' because the digits mean '1 two and 0 ones'.

2's	1's
1	0

...the digits 1001 means 'the number nine' because the digits mean '1 eight, 0 fours, 0 twos and 1 one'.

8's	4's	2's	1's
1	0	0	1

Converting from Binary to Decimal/Denary

To convert a binary number into decimal/denary, the process is thankfully really easy! All we need to do is add up the column values which contain a one and ignore the column values which contain a zero.

For example, the following binary number has the decimal/denary value of 155. This is because the 1s in the binary number represents **128 + 16 + 8 + 2 + 1 = 155**

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

➔ 155

Converting from Decimal/Denary to Binary

Converting from decimal/denary to binary, is also not too hard! We just need to work out which of the column values add together to form the decimal value that we needed to convert.

The easiest way is to do this is work from left to right along the binary column values and if the column value can fit into our decimal number, we place a 1 under that value's column, subtract the column value from the decimal number and continue the process. For example, if we want to convert the decimal number 202 into binary, we would do the following:

Does 128 fit into 202? Yes! So, we add a 1 to the column and subtract 128 from 202 leaving 74

128	64	32	16	8	4	2	1
1							

Does 64 fit into 74? Yes! So, we add a 1 to the column and subtract 64 from 74 leaving 10

128	64	32	16	8	4	2	1
1	1						

Does 32 fit into 10? No! So, we add a 0 to the column and move on!

128	64	32	16	8	4	2	1
1	1	0					

This process continues until we've made the remaining value of ten by placing 1s under the 8 and 2 columns (because 8 + 2 = 10), and zeros under the others.

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

Key Vocabulary

Key Word	*Definition*
Denary / Decimal	Base 10 number system - The number system we learnt in primary school
Binary	Base two number system – the only number system computers know – they can only understand two digits because they are made up of switches that can only be in the on (1) and off (0) state.
ASCII	Universally *accepted* binary numbers for each keyboard character
Bitmap	A computer image file which is made up of tiny pixels of colour. Each pixel is represented by a set of binary bits and mapped to the screen
Sampling	Recording analogue sound at regular intervals and converting each snippet of sound to a binary value.
Digital Processing	Applying maths on the binary which represents sound in order to manipulate how it sounds.

Binary Addition

Adding binary numbers is much like adding denary numbers (the system we use as humans).

Things to remember:

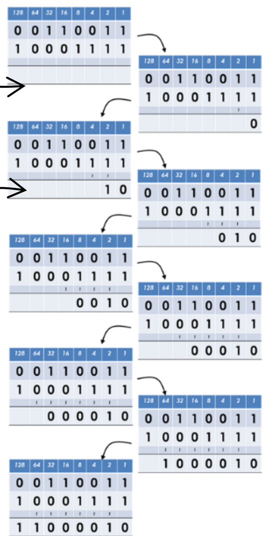
➔ Keep your numbers in the correct columns
➔ 1+1 = 10 in binary
➔ 1+1+1 = 11 in binary

We begin, like we would normally when adding two numbers together, with the right most column.

In this example, we add two 1s together, which of course is 2! However, in binary, 2 is represented as 10, and as there are two digits in this answer, we place the right digit (zero) under the column and carry the left digit (one) over to the next (left) column.

Now we focus on the second column from the right. Here we have 1+1+1 (including the carry), which of course equals 3. But in binary 3 is represented as 11. We therefore place the right digit underneath and carry the left digit.

This process then continues, moving through the columns to the left, until we have added the binary numbers.



Representing Characters

- As we know, computers can only deal with 0s and 1s (binary).
- All data that it needs to work with (numbers, sound, images etc) must be converted into binary for the computer to be able to process it.
- It is exactly the same for text, or one piece of text known as a character.
- Each time you hit a key on a keyboard, the computer generates a code for that letter, which is then processed by the CPU and the result might be the letter appearing on the screen or being printed on paper.
- So that all computer systems behave in a similar way it is important that there is an agreed set of codes for characters.
- The agreed set of codes to represent the main characters in the English language is known as ASCII (American Standard Code for Information Interchange).

Below, you can see that each character is represented by a number. The binary table contains 5 binary numbers, and by working out the value of each binary number, we can see which letter it represents by looking it up in the ASCII table.

ASCII TABLE

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

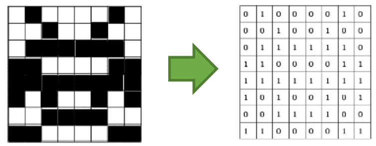
What is my name?

64	32	16	8	4	2	1	
1	0	1	0	0	1	0	82 - **R**
1	0	0	1	0	0	1	73 - **I**
1	0	0	1	1	1	0	78 - **N**
1	0	0	0	1	1	1	71 - **G**
1	0	0	1	1	1	1	79 - **O**

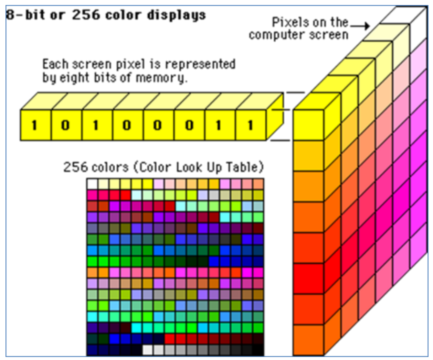
Representing Images

Bitmaps

Bitmap images are made up of rows of "dots" called "pixels" (picture elements). Each pixel is represented by a binary number. Behind the scenes, this 1-bit image (with each shade represented by a bit) is in fact a series of numbers:



In a coloured bitmap, longer binary numbers represent a different colour:



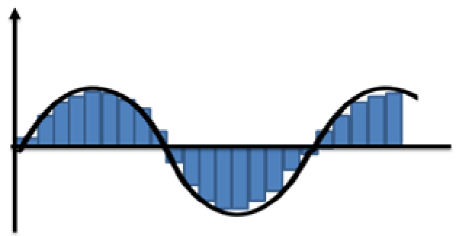
As images get more colourful, longer binary numbers are needed so that a bigger combination of colours can be shown.

Representing Sound

Analogue sounds (sound waves that continuously vary) are pure and of perfect quality.
However, computer recorded sound is not pure, not real and not of perfect quality and this is because sound has been digitised – it has been sampled at set intervals.

Sampling

Sampling is the process by which computers digitise sound.
They measure the height of sound waves at regular intervals and record the measurement as a binary number.
So, whereas analogue sound is continuous over time, digitised sound is made up of lots of 'sound bites' over time.



When computers play sound through a speaker, they process each of the binary measurements and send signals to the speaker making it vibrate in different ways, according to the binary data.