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'.
...the number 33 means 'thirty three' because the digits mean '3 tens and 3 ones'.
...the number 528 means 'five hundred and twenty eight' because the digits mean '5 hundreds, 2 tens and 8 ones'.

10's	1's
1	0

10's	1's
3	3

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

2's	1's
1	0

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:

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.

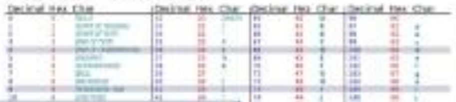
This process then continues, moving through the columns to the left, until we have added the binary numbers.

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

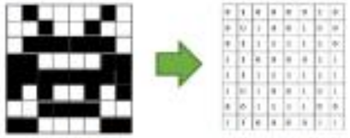


What is my name?

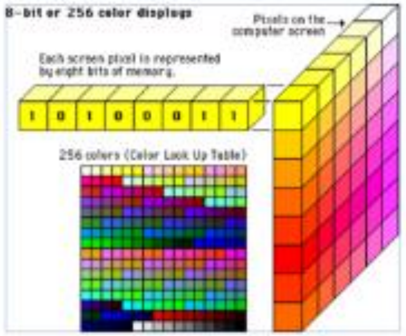
64	32	16	8	4	2	1	
1	0	1	0	0	1	0	82 - **R**
1	0	0	1	0	0	1	73 - **I**
1	0	0	1	1	1	0	78 - **N**
1	0	0	0	1	1	1	71 - **G**
1	0	0	1	1	1	1	79 - **O**

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:

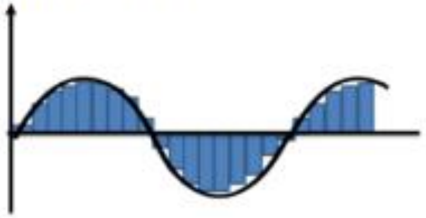


As images get more colourful, longer binary numbers are needed so that a bigger combination of colours can be shown.

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.

Knowledge Organiser: Data Representation

Summary—Data Representation

Computers use electrical signals that are on or off, so they have to see everything as a series of **binary** numbers. This data is represented as a sequence of 1s and 0s (on and off). All data that we want a computer to process needs to be converted into this binary format.

Computers use binary to process data. There are simple techniques to convert between binary and denary and to add two binary numbers together.

How do Computers see the world

Computers use electrical signals that are on or off, so they have to see everything as a series of **binary** numbers. This data is represented as a sequence of 1s and 0s (on and off). **All data that we want a computer to process needs to be converted into this binary format**.

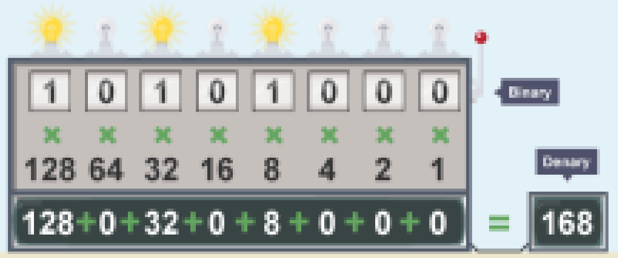
What is Binary

Binary is a number system that only uses two digits: 1 and 0. All information that is processed by a computer is in the form of a sequence of 1s and 0s. Therefore, all data that we want a computer to process needs to be converted into binary.

The binary system is known as a 'base 2' system. This is because:

- there are only two digits to select from (1 and 0)

when using the binary system, data is converted using the power of two.



Alan Turing

Turing was quite a brilliant mathematician, most famous for his work on breaking the German Enigma codes... Without his outstanding contribution the history of the Second World War could have been very different.

Turing was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the Turing machine, which can be considered a model of a general-purpose computer. Turing is widely considered to be the father of theoretical computer science and artificial intelligence.

What is Denary

Thousands	Hundreds	Tens	Units
(1000s)	(100s)	(10s)	(1s)

People use the **denary** (or decimal) number system in their day-to-day lives. This system has 10 digits that we can use: 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9.

Binary to Denary Conversion

To convert a **binary** number to denary, start by writing out the binary place values.

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

Example the binary value of **10101000 = 168 in Denary**

1 x 128
0 x 64
1 x 32
0 x 16
1 x 8

Binary to Denary Conversion

Example 199 **Denary**

199 – 128 = 71. Because 128 could be taken off, put a 1 in the '128' place value column:

128	64	32	16	8	4	2	1
1							

Now repeat for 64: **71 – 64 = 7**

128	64	32	16	8	4	2	1
1	1						

Again for 32: **7 – 32** won't work, so put a 0 in that place value column.

128	64	32	16	8	4	2	1
1	1	0					

Try again for 16: **7 – 16** won't work, so add a 0 to that place value column

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

Next is 8: **7 – 8** won't work. Add a 0 to the '8' place value column

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

Try again for 4: **7 – 4 = 3**, so add a 1 to the '4' place value column.

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

Next try 2: **3 – 2 = 1**, so add a 1 to the '2' place value column.

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

And finally, 1: **1 – 1 = 0** – add a 1 to the '1' place value column.

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

This means that 199 as a binary number is 1100 0111.

Knowledge Organiser: Hexadecimal

Summary—Hexadecimal

Hex is easier to understand than binary. Programmers often use hex to represent binary values as they are simpler to write and check than when using binary.

Hexadecimal to denary

Whereas denary place values are powers of 10, and binary place values are powers of 2, hex place values are powers of 16.

In Computer science, different number bases are used:

- **Denary** is base 10, which has ten units (0-9).
- **Binary** is base 2 , which has two units (0-1)
- **Hexadecimal**, also known as hex, is the third commonly used number system. It has 16 units (0-9) and the letters A, B, C, D, E and F.

Hexadecimal to Denary

Whereas denary place values are powers of 10, and binary place values are powers of 2, hex place values are powers of 16.

Each place value can be represented by the units 0 through to F.

65,536	4,096	256	16	1

To convert hex to denary, simply take each place value that has a unit in it, and add them together.

65,536	4,096	256	16	1
			7	C

Denary to hexadecimal

If the denary number is bigger than 16, divide it by 16. Take the hex equivalent of this result - this represents the first digit. Take the hex equivalent of the remainder - this represents the second digit. If the denary number is smaller than 16, take the hex equivalent of the denary number.

Example - convert 138 to hex
138 ÷ 16 = 8 remainder 10
8 = hex 8
10 = hex A
Result: 8A

Converting between Binary and Hexadecimal

1. Start at the rightmost digit and break the binary number up into groups of four digits. These are known as nibbles. If there are less than four digits, use just that number of digits for that group.
2. Next, convert each group of four digits into denary.
3. Convert each denary value into its hex equivalent.
4. Put the hex digits together.

Example - 11000011 to hex
Break into groups of four - 1100 0011
1100 = denary 12 0011 = denary 3
12 = hex C 3 = hex 3
Result: C3

Converting between Hexadecimal to Binary

1. Split the hex number into individual values.
2. Convert each hex value into its denary equivalent.
3. Next, convert each denary digit into binary, making sure you write four digits for each value.
4. Combine all four digits to make one binary number.

Example - hex FC to binary
F = denary 15 C = denary 12
15 = binary 1111 12 = binary 1100
Result: 11111100

Binary and Hexadecimal Number Systems

8 bits	1 byte (B)
1,000 bytes (1,000 B)	1 kilobyte (KB)
1,000 kilobytes (1,000 KB)	1 megabyte (MB)
1,000 megabytes (1,000 MB)	1 gigabyte (GB)
1,000 gigabytes (1,000 GB)	1 terabyte (TB)
1,000 terabytes (1,000 TB)	1 petabyte (PB)

The binary unit system

Denary	Binary	Hexadecimal
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

The Hexadecimal Number system