Data Representation - AQA Computer Science Cheat Sheet

by [deleted] via cheatography.com/56036/cs/15729/

Number Bases

Denary or **Decimal**	Base 10.
Binary	Base 2. Used by computers to represent all data and instructions. Uses 1s and 0s to powers of 2 to represent whole numbers.
Hexadecimal	Base 16. Used in computing because more values can be represented by fewer characters. This makes it easier for humans to read and understand.

Converting Between Number Bases

Denary to **Binary**	Divide by 2, then read the remainders backwards.
Binary to **Denary**	Multiply the binary numbers (i.e. every single digit) by the relevant place value, then add all of these together.
Denary to **Hexadecimal**	Divide by 16, then read the divisors and remainder backwards. Then convert digits to hex digits.
Hexadecimal to **Denary**	1. Separate the hex digits 2. Convert each digit to binary 3. Concatenate, then convert to denary
Binary to **Hexadecimal**	Convert to denary, then hex.
Hexadecimal to **Binary**	Convert to denary, then binary.

Binary Addition

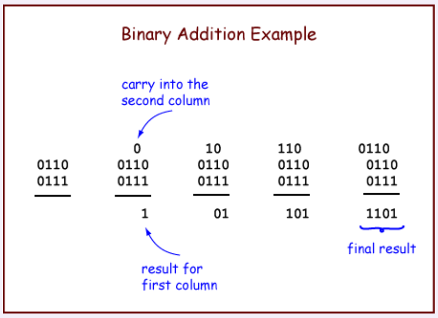
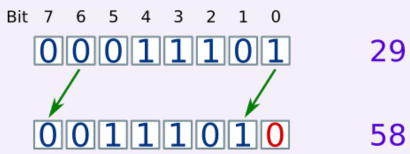


Image: http://chortle.ccsu.edu/assemblytutorial/Chapter-08/ass08-_3.html

Binary Shifts



Binary shifts can be used for multiplication and division by powers of two.

Image: http://wiki.schoolcoders.com/gcse/data-representation/numbers/binary-shift/

Units of Information

Bit	b	A single binary digit.
Byte	B	8 bits.
Kilobyte	kB	1,000 bytes
Megabyte	MB	1,000 kilobytes.
Gigabyte	GB	1,000 Megabytes.
Terabyte	TB	1,000 Gigabytes.

ASCII vs Unicode

What is **ASCII**?

> A character set that uses 7 bits, so can represent up to 128 characters; this means that only Latin letters can be used (i.e. characters/letters from other languages can't be represented). However, it takes up less space that Unicode.

What is **Unicode**?

> Unicode is also a character set, but it can represent many, many values, including non-Latin-based languages.

Representing Images

Pixel	A single point in a graphical issues. Short for 'picture element'.
Bitmap	A grid of pixels, with each pixel represented by a binary number.
Colour depth	Number of colours that can (not necessarily are) be represented in an image, and the corresponding number of bits needed to represent each pixel (e.g. 2 bits for 4 colours). The greater the colour depth, the bigger the file size.
Resolution depth	How much detail there is in an image. The more pixels per inch, the higher the resolution. The higher the resolution, the bigger the file size.
Metadata	Gives the software the information needed to display the image properly (size, resolution depth, colour depth).

Bitmap file size = width x height x colour depth (in bits)

Data Compression

Data compression is used to reduce file size, which means that they take up less storage space. **Lossy** compression is where some data is removed - this means that an image would lose some detail. **Lossless** compression preserves all of the information.

Run length encoding (RLE) uses data frequency pairs to reduce the amount of data stored. It does so by stating the character and then the length of the run. Example:
1001 1111 0101 can be shown as 1 1 2 0 5 1 1 0 1 1 1 0 1 1

Huffman coding is more efficient than RLE. It is also lossless. It finds the frequency of each data item to create a Huffman tree, which assigns the most frequent items the shortest code. When you move down a branch to the left, a 0 is assigned. When you move to the right, a 1 is assigned.
total bits needed = number of bits needed per character x number of characters

Representing Sound

Sample	A measure of amplitude at a given point. Used to convert an analogue wave into a digital format.
Sampling rate	The number of samples taken in a second. Measured in Hertz.
Sampling resolution	The number of bits per sample.
Bit rate	The number of bits used per second of the audio. Usually measured in kilobits per second (kbps).

File size (bits) = sampling rate x resolution x lenght of recording (seconds)