# Knowledge Organiser 1.2 : Data Storage

## 1. Data units

| | |
|---|---|
| Bit (b) | The smallest unit of data. 0 or 1 |
| Nibble (N) | 4 bits |
| Byte (B) | 8 bits (note the difference between b and B) |
| Kilobyte (KB) | 1000 bytes. Note KB is different from Kb |
| Megabyte (MB) | 1000 KB |
| Gigabyte (GB) | 1000 MB |
| Terabyte (TB) | 1000 GB |
| Petabyte (PB) | 1000 TB |

## 2. Conversions

| |
|---|
| Binary to Denary |
| Denary to Binary |
| Hexadecimal to Denary |
| Denary to Hexadecimal |
| Binary to Hexadecimal |
| Hexadecimal to Binary |
| Left Binary Shift |
| Right Binary Shift |

## 3. Operations

| | |
|---|---|
| Binary addition | You should arrange the two binary numbers above each other so that the columns line up. Start on the rightmost digit and add them. If there are any carries, write them down next to the next left column. |
| Overflow | If the answer to the left column results in a carry, this is known as an overflow and it causes an overflow error. This can cause problems if a computer program hasn't been written to handle overflows. |
| Left Binary Shift | Make the number longer, and therefore bigger. Each place it shifts will double the value. A binary left shift of one place (<<1) will double the value, a binary left shift of two places (<<2) with quadruple. |
| Right Binary Shift | Make the number shorter, and smaller. The right most digit is "lost", so we forget about it. A binary right shift of one place (written as >>1) halves the number, and a binary right shift of two places (>>2) will quarter it. |

## 4. Characters

| | |
|---|---|
| Individual Characters | Each character is assigned an individual binary code to represent it. The number of bits depends on the 'encoding' used |
| Character Set | The name given to a collection of characters matching to binary codes. There are many examples. |

## 5. Examples of Character Sets

| | |
|---|---|
| ASCII | 7-bits to represent characters allowing 127 characters to be represented |
| Unicode | 16 / 24 / 32 bits. Covers many modern and historic languages, as well as lots of |

## 6. Images

| | |
|---|---|
| Pixel | The smallest element of a bitmap image. Pixels desk |
| Vector vs Bitmap | A vector image describes the lines and shapes. A bitmap image consists of rows of coloured dots. |
| Colour Depth | The number of bits used to represent each pixel in a bitmap image. An 8 bit image can show $2^8$ or 256 colours. |
| Resolution | In a bitmap image resolution is measured in DPI (dots per inch). The higher the resolution the better the picture quality |
| Metadata | Data that is saved before and after the image to tell the computer how to decode the image. It includes the size in pixels (width x height), the colour depth, the resolution |

## 7. Sound

| | |
|---|---|
| Analogue / Digital | Analogue sound waves must be converted into digital sound waves by taking a sample of the sound at set intervals. This is because computers can only work with digital 'numbers', and not analogue 'sound' |
| Sample rate | Number of times analogue signal is sampled per second. Measured in Hertz |
| Bit depth | Number of bits used per sample. Sometimes known as sample resolution |
| File size | Sample rate x sample resolution x seconds |

## 8. Compression

| | |
|---|---|
| Compression | Compression is when a file is encoded so it uses fewer bits than the original file format |
| Lossless compression | Gets rid of unnecessary data to re-present data without losing any information. This process is reversible |
| Lossy compression | Gets rid of the least essential data. This is an irreversible process: once data is lost it can't be recovered |