# Number System - Gist

## Binary Numbers

Decimal number 13 in binary is 00001101

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

8+4+1 = 13

| Base 10 (Decimal) | Base 2 (Binary) | Base 16 (Hex) |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 1 | 1 |
| 2 | 10 | 2 |
| 3 | 11 | 3 |
| 4 | 100 | 4 |
| 5 | 101 | 5 |
| 6 | 110 | 6 |
| 7 | 111 | 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 |

### Adding Binary numbers (start adding from the right)

1. Keep your numbers in the correct columns
2. 1 + 0 = 1
3. 1+1 = 10 *in binary*
4. 1+1+1 = 11 *in binary*

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| | + | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| Carry on | | 1 | 1 | 1 | | | 1 | | |
| Answer | | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |

### Binary Shift

**Left shift – Multiplication**

11011 after left shift becomes 110110

| 32 | 16 | 8 | 4 | 2 | 1 | |
|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 1 | =27 |
| 1 | 1 | 0 | 1 | 1 | 0 | =54 |

**Right shift – Division**

11011 after right shift becomes 1101

## ASCII - American Standard Code for Information Interchange

**Character set** - Every symbol printed (on screen or paper) is a character and has a unique computer code. A complete collection of characters is a character set.

| Standard code | Bits used | Maximum characters in the character set |
|---|---|---|
| ASCII | 7 | 128 |
| Extended ASCII | 8 | 256 |
| Unicode | 16 or more | Over 1.1 million |

ASCII could not include characters from European languages, therefore Extended ASCII was developed. Extended ASCII could not include characters from all world languages therefore Unicode was introduced.

**Binary to Hex**

**Divide the binary number into nibbles. Convert each nibble to Hex**

**11010101 → 1101 0101 →**

| 8 | 4 | 2 | 1 |
|---|---|---|---|
| 1 | 1 | 0 | 1 |
| 13 → D | | | |

| 8 | 4 | 2 | 1 |
|---|---|---|---|
| 0 | 1 | 0 | 1 |
| 6 → 6 | | | |

**Answer D6**

# Hexadecimal numbers

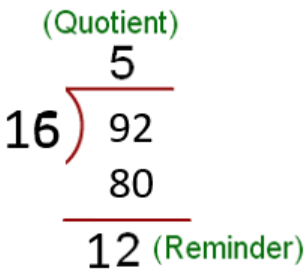
**Base: 16**

| Denary | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Hexadecimal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |

**Decimal to Hexadecimal numbers**

1. Divide Decimal number by 16
2. We get quotient and remainder
3. Convert quotient and remainder to Hex
4. Join quotient and remainder to form the Hex number

Example: 92 in Hex is 5C




**Hexadecimal to Decimal numbers**

Convert 3B to decimal

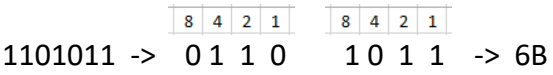
3*16= 48

B= 11

48+11=59

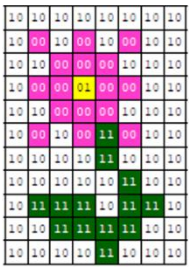
**Binary to Hex**

1. Start from the right.
2. Break the Binary into nibbles.
3. Convert each nibble into Hex.




# Images- Bitmap Graphics


A bit-map graphic is made up of rows and columns of pixels.
Each pixel stores a colour code in the form of a binary number.
The binary number corresponds to a specific colour.
**Colour depth** is the number of bits used to store a colour.
The colour depth determines the maximum number of possible colours an image can have.



A colour depth of 24 bits is often called "**true colour**"
A colour depth of 1 bit is used for black and white image.

- a jpg graphic is always 24 bit colour.
- a gif file is always stored using 8 bit colour.
- a png file can vary its colour depth from 1 bit to 32 bit colour.

***Storage Requirements = Resolution (Length*Width) x Colour Depth***

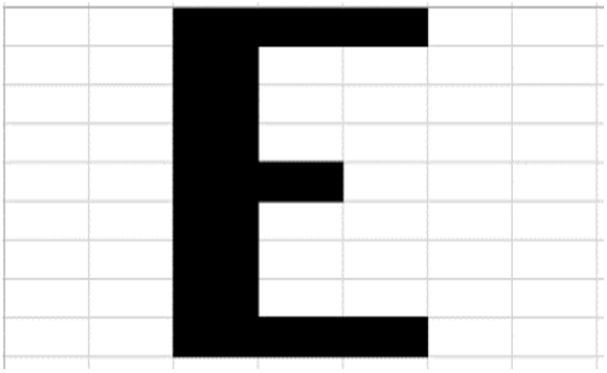
**Meta data:** Data about data

Examples of meta data= image width, depth, time and date of image when taken, location, camera model

# Images- Vector Graphics

A vector graphic is produced using shapes. Instead of storing every single dot in the picture as a bitmap, vector images store a set of instructions of HOW to draw the shape.
Every time you open the file, the instructions are executed and the image is drawn.

## File Encoding **Run length Encoding**



| | |
|---|---|
| Wwbbbww | 2w3b2w |
| Wwbwwww | 2wb4w |
| Wwbwwww | 2wb4w |
| Wwbwwww | 2wb4w |
| Wwbbwww | 2w2b3w |
| Wwbwwww | 2wb4w |
| Wwbwwww | 2wb4w |
| Wwbwwww | 2wb4w |
| wwbbbww | 2w3b2w |

Size before RLE = 63 characters

Size after RLE = 48 characters

## Sound

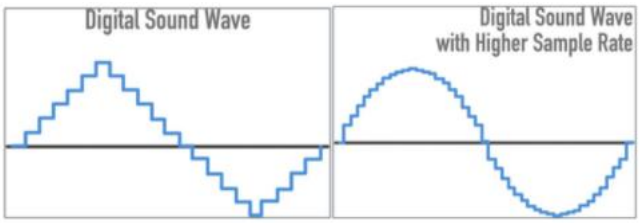
Sound exists as waves – however as computers only understand binary values sound needs to be converted into binary.
Thousands of samples per second are taken of sound and the amplitude of each sample is then stored as binary data.
The quality of digital sound depends on different factors:

- sample frequency/rate - The number of audio samples captured every second, measured in Hertz
- sample size/ bit depth - Number of bits available to store each sample e.g. 16-bit

**Sampling** = Recording analogue sound, at regular intervals and converting the amplitude of each snippet of sound to a binary value.

**Sample rate** = Number of samples taken per second (Hertz) also called frequency

**Bit depth** is the number of bits used to store each sample.
**Bit rate =** Number of bits of data processed per second. (kilobits per second kbps).
**Bit rate =** sample rate × bit depth × channels




Sample rate for music CDs is 44,100 samples per second = 44,100 hertz or 44.1 kHz
Telephone networks and VOIP services use a sample rate as low as 8 kHz

Sound file size = Sample rate * Bit depth * Number of channels (mono or stereo) * Length of the recording (in seconds)

# Compression

Compression means reducing the size of the file so that it takes up less storage space or bandwidth when it is transmitted.

**Benefits:**

- It uses less bandwidth when they are downloaded
- The transfer speed is quicker
- It takes up less storage space on the servers of the storage providers
- The smaller files reduce congestion on the internet, which is good for everyone
- It makes audio and video files suitable for streaming

**Usage:**
Files, especially audio and graphic files can become very large. Millions of images, audio and video files are uploaded everyday on social media. Users need the uploading and downloading time to be short and the social media sites want to store them in as small space as possible. That is why compression is important.

## Compression Methods

**Lossless compression**:

If the compressed file can be decompressed without any loss of data, it is called lossless compression.
Lossless compression uses algorithm and Index file to compress the file. The algorithm checks for redundancy: repetition of items.

- The repeated items along with their locations are recorded in an index.
- The repeated words are removed and the file is saved along with its index.
- When the file is opened, the index is referred and the words replaced in the listed locations. Hence presenting the original file.

Used for Text files

**Lossy compression**.

If the file is compressed by removing some of the data, the original cannot be recovered and it is called lossy compression.

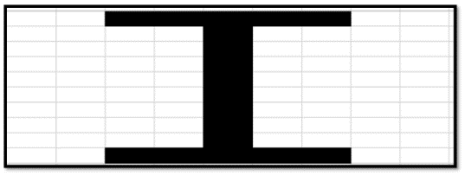
Used for image, sound and video files

**Run- Length encoding (RLE)**

RLE works by reducing the physical size of a repeating string of characters. This repeating string, called a **run**, is encoded in two bytes. The first byte represents the number of characters in the run and the second gives the character.

For example the 18 byte string aaabbbbbbbccccccccc, after run-length encoding 3a6b9c is reduced to 6 bytes.

Find the RLE algorithm for this image. The first row has been done for you.



Wwbbbbbww 2w5b2w