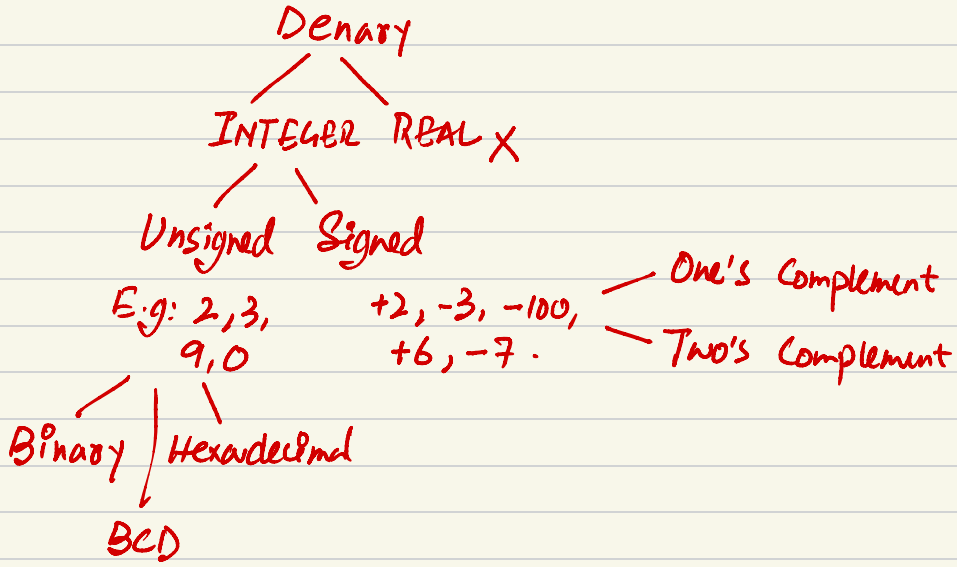
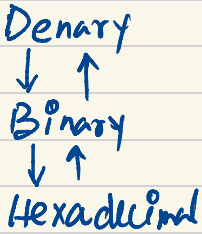
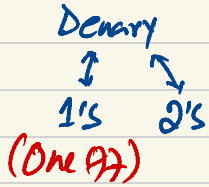
Number System:







Denary (Base 10):

0 1 2 3 4 5 6 7 8 9

Total Digits in Denary = 10 ← Base

5693

5	6	9	3	→ Position (right to left)
10^3	10^2	10^1	10^0	→ Base
1000	100	10	1	→ Worth, weightages
5×1000	6×100	9×10	3×1	

5000 + 600 + 90 + 3 = 5693

Binary (Base 2):

0 1 → Binary Digits → Bit

Total bits = 2 → Base 2.

Constraint: We can only write Binary numbers in the combinations of 8 bits. That is **8 or 16** or 24 etc.

Syllabus requirement is to have up to 16 bits solutions.

Working:

$(01100100)_2 = (100)_{10}$

0	1	1	0	0	1	0	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

0 + 64 + 32 + 0 + 0 + 4 + 0 + 0 = 100

Hexadecimal (Base 16):

0 — 9 0 1 2 3 4 5 6 7 8 9

A — F A B C D E F ✓

10 + 6 Base 16

7	0 1 1 1	
10	1 0 1 0	

8 4 2 1

0 0 0 0 Smallest

1 1 1 1 Largest

Reason for use:

1. Compact representation.
2. Memory addressing.
3. Color Representations.
4. Debugging and low-level programming.

Single Byte Conversions

(msb) Most sig bit — Binary — Least significant bit (lsb)

A B C D E F
10 11 12 13 14 15

0 Even
1 Odd.

Denary	128	64	32	16	8	4	2	1	Hex
	8	4	2	1	8	4	2	1	
100	0	1	1	0	0	1	0	0	64
206	1	1	0	0	1	1	1	0	CE
7	0	0	0	0	0	1	1	1	07
127	0	1	1	1	1	1	1	1	7F
150	1	0	0	1	0	1	1	0	96
200	1	1	0	0	1	0	0	0	C8
163	1	0	1	0	0	0	1	1	A3

Range: How many numbers? Largest & Smallest numbers? } For the given number of bits.

$2^4 = 16$ → 0 … 15

$2^{16} = 65536$ → 0 … 65535

$2^8 = 256$ → 0 … 255

Two Bytes Conversions

$(532)_{10} = (?)_{2} = (?)_{16}$

$(214)_{16} = (532)_{10} = (0000001000010100)_{2}$

	512	255	128	64	32	16	8	4	2	1
532	0	0	1	0	0	0	0	1	0	0
	2			1				4		

BCD (Binary Coded Decimal)

Denary
0 → 9

4-bits BCD representation.

$2^4 = 16$ — 0–9

~~15~~

0 1 2 3 4 5 6 7 8 9

~~10 11 12 13 14 15~~

Format of BCD:

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

$(29)_{10} = (00101001)_{BCD}$

$(128)_{10} = (0000000100101000)_{BCD}$

1000 0011 0111 0001 ← BCD

1 3 7 1 ← Denary (Decimal).

Uses:
- Digital Display
- Financial Calculations
- Embedded Systems.
- Legacy Computing Systems

Signed Binary Systems:

One's & two's complement.
Expectations are to convert:
- From unsigned to signed (binary)
- From signed Denary to Bin.
Confined to a byte only.

1's Complement:

+92	01011100
-92	10100011

Invert all

Uses: * Representation of -ve numbers.
* Simple arithmetic operations.

In One's Complement there are two zeros.

+0	+v zero	0000
-0	-v zero	1111

Binary additions and subtractions:

1's Complement additions:

-2 + 7

0 +ve
1 -ve

+2	00000010
-2	11111101
+7	00000111

+7	00000111
-2	+ 11111101
	100000100
	+ 00000001
	00000101

→ end-around carry.

	C	S
0	0	0
1	0	1
2	1	0
3	1	1

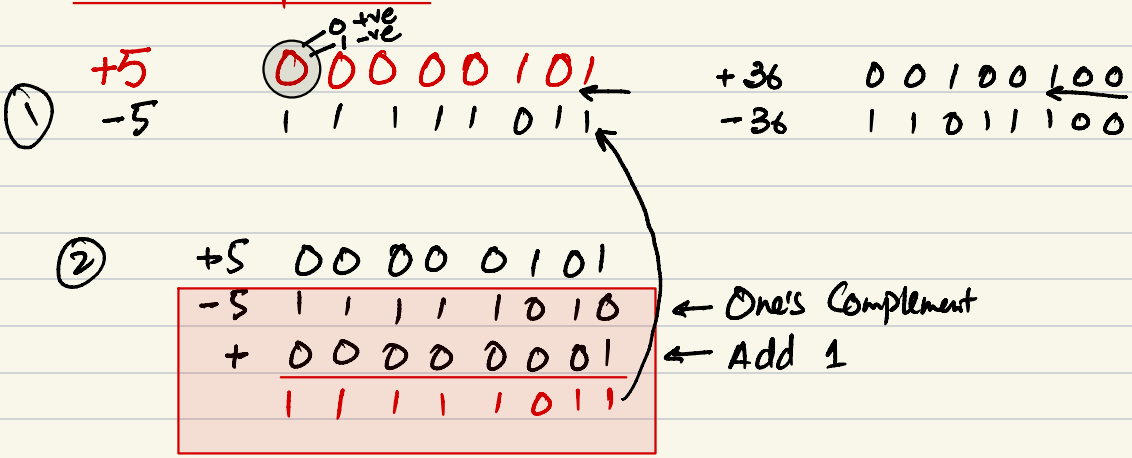
+10	00001010
-10	11110101

-10 + 3

-10	11110101
+3	00000011
-7	11111000

11111000 ✓	-7
00000111	+7

Two's Complement:



Features of 2's Complement:

* There is only one representation for zero (00000000)
* To find the negative of a number, you take its two's.
* Arithmetic ops like addition and subtraction are more straight forward b/c there is no need to handle carry.

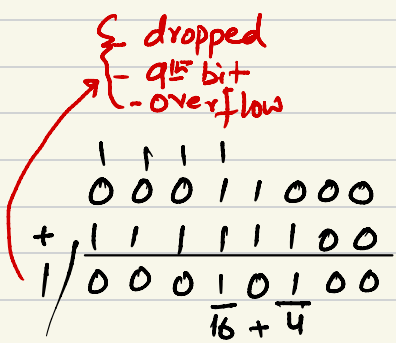
Uses:

* Single zero
* Ease of arithmetic ops.
* Widely supported.

Addition & Subtraction:

a. +24 − 4

+24
− 4
+20



Range 2's Complement:
−128 ⟷ +127

If expected answers are within range (−128 − +127) then a carry drop doesn't matter.

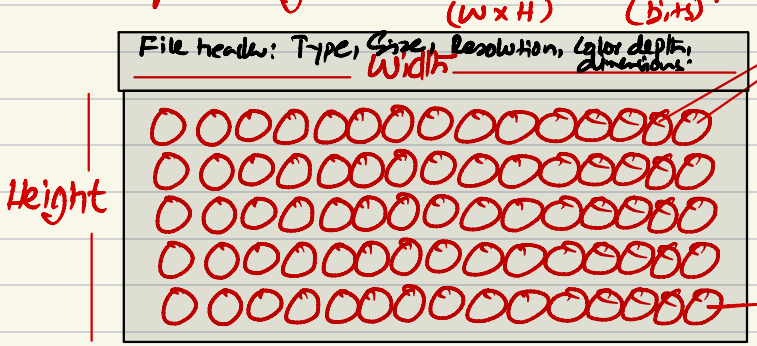
+5 +4 -3

	1	1	1	1	1		1	
+5	0	0	0	0	0	1	0	1
+4	0	0	0	0	0	1	0	0
-3	1	1	1	1	1	1	0	1
+6	1	0	0	0	0	0	1	0

					1			
+4	0	0	0	0	0	1	0	0
+23	0	0	0	1	0	1	1	1
+27	0	0	0	1	1	0	1	1

16 + 8 + 2 + 1 = 27

Image: - Bitmap Image Size = Resolution (W x H) x Color depth (bits)



Pixel:

Image Resolution: Pixels in Height & Width
Width x Height

Screen Resolution:

Color Depth: How many bits are used to represent the color of each pixel.

2^5 = 32 Colors 1-bit 2^1 = 2 Colors 16 bits 2^{16} = 65,536 color

8-bit 2^8 = 256 Colors

Sizing:

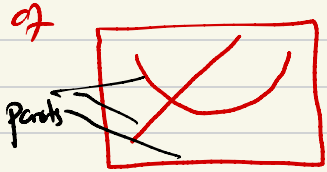
1 Byte = 8 Bits.

2^{10} =	1024 Bytes = 1 Kibibyte	Ki
2^{20} =	1024 KiB = 1 Mebibyte	Mi
2^{30} =	1024 MiB = 1 Gibibyte	Gi
2^{40} =	1024 GiB = 1 Tebibyte.	Ti

Vector Graphics:

Encoding: Geometric description, scaled to any size without distortion.

Drawing Objects: represent parts of the image. line, curve, polygon, or a composite.

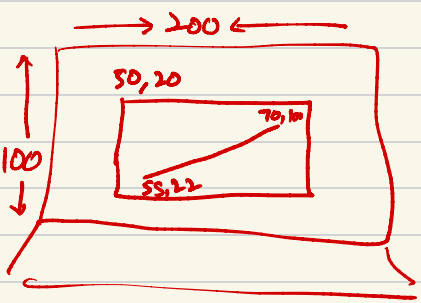


drawing list

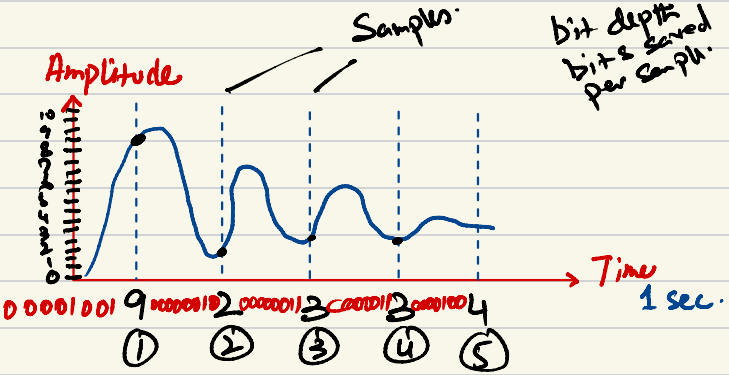
drawing object.

```
< rect  x="50"  y="10"  width="20"
  height="10"  fill="red" >
< line  x1="70"  y1="100"  x2="55"
  y2="22"  stroke="red" >
```

property



Sound:



bit depth = 8 bits per sample.

Sound file size = Sample resolution (bit depth) × Sampling rate Samples/sec × Time (sec.).

$8 \times 5 \times 5$

$= 200$ bits $/ 8$

$= 25$ Bytes.