

PYTHON PROGRAMMING

GATE DA/DSA

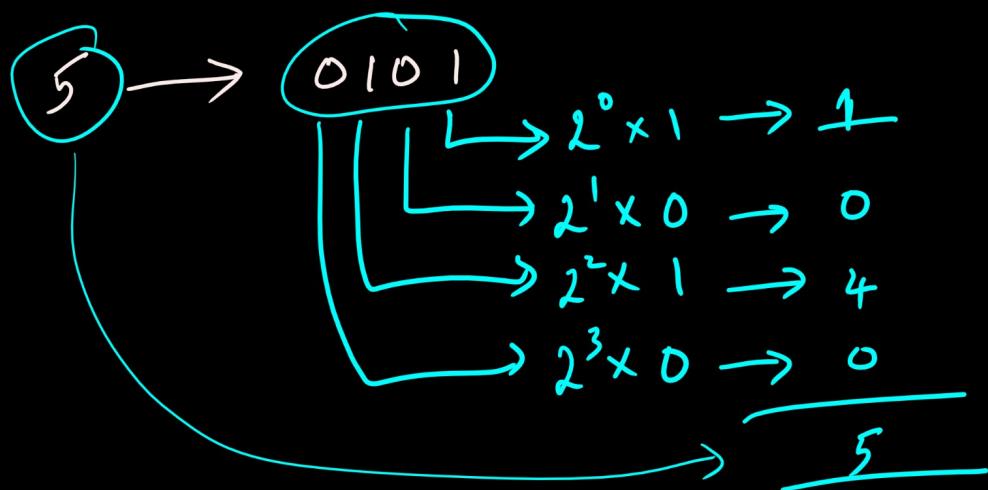
Agenda :

- Operators (Assignment, Bitwise, Membership, Identity)
- Operator Precedence ←
- Unicode Representation ←

Operators
↳ 7 operators

Operators: Assignment ✓
Bitwise

Bitwise Operators :



$\&$ - Bitwise AND

X	Y	Result (4) - Bitwise And.
0	0	$0 \& 0 = 0$
0	1	$0 \& 1 = 0$
1	0	$1 \& 0 = 0$
1	1	$1 \& 1 = 1$

$$\begin{array}{ccc}
 0 & 0 & 0 \\
 \hline
 0 & & \\
 1 & 1 & 1
 \end{array}
 \rightarrow
 \begin{array}{c}
 0 \\
 0 \\
 1
 \end{array}$$

$$\begin{array}{ccccccc}
 x_1 & x_2 & x_3 & \dots & x_n & \rightarrow & R_{(F)} \\
 | & | & | & \dots & | & \rightarrow & 1
 \end{array}$$

$| \rightarrow \text{Bitwise OR}$

x	y	$\text{Result}_{\text{Bitwise OR}}(1)$
0	0	$0 0 = 0$
0	1	$0 1 = 1$
1	0	$1 0 = 1$
1	1	$1 1 = 1$

$$0 \quad 0 \quad 0 \rightarrow 0$$

$\wedge - (\text{Bitwise XOR})$

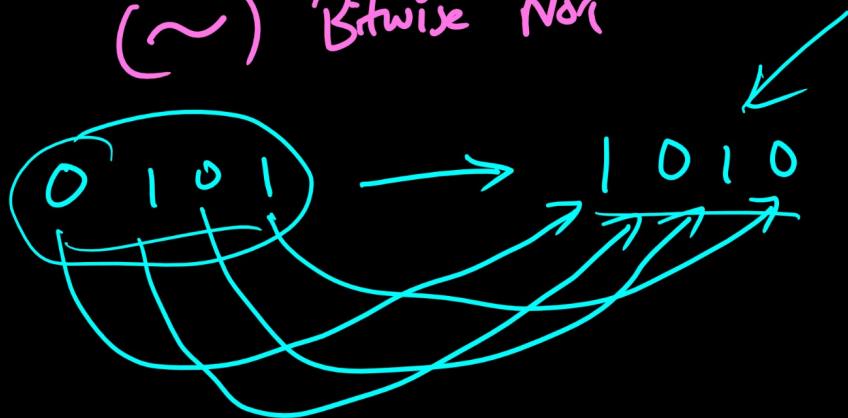
x	y	$x \wedge y \text{ (Result)}_{(^)}$
0	0	0
0	1	0
1	0	0
1	1	1

$$\begin{array}{c} 0 \wedge 0 \wedge 0 \rightarrow \\ 0 \wedge 0 \rightarrow 0 \end{array}$$

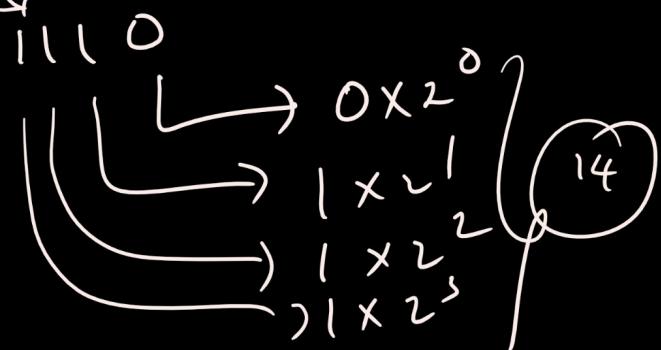
$$\begin{array}{c} 0 \wedge 1 \wedge 0 \\ \perp \wedge 0 \rightarrow 1 \end{array}$$

~ 13

(\sim) Bitwise NOR

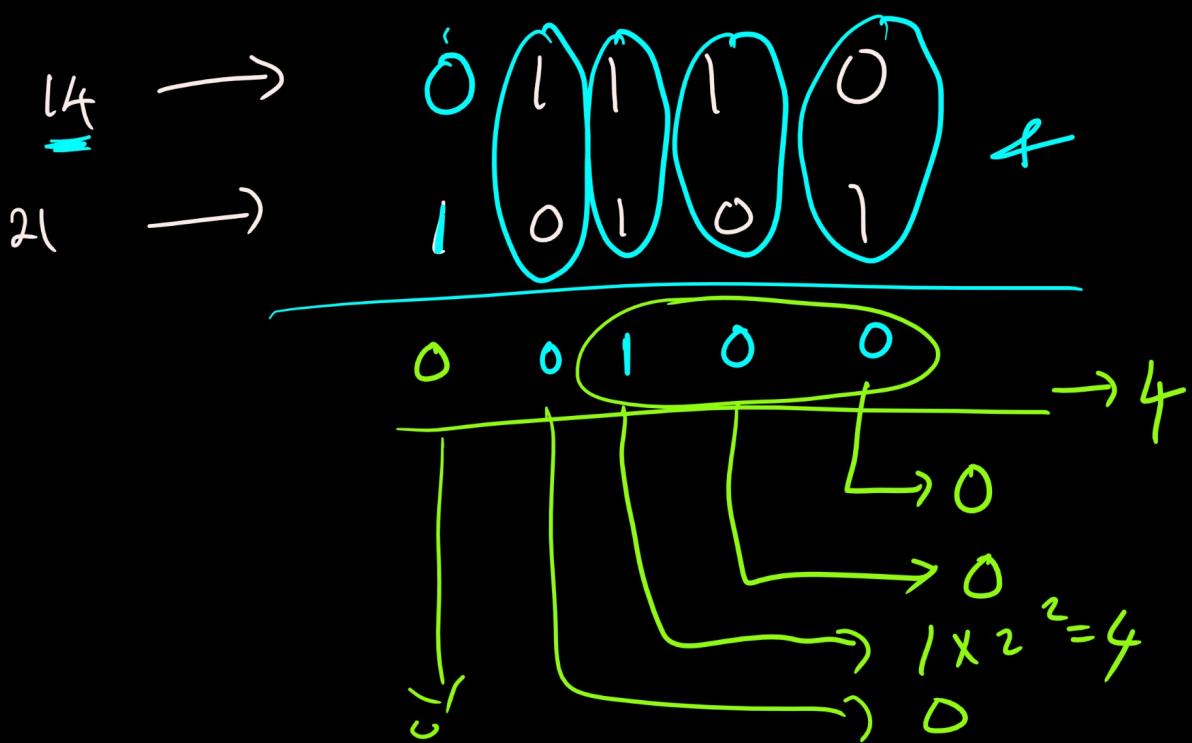


14



$$21 \rightarrow 10101$$

0	1	1	1	0	$\xrightarrow{4 \text{ bit}}$
					$\xrightarrow{14 \rightarrow 1110}$
					$\xrightarrow{21 \rightarrow 10101}$

$$\begin{array}{c|cc} 0000 & 1110 \\ \hline 0001 & 0101 \end{array}$$


0	→	0 0	$\xrightarrow{7 \rightarrow 111}$
1	→	0 1	$\xrightarrow{8 \rightarrow 1000}$
2	→	1 0	
3	→	1 1	
4	→	1 0 0	
5	→	1 0 1	
6	→	1 1 0	

Diagram illustrating the binary tree representation of the number 21. The root node is circled in blue and labeled 21. The tree has three levels of nodes. The path from the root to the leaf node 1 is highlighted in green. The values of the nodes are as follows:

- Root: 21
- Level 1: 2, 23
- Level 2: 2⁰, 2², 2¹, 2⁰
- Leaf node: 1

The path sum for the green-highlighted path is calculated as $16 + 4 + 1$.

$$1 + 2 + 4 + 16 + 32 +$$

$$\begin{array}{r} 14 \rightarrow (0)(1) \\ 21 \rightarrow (1)(0) \\ \hline \end{array}$$

$$\underline{1} \quad \underline{1} \quad \underline{1} \quad \underline{1} \quad \underline{1}$$

$$\begin{aligned} & \left(\begin{array}{l} 1 \\ 2 \\ 4 \\ 8 \\ 16 \end{array} \right) \rightarrow 2^0 \times 1 = 1 \\ & \left(\begin{array}{l} 1 \\ 2 \\ 4 \\ 8 \end{array} \right) \rightarrow 2^1 \times 1 = 2 \} \\ & \left(\begin{array}{l} 1 \\ 4 \\ 8 \end{array} \right) \rightarrow 2^2 \times 1 = 4 \} \\ & \left(\begin{array}{l} 1 \\ 8 \end{array} \right) \rightarrow 2^3 \times 1 = 8 \} \\ & \left(\begin{array}{l} 16 \end{array} \right) \rightarrow 2^4 \times 1 = 16 \} \end{aligned}$$

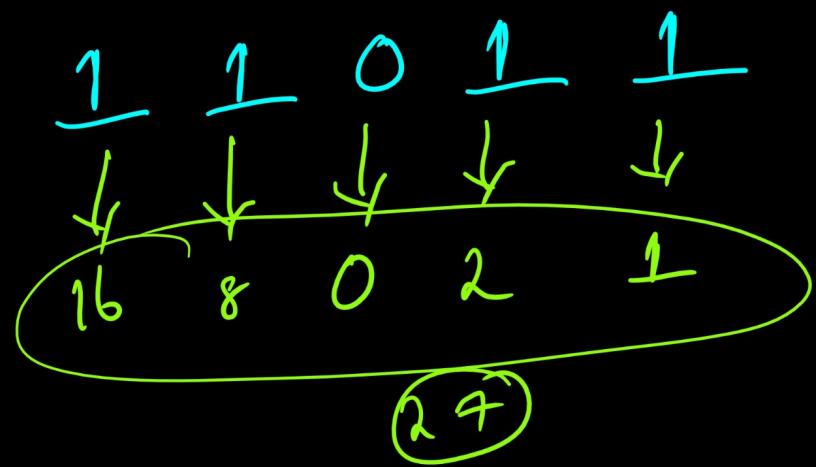
Summing $\rightarrow \frac{11111}{2^5 - 1} = 31$

$\tau \rightarrow 111111$

$$2^\tau - 1 = \underline{127}$$

$14 \rightarrow 01110$ (\wedge)

$21 \rightarrow \underline{101101}$



$\boxed{\tau, 1, \wedge}$

$\sim \circlearrowleft \rightarrow 13$

$13 \rightarrow$

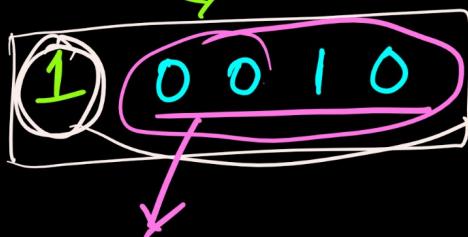
$\sim 13 \rightarrow$

$$13 \rightarrow \begin{array}{r} 1101 \\ \hline 0010 \end{array}$$

← ②

Sign bit → —

$$\begin{array}{r} 13 \\ \textcircled{0} \quad \begin{array}{r} 1101 \\ \hline 0010 \end{array} \\ \text{sign bit} \end{array}$$



if sign bit is 1,
it indicates
it's a
negative
number.

$$14 \rightarrow \begin{array}{r} 01110 \\ \hline \end{array}$$

$$\begin{array}{r} 1 \\ 7 \\ 5 \\ 2 \\ 2 \end{array}$$

$$\textcircled{-14} \rightarrow \begin{array}{r} 1110 \\ \hline \end{array}$$

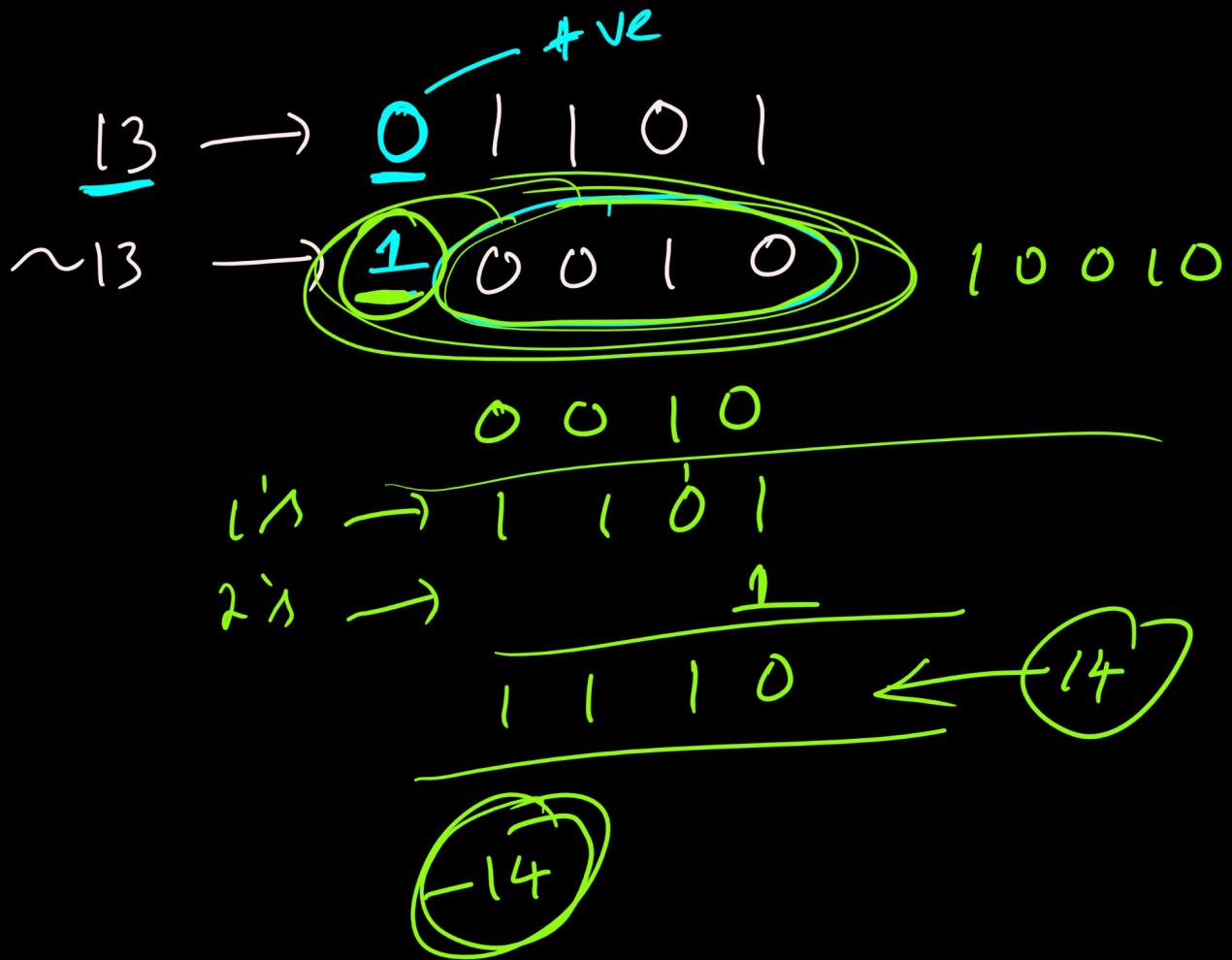
$$\begin{array}{r} 1110 \\ \hline 0001 \\ 1 \end{array}$$

$$\begin{array}{r} 0 \quad 0 \quad 1 \quad 1 \\ \overline{0} \quad \overline{1} \quad \overline{0} \quad \overline{1} \\ \hline 1 \quad 1 \quad 2 \end{array} \xrightarrow{\text{Complement}} \begin{array}{r} 1110 \\ \hline 0010 \end{array}$$

$$\boxed{10010}$$

→ -14
computer

Negative numbers stored in
2's Complement format.



$$\boxed{\sim n \Rightarrow -(n+1)}$$

$$\begin{aligned}\sim 13 &\Rightarrow -(13+1) \\ &= -14\end{aligned}$$

$$\begin{aligned}\sim 105 &\Rightarrow -(105+1) \\ &= -106\end{aligned}$$

$$\begin{aligned}
 \sim -14 &= -(-14+1) \\
 &= -(-13) \\
 &= \underline{\underline{13}}
 \end{aligned}$$

$\sim \textcircled{-14}$ is: $\begin{array}{r} 0 \\ \boxed{1} \\ 000 \\ \hline 110010 \end{array}$
 2's: $\overbrace{\quad\quad\quad\quad\quad\quad}^{\text{negate this}}$

$-14 \rightarrow \begin{array}{r} 110010 \\ \textcircled{0} \textcircled{1101} \\ \hline 110010 \end{array}$
 +ve $\quad \underline{13}$

$\sim -14 \rightarrow \underline{\underline{13}}$

$-14 \rightarrow \begin{array}{r} 1 \\ 000 \\ \hline 10 \end{array}$
 sign bit
 $\downarrow \quad -16 + 1 = \underline{\underline{-14}}$

$$\boxed{\sim \textcircled{n} \rightarrow -(n+1)}$$

+ve
 -ve

$$\boxed{101 \rightarrow -(101+1) = -102}$$

$$\boxed{-101 \rightarrow -(-101+1) = 100}$$

$$14 \rightarrow \begin{array}{r} \underline{0} \\ \text{Signbit} \end{array} \begin{array}{r} \underline{1110} \\ \swarrow \\ 0001 \end{array}$$

$$-14 \quad \begin{array}{r} \underline{1} \\ \text{Signbit} \end{array} \begin{array}{r} \underline{0010} \\ \overline{0010} \end{array}$$

$$\begin{array}{r} 10 \quad \underline{0} \quad 1010 \\ \hline -10 \quad \underline{1} \quad 0110 \end{array} \quad \begin{array}{r} \cdot \\ \hline 0101 \end{array} \quad \begin{array}{r} 1 \\ \hline 0110 \end{array}$$

(& | ^ ~)

Left Shift

4 Right Shift

Left Shift : 10

Right Shift : 10



10

2

shift
2 bits
to left

10 → 00001010
00001010
00101000
320800
 $32 + 8 = 40$

14 → 0000111000
00
00111000

128 64 32 16 8 4 2 1
255 → | | | | | | | | → 255.

0 0 0 0 [1 1 1 1 1 1 1 1]

0 0 1 1 1 1 1 0 0 → 1020
512 256 128 64 32 16 8 4 2 1

$n \gg 1$

↪ $n \times 2$

$n \gg 2$

↪ $n \times 4$

n << K

$n * 2^k$

Left Shift
(\ll)

Right Shift
($>>$)

$n \ll k$

Result: $n * 2^k$

$n >> k$

$n // 2^k$

$n =$ 20

16 8 4 2 1

20 → 00 0 1 0 1 0 0

0 0 0 1 0 1 0 1 0 0

Left Shift by 1: 0 0 1 0 1 0 0 → 40

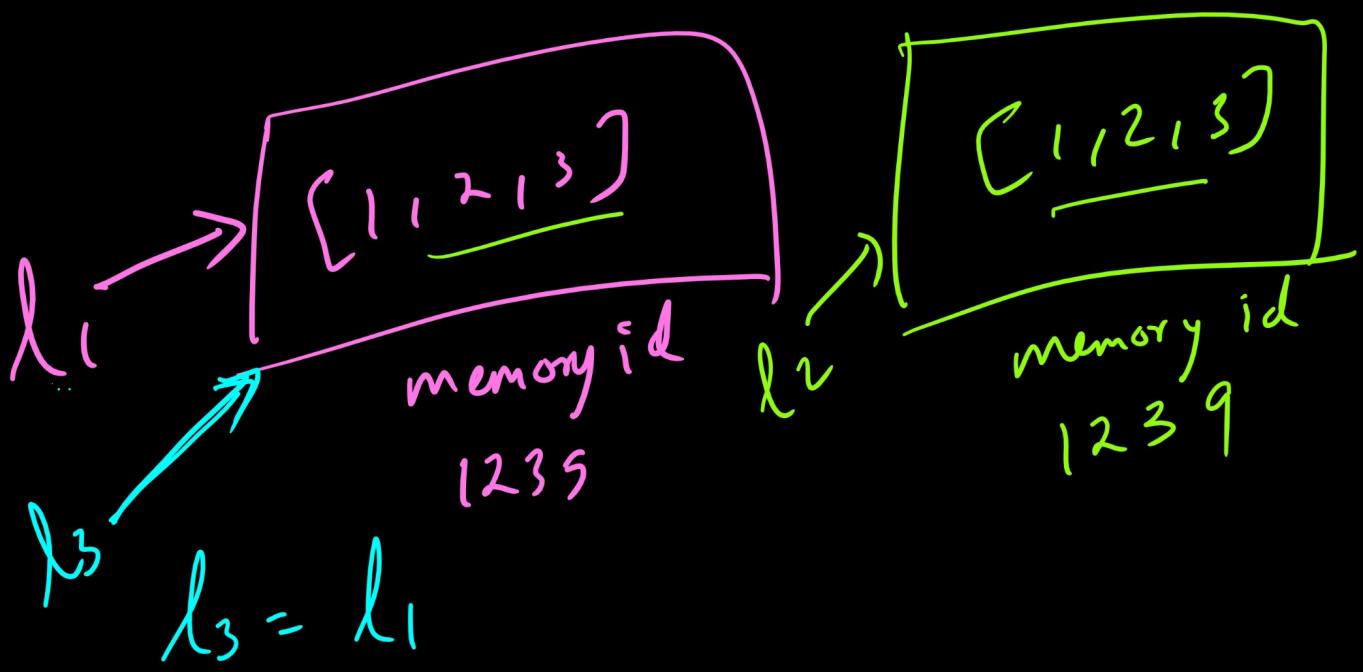
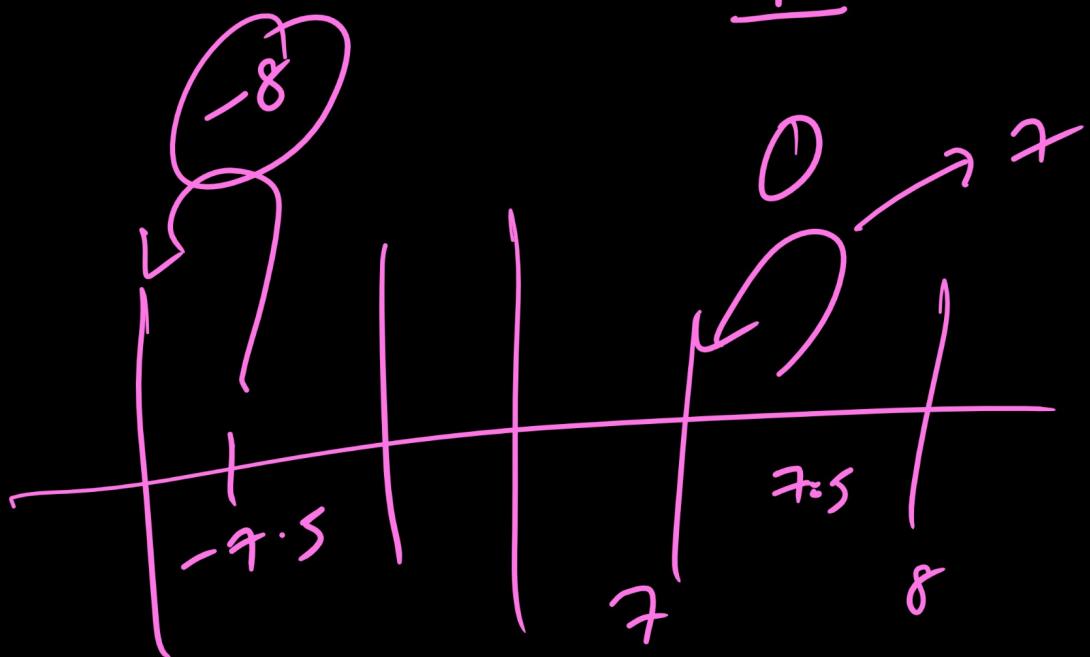
Left Shift by 2: 0 1 0 1 0 0 0 → 80

160

20 → 0 0 0 1 0 1 0 0 20
↓

Right Shift by 1: 0 0 0 1 0 1 0 → 10

Right Shift by 2: 0 0 0 1 0 1 → 5



Operator Precedence :

2. Arithmetic Operators

$$\overbrace{5 + 12 * 7}^{\substack{\rightarrow \\ 17 * 7 \\ 11 * 7}} \overbrace{(2 - 10)}^{\leftarrow} \overbrace{58 \cdot 5 - 10}^{\substack{\rightarrow \\ 48 \cdot 5}} \rightarrow \underline{48 \cdot 5}$$

$$x / -8$$

1. Parenthesis / Brackets → (), [], { }

2. Exponents Order → ~~*~~ * Right to Left

3. Division
Multiplication } → * . / , //, -/.
Left to Right

4. Addition
Subtraction $\rightarrow (+, -)$
Left to Right

$$5 + 9 = 2$$



$$5 + \textcircled{12 * 7 / 2} - 10$$

1. *, /

$$\text{i)} 84$$

$$\text{ii)} 84 / 2 \rightarrow 42.0$$



$$5 + 42.0 = 10$$

$$\text{iii)} 47.0 - 10 = 37.0$$

$$8^{\frac{2}{3}} = 64^{\frac{1}{2}}$$

$2^{**3} * 2^2$

2^{**1}

Right to Left

$3^2 = 9$ $2^9 = 512$

Operator

Precedence

Table :-

Precedence	Operator	Associativity
1	Parenthesis	Left to Right
2	Exponentiation $(**)$	Right to Left
3	+x, -x, $(\sim x)$ → bitwise not	Right to Left
4	*	Left to Right
5	/, //, %	Left to Right
6	<<, >>	Left to Right

7.

AND

\wedge
bitwise AND

Left to
Right

8

XOR

\wedge
bitwise XOR

Left to
Right

9

OR

\vee
bitwise OR

Left to
Right

10

in, not in,
is, is not,
<, <=, >=,
!=, ==

Left to
Right

11

not X
Boolean not

Right to
Left

12

and

Boolean AND

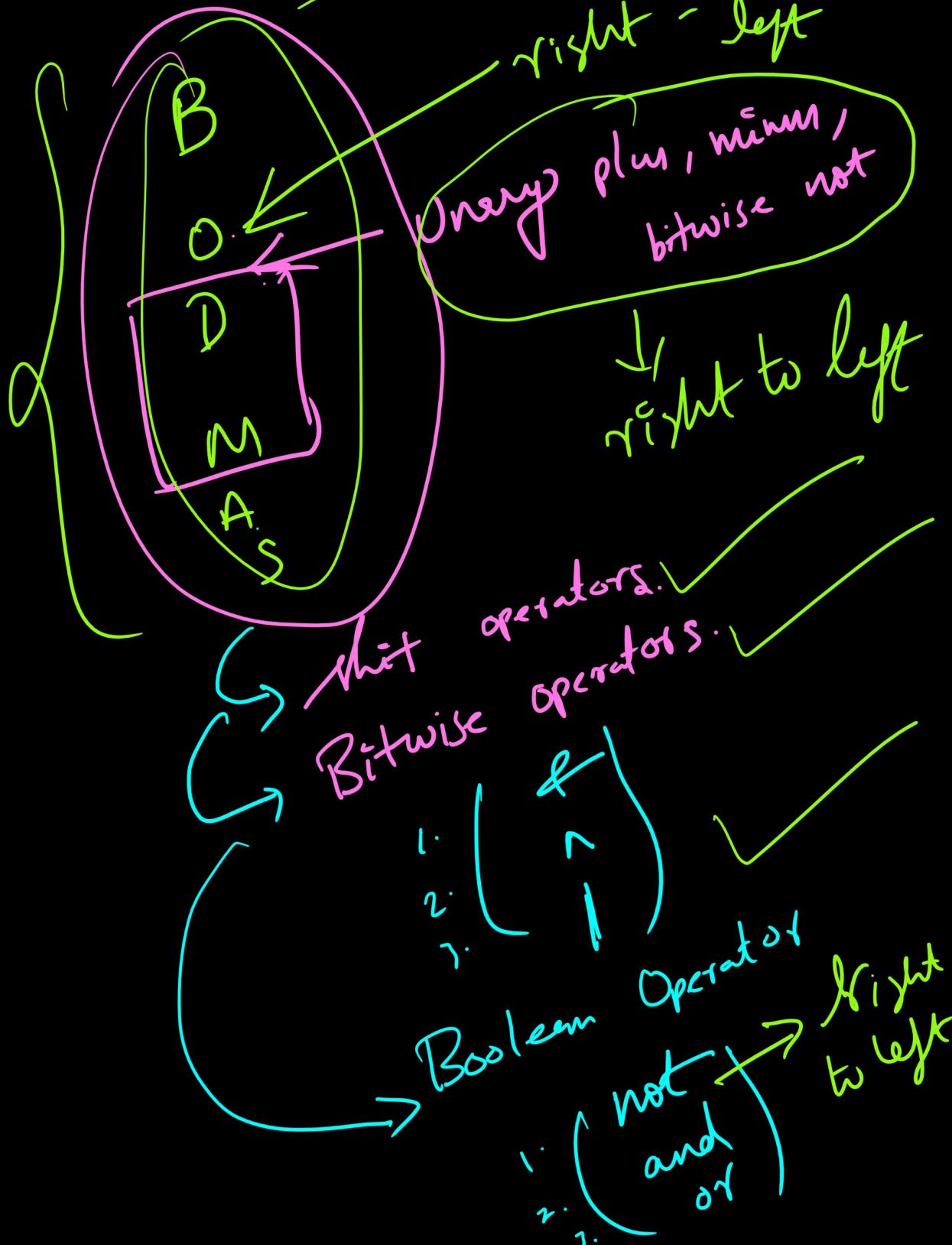
Left to
Right

13

Boolean or
or

Left to
Right

w.r.t to Grade



Unicode

tomorrow

Colab Class - 5:

<https://colab.research.google.com/drive/1Ok6U8iQBNJG6Y4Xm1-IkFotd4Lcsb2B5?usp=sharing>