

## Homework

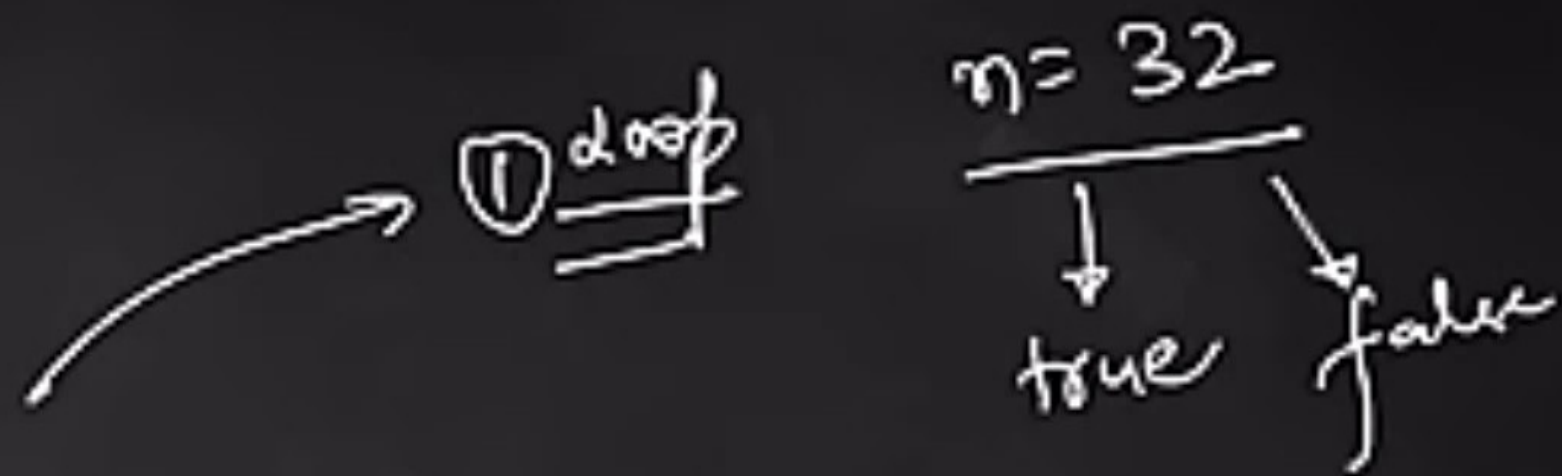


Figure out how to find if a number is power of 2 without any loop.

② Bits     $\ll, \gg$

WAF to reverse an Integer n.



# Bitwise Operators

same  $\rightarrow 0$   
dif  $\rightarrow 1$

$$\begin{array}{ll} 0^{\wedge}0 = 0 & 1^{\wedge}1 = 0 \\ 0^{\wedge}1 = 1 & 1^{\wedge}0 = 1 \end{array}$$

Bitwise &  
AND

011  
11

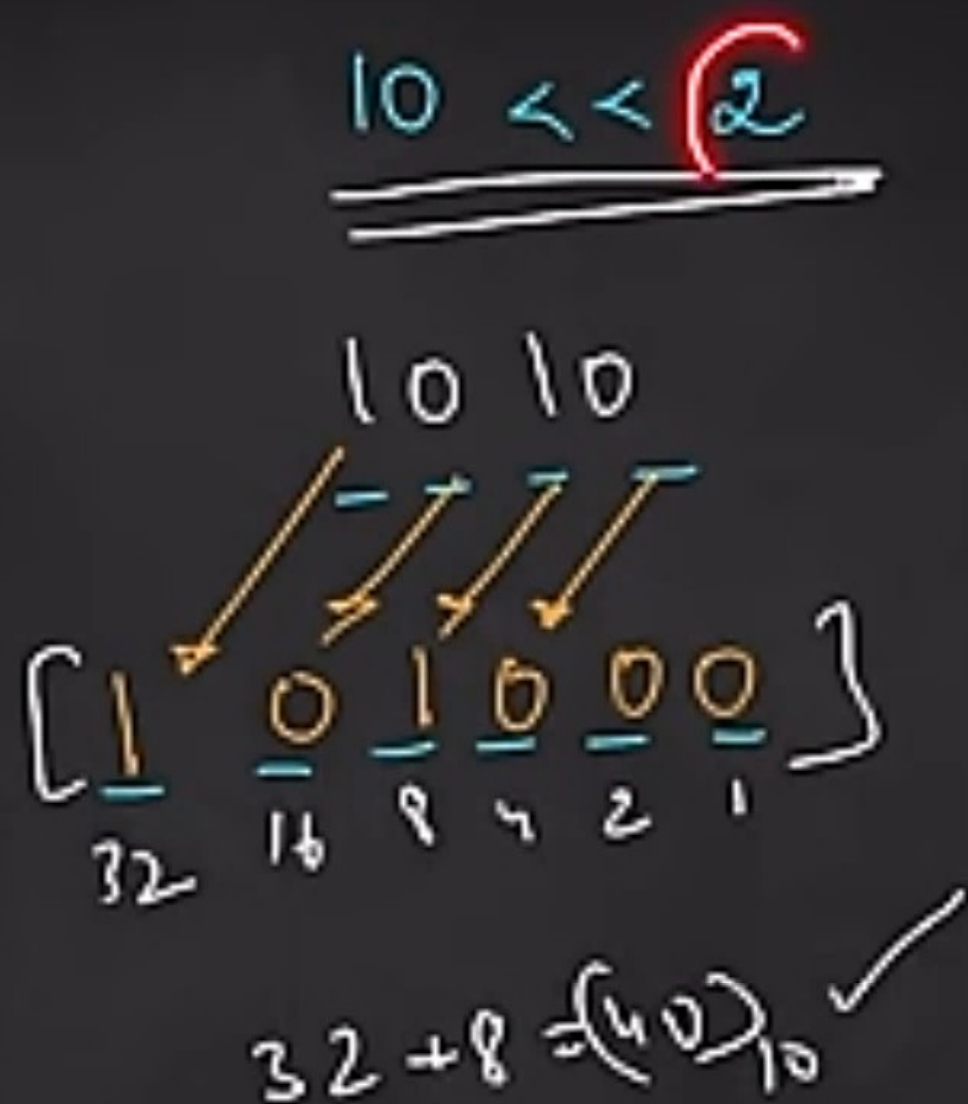
Bitwise |

Bitwise ^  
XOR  
[exclusive OR]



# Bitwise Operators

Bitwise  $\ll$   
Left shift op.



Bitwise  $\gg$   
Right shift op

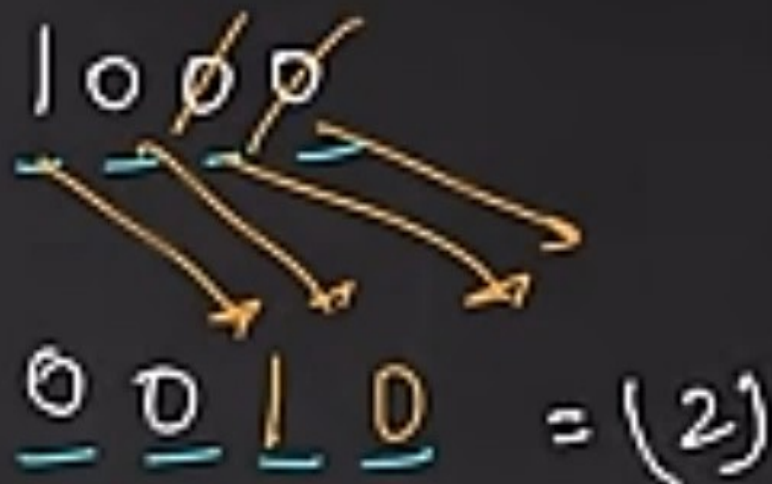




# Bitwise Operators

Bitwise <<  
Left shift op.

Bitwise >>  
Right shift op

$$8 \gg 2$$


The diagram illustrates the right shift operation. The top row shows the binary representation of 8 as 1000. The bottom row shows the result of shifting right by 2 positions as 0010, which is equal to the decimal value 2. Orange arrows indicate the movement of bits from the top row to the bottom row, showing the first two bits (1 and 0) being shifted out and the remaining two bits (0 and 0) moving two positions to the right.

$$\underline{0} \underline{0} \underline{1} \underline{0} = (2)$$



# Bitwise Operators

$$a \ll b$$

Bitwise  $\ll$   
Left shift op.

$$\begin{aligned} \text{ans} &= a * 2^b \\ &= 8 * 2^1 = \underline{16} \end{aligned}$$

$$\begin{aligned} \frac{8}{a} &\ll \frac{1}{b} \\ &\downarrow \\ 1000 &\downarrow \\ (10000)_2 &= (16)_{10} \end{aligned}$$

Bitwise  $\gg$   
Right shift op

$$\begin{aligned} a \gg b \\ \text{ans} &= a / 2^b \\ &= 8 / 2^1 \\ &= \underline{4} \end{aligned}$$

$$\begin{aligned} (8) \gg (1) \\ 1000 &\downarrow \\ (100) &= (4)_{10} \end{aligned}$$





# Homework

Solve for :

- $6 \& 10$
- $6 | 10$
- $6 \wedge 10$

Solve for :

- $10 \ll 2$  ✓
- $10 \gg 1$  ✓



# Operator Precedence

$$\text{ans} = 5 \ominus 2 \otimes 6 = 5 - (2 \times 6)$$

Operators	Precedence
!, +, - (unary operators)	first <i>R to L</i>
*, /, %	second <i>L to R</i>
+, -	third <i>L to R</i>
<, <=, >=, >	fourth <i>L to R</i>
==, !=	fifth <i>L to R</i>
&&	sixth <i>L to R</i>
	seventh <i>L to R</i>
= (assignment operator)	last <i>R to L</i>

Bitwise operations



# Scope

area variable  
↳ access / usage

↳ variables

↳ local

↳ global

if - else + loops

functions

Block of code

{ ✓ }





# Data Type Modifiers

*change meaning of data types*

$x = -10$   $\rightarrow$  2's complement

{ 1 - - - - - }

- long
- short
- long long
- signed
- unsigned

