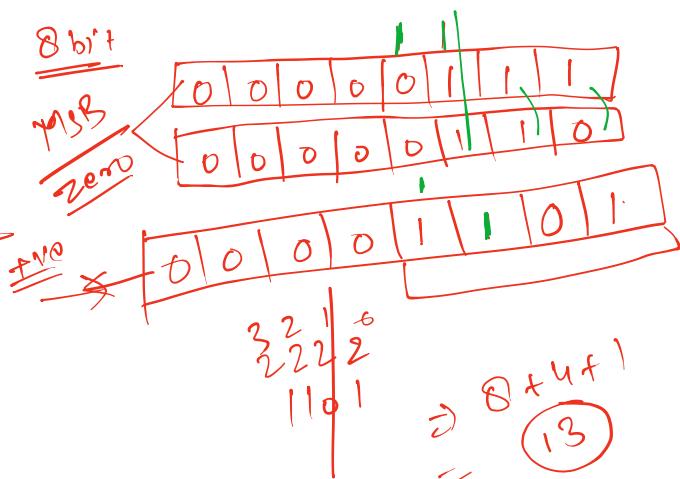


## Session 5

Monday, 14 July 2025 9:43 AM

$$\begin{array}{r} (+) 7 \\ + (+) 6 \\ \hline + 13 \end{array}$$



$$\begin{array}{r} 1 \\ + 1 \\ \hline 10 \\ + 1 \\ \hline 11 \end{array}$$

$$\begin{array}{r} (+) 11 \\ - (+) 14 \\ \hline \end{array}$$

$+11 + (-14)$

$\left\{ \begin{array}{l} (+) 11 - (+) 14 \\ (+) 11 - (-14) \end{array} \right.$

$\Rightarrow 1011$

$0000 1110$

$\xrightarrow{\text{MSB}}$

$11110010$

$\xrightarrow{\text{MSB}}$

$11111110$

$$\begin{array}{r} 1 \\ 00001011 \\ \hline \end{array}$$

$$\begin{array}{r} 1 \\ 10010010 \\ \hline \end{array}$$

$$\begin{array}{r} 1 \\ 10011101 \\ \hline \end{array}$$

$$\begin{array}{r} 1 \\ 11111101 \\ \hline \end{array}$$

$$-7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1$$

$$-128 + 64 + 32 + 16 + 8 + 4 + 1$$

$$-128 + 125 \Rightarrow -3$$

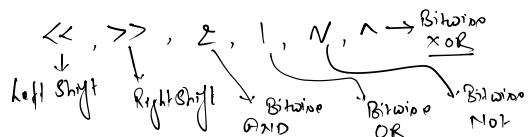
$$\begin{array}{r}
 +11 \\
 +(-)14 \\
 \hline
 -3
 \end{array}
 \quad
 \begin{array}{r}
 +0000|011 \\
 +11110010 \\
 \hline
 01111101
 \end{array}
 \quad
 \begin{array}{r}
 2|11 \\
 2|511 \\
 2|211 \\
 10
 \end{array}$$

$\leftarrow$

$$\begin{array}{r}
 00001110 \\
 = 11110010
 \end{array}
 \quad
 \begin{array}{l}
 2^7+2^6+2^5+2^4+2^3+2^2+2^0 \\
 -128+64+32+16+8+4+1 \\
 -128+125 \Rightarrow -3
 \end{array}$$

## BITWISE OPERATORS

Operators	Type of Operators	Operation Type
<code>++, --</code>	Increments/Decrements Operators	Unary Operator
<code>+, -, *, /, %</code>	Arithmetic Operators	
<code>&lt;, &lt;=, &gt;, &gt;=, ==, !=</code>	Relational Operators	
<code>&amp;,  , ^, ~</code>	Logical Operators	
<code>&amp;&amp;,   , &gt;&gt;, &lt;&lt;, ^</code>	Bitwise Operators	
<code>=, +=, -=, *=, /=, %=</code>	Assignment Operators	
<code>sizeof()</code> , <code>*</code>	Special Operators	
<code>?:</code>	Ternary or Conditional Operator	Ternary Operator



Used to Perform Manipulation @ Bit Level.

AND TruthTable

int `x = 14;`  
`y = 15;`      Bitwise AND  
`z = x & y;`       $z = \begin{array}{l} 00001110 \\ 00001111 \\ \hline 00001110 \end{array}$        $14$   
 $z = 14$        $14 = 0110$        $0 + 4 + 2 = 14$

int `x = 11;`  
`y = 12;`      Bitwise AND  
`z = x & y;`       $z = \begin{array}{l} 00001011 \\ 00001100 \\ \hline 00001111 \end{array}$   
 $z = 15$        $15 = 1101$        $1 + 4 + 2 + 1 = 15$

$z = \begin{array}{l} 1011 \\ 1101 \\ \hline 1000 \end{array} \rightarrow 8$

int `x = 11;`  
`y = -12;`  
`z = x & y;`

$\Rightarrow$  2 Bytes

$\begin{array}{l} 11110100 \\ 00001100 \\ \hline 11110100 \end{array}$

$\Rightarrow$   $z = \begin{array}{l} 1111111111111000 \\ 0000000000000101 \end{array}$

$\begin{array}{c} 1111 \\ \text{Zero} \\ \text{---} \\ -1 \end{array}$   
 $\leftarrow 0000000000000000$

$$\begin{array}{l}
 \stackrel{11}{=} \Rightarrow \quad \begin{array}{c} 00000|011 \\ | | | | | \end{array} \quad \boxed{\text{011}} \\
 \stackrel{-11}{=} \Rightarrow \quad \begin{array}{c} | | | | 010 \\ | | | | | \end{array} \quad \boxed{\text{010}}
 \end{array}$$

~~XOR~~  $\Rightarrow$  ~~Truth Table of XOR~~

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

Truth Statement:  
 if inputs Different  $\rightarrow 1$   
 if Inputs Same  $\rightarrow 0$   
 if Input is zero  $\rightarrow 0$

$$\begin{array}{l}
 \text{int } x = 12; \\
 \text{int } y = 11; \\
 \text{int } z = x \wedge y;
 \end{array}$$

$\begin{array}{c} 00000|100 \\ 00000|011 \\ \hline 00000|011 \end{array}$

$\textcircled{7} \Rightarrow \underline{\underline{0000011}}$

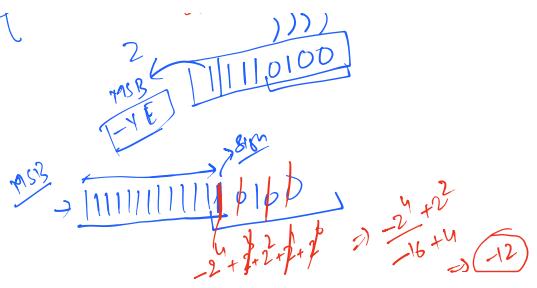
$\text{~N} (\text{Bitwise Not})$

~~int x = 11;~~      ~~it alters the~~  
~~int z = ~x;~~      ~~Int will ones complement~~

~~16~~  $\rightarrow \textcircled{11}$

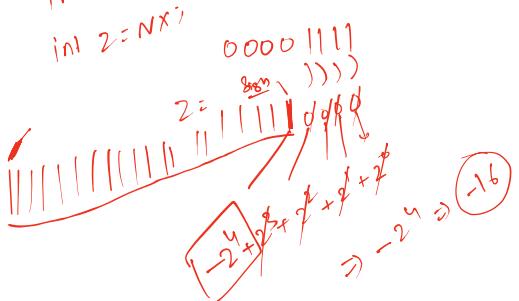
~~0000000000000000|011~~  
~~(11)~~  
~~M11111111111111110110~~  
~~23456789~~  
~~-16 + 8 + 4 +~~  
~~= -12~~

$\textcircled{2} = \text{NX}$   
 $\begin{array}{c} 1111 \\ + \textcircled{11} \\ \hline 0000 |011 \end{array}$



int n=15;

$$\ln 2 = Nx$$

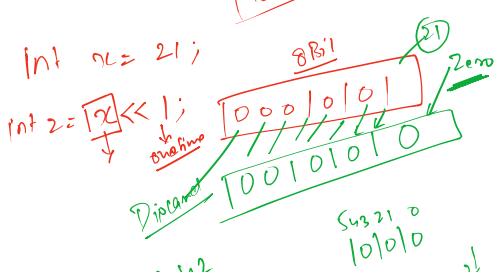


The diagram illustrates the long division of 15940 by 594. It shows the quotient 20 above the dividend 15940. The remainder 10 is circled. The process involves shifting digits and performing subtraction. Arrows indicate the movement of digits and the subtraction step.

If I am Shift any Number.

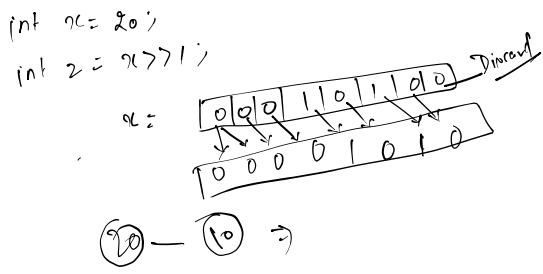
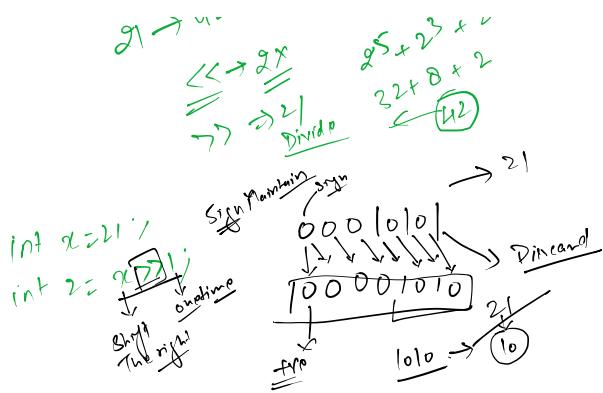
I place @ left side then the Value

In Increasab  Time

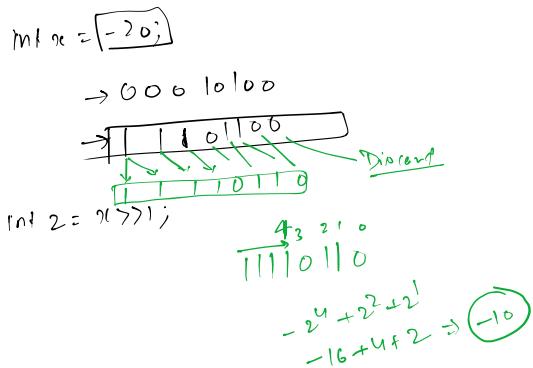


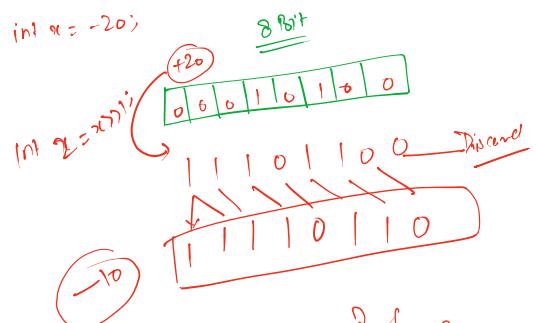
$$49 \begin{cases} \nearrow \\ \searrow \end{cases} 49$$

(10)  
80°

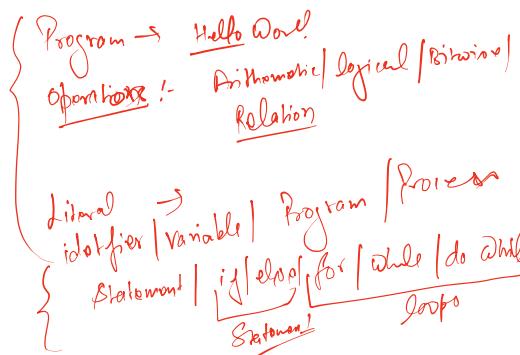
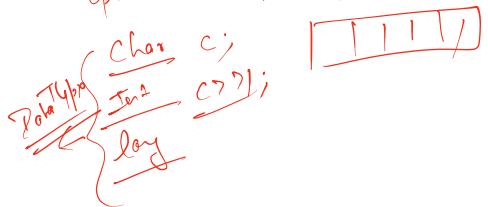


$x \gg 2;$   
 $20 \rightarrow 10 \rightarrow 5$

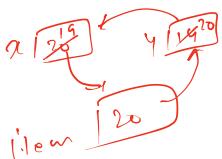
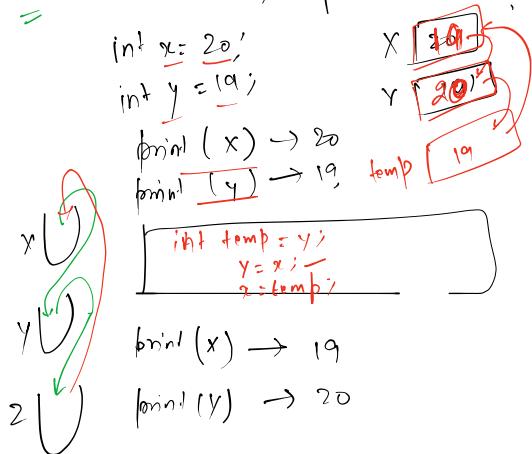




→ Bitwise operator can perform operation only on integral type.



Q WAP to swap two values?



Q Print those Number which is  
multiple of  $\boxed{5}$ ;

#include <stdio.h>  
int main() {  
 ② //  $\boxed{100}$   
 int x=1;  
 int n=100;  
 while(x<=100) {  
 if( $\boxed{x/5} == \boxed{0}$ )  
 printf(" %d ", x);  
 x++;  
 }  
}

Gives me the  
Roman value  
of numbers.