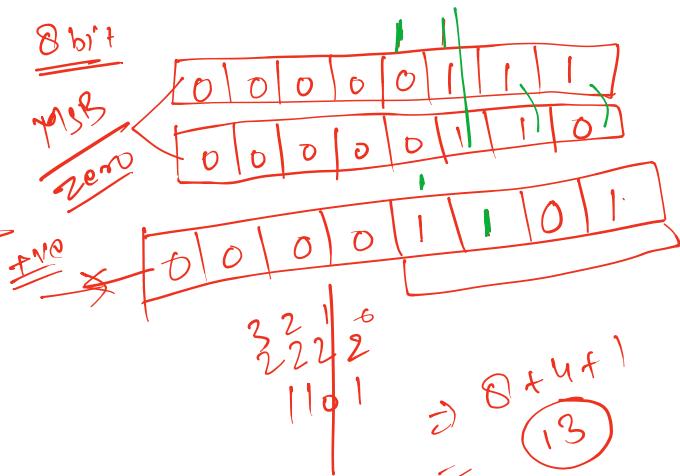


Session 5

Monday, 14 July 2025 9:43 AM

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



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

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

$+ 11 + (-14)$

$\{ (+) 11 - (+) 14 \rightarrow -3$

$+ 11 - 14 \rightarrow -3$

$0000\ 1110$

11110010

MSB

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

$$\begin{array}{r} 10110010 \end{array}$$

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

$$\begin{array}{r} 11111101 \\ 76543210 \end{array}$$

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

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

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

$$\begin{array}{r}
 +11 \\
 +(-)14 \\
 \hline
 -3
 \end{array}
 \quad
 \begin{array}{r}
 +00001011 \\
 +11110010 \\
 \hline
 01111101
 \end{array}
 \quad
 \begin{array}{r}
 2 | 11 \\
 2 | 5 11 \\
 2 | 2 11 \\
 \hline
 1 0
 \end{array}$$

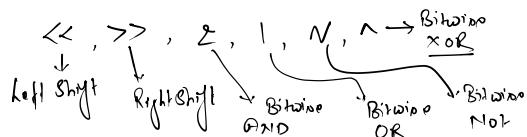
\leftarrow

$$\begin{array}{r}
 00001110 \\
 = 11110010
 \end{array}$$

$2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^0$
 $-128 + 64 + 32 + 16 + 8 + 4 + 1$
 $-128 + 125 \Rightarrow -3$

Operations

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



$$\begin{array}{l}
 \text{int } x = 11; \\
 \text{int } y = 12; \\
 z = x / y; \\
 z = \boxed{15}
 \end{array}
 \quad
 \begin{array}{r}
 0000\ 1011 \\
 0000\ 1100 \\
 \hline
 0\ 0001111
 \end{array}$$

$$2 \frac{121}{110} \rightarrow 0$$

The diagram illustrates the addition of two 16-bit integers, $x = 11$ and $y = -12$, resulting in $z = -1$.

Binary Representation:

- int x = 11;** is represented as 0000000000001011 .
- int y = -12;** is represented as 1111111111111000 .
- int z = x + y;** is represented as 1111111111111001 .

Addition Process:

The addition is performed using binary arithmetic. The sum is calculated as follows:

$$\begin{array}{r}
 0000000000001011 \\
 + 1111111111111000 \\
 \hline
 1111111111111001
 \end{array}$$

The result is 1111111111111001 , which corresponds to the value -1 in two's complement form.

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

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

~~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 $\rightarrow 0$

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

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

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

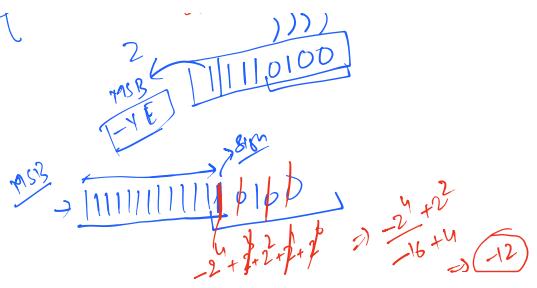
~N (Bitwise Not)

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

$\text{int } z = \cancel{\text{~}}\overset{16}{x} \rightarrow \text{11}$

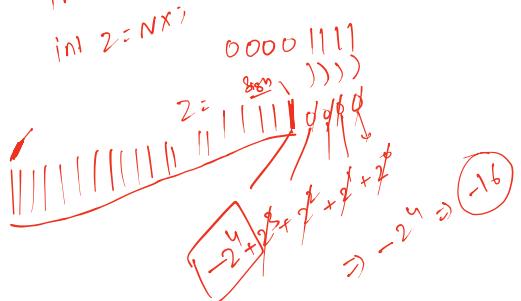
~~0000000000000000|011~~
~~(1111111111111111|011)~~
~~1111111111111111|011~~
~~^ ^ ^ ^ ^~~
~~- 16 + 8 + 4 +~~
~~= -12~~

$\text{2} \stackrel{111}{=} \text{~N} \times$
 $\begin{array}{c} + \text{11} \\ \hline 10000|011 \end{array}$



int n=15;

$$\ln z = Nx$$

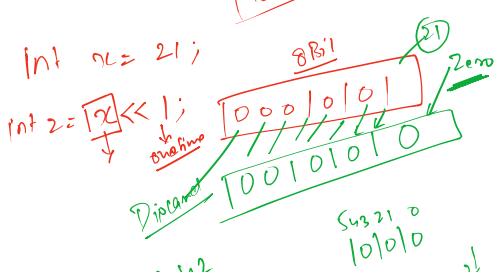


The diagram illustrates the division of 15940 by 594 using long division. It shows the quotient 20.00 and the remainder 1594. The remainders are then converted into fractions: $\frac{1594}{10}$, $\frac{1594}{100}$, and $\frac{1594}{1000}$.

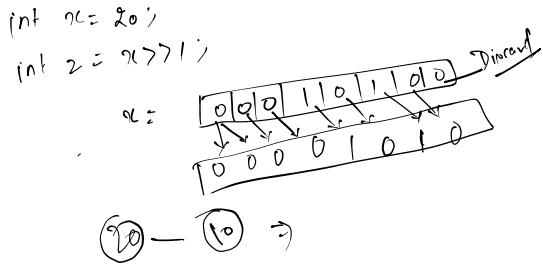
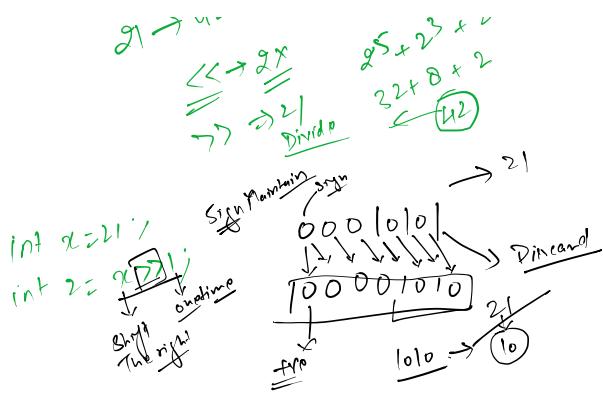
If I am Shift any Number

I place @ left side then the Value

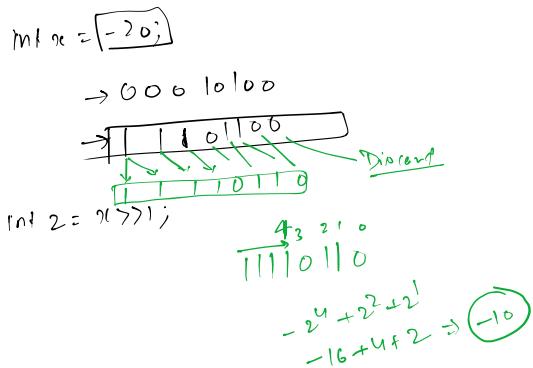
In Increasab time

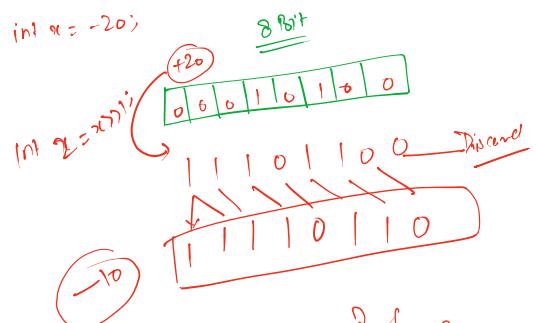


$$49 \begin{cases} 10 \\ 80^\circ \end{cases}$$

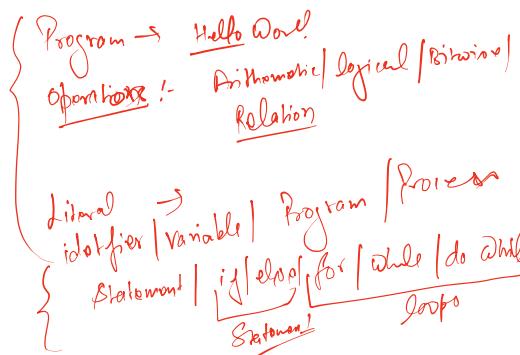
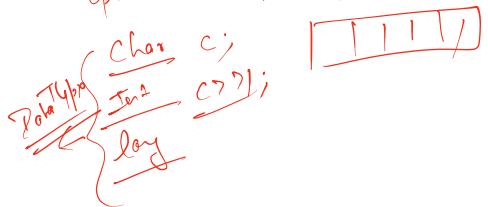


$x \gg 2;$
 $20 \rightarrow 10 \rightarrow 5$

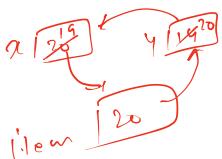
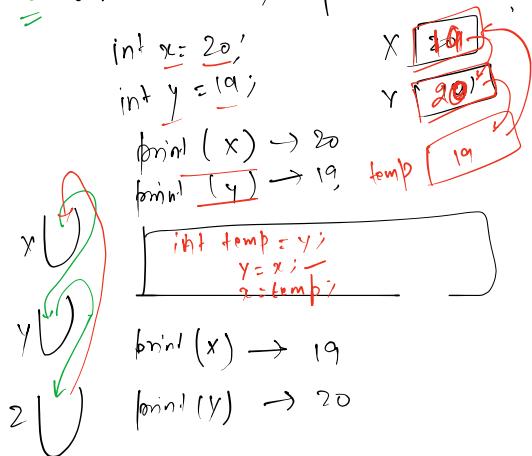




→ Bitwise operator can perform operation only on integral type.



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.