**ASSIGNMENT ON OPERATORS**

**Bitwise Operators:**

Bitwise operators are operators (just like +, \*, &&, etc.) that operate on int and units at the binary level. This means they look directly at the binary digits or bits of an integer.

**List of Bitwise Operators:**

* & (bitwise AND)
* | (bitwise OR)
* ~ (bitwise NOT)
* ^ (bitwise XOR)
* << (bitwise left shift)
* >> (bitwise right shift)

**Example for Bitwise Operators**

#include <stdio.h>

int main()

{

int a = 20; /\* 20 = 010100 \*/

int b = 21; /\* 21 = 010101 \*/

int c = 0;

c = a & b; /\* 20 = 010100 \*/

printf("AND - Value of c is %d\n", c );

c = a | b; /\* 21 = 010101 \*/

printf("OR - Value of c is %d\n", c );

c = a ^ b; /\* 1 = 0001 \*/

printf("Exclusive-OR - Value of c is %d\n", c );

getch();

}

**Output**

AND- value of c is 20

OR- value of c is 21

XOR- value of c is

**Conditional or Ternary Operators :**

The conditional operator is kind of similar to the [if-else statement](https://www.geeksforgeeks.org/decision-making-c-c-else-nested-else/) as it does follow the same algorithm as of [if-else statement](https://www.geeksforgeeks.org/decision-making-c-c-else-nested-else/) but the conditional operator takes less space and helps to write the if-else statements in the shortest way possible.

**Syntax**

The conditional operator is of the form

variable = Expression1 ? Expression2 : Expression3

It can be visualized into if-else statement as:

if(Expression1)

{

variable = Expression2;

}

else

{

variable = Expression3;

}

**Example**

int a = 10, b = 20, c;

if (a < b) {

c = a;

}

else {

c = b;

}

printf("%d", c);