Bitwise operator-

Bitwise operators are part of Binary operators that operates with two operands. These are used for manipulating data at the bit level. some bitwise operators are-

1.&(bitwise and)- syntax a&b.

2.|(bitwise or)-syntax a|b.

3.^(Bitwise exclusive or)-syntax a^b.

4.<<,>>(bitwise shift operators)-syntax a<<b,a>>b.

5.~(Bitwise complement operator)-syntax a~b.

Example-

#include <stdio.h>

int main()

{

char a = 5, b = 9;

printf("a = %d, b = %d\n", a, b);

printf("a&b = %d\n", a & b);

printf("a|b = %d\n", a | b);

printf("a^b = %d\n", a ^ b);

printf("~a = %d\n", a = ~a);

printf("b<<1 = %d\n", b <<1);

printf("b>>1 = %d\n", b >> 1);

return 0;}

Output:

a=5,b=9

a&b=1

a|b=13

a^b=12

~a=250

b>1=4

these are executed in binary language bit by bit and output is shown in numbers.

Ternary operator-

It is used to execute code based on the result of a binary condition. It takes binary condition as a input. It can not be used to execute a code. It must be either returned in a function ,or set equal to variable with the same data type as the returned value.

Syntax-

Variable= expression 1 ? expression2 : expression3

Example-

#include <stdio.h>

int main()

{

int n1 = 5, n2 = 10, max;

max = (n1 > n2) ? n1 : n2;

printf("Largest number between"

" %d and %d is %d. ",

n1, n2, max);

return 0;

}

Output-Largest number between 5 and 10 is 10.