23-2-26 BCA Sem IV

Bitwise Operator in C Programming.

The bitwise operators are the operators used to perform the operations on the data at the bit-level. When we perform the bitwise operations, then it is also known as bit-level programming. It consists of two digits, either 0 or 1. It is mainly used in numerical computations to make the calculations faster.

There are different types of bitwise operators in the C programming language. Following are the list of bitwise operators:—

Operator	Meaning of operator
&	Bitwise AND operator
|	Bitwise OR operator
^	Bitwise exclusive OR operator
~	One's complement operator (It is an unary operator)
<<	Left shift operator
>>	Right shift operator.

Now let's look at the truth table of these operators:—

X	Y	X & Y	X | Y	X ^ Y
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Bitwise AND is denoted by &. Two integer operands are written on both sides of the & operator. If the corresponding bits of both are 1, then the output of bitwise AND is 1; otherwise the output will be 0.

For example,
We have a = 4, and b = 7 are two variables.
The binary representation will be :—

a = 0100
b = 0111

When we apply the bitwise AND operation in the above two variables, then

a & b would be 0100.

That means a & b would be 7.

Let us understand the bitwise AND operator through the following C Program: —

```
# include<stdio.h>
int   main()
 {
        int a, b;
    printf("\n Enter two integers :");
    scanf(" %d %d", &a, &b);
    printf(" \n The Output of a & b is  %d", a&b);
    return 0;
 }
```

When we run the above example if we give two integers as :- 6 and 14 respectively, then we will get the output of 6.

Explanation:-

$$\begin{array}{rl} a = 6 = & 0110 \\ b = 14 = & 1110 \\ \hline a \& b \quad = & 0110 = 6 \end{array}$$

Bitwise OR operator is represented by a single vertical line |. Two integer operands are written on both sides of the | symbol. If the bit value of any of the operand is 1, then the output would be 1, otherwise 0.

For example :-

We consider two variables,

$a = 23; \quad b = 10;$

The binary representation of the two integers would be

```
 a  = 00010111
 b  = 00001010
-----------------
a|b = 00011111
```

that means 31.

Now, let us understand the bitwise OR operator through a program :-

```
#include <stdio.h>
int main() {
        int a = 23;
        int b = 10;
     printf("a|b = %d", a|b);
        return 0;
        }
```

When we run the program :-

$\boxed{a|b = 31}$ will be shown as output-