

## C-18 $\Rightarrow$ Operators in C - Part 6

### [Bitwise Operators]

Bit:

Smallest unit to store some data in computer memory.

Byte:

1 byte = 8 bits.

### Types of Bitwise Operators:

- \*  $\&$   $\rightarrow$  Bitwise AND
- \*  $|$   $\rightarrow$  Bitwise OR
- \*  $\wedge$   $\rightarrow$  Bitwise XOR
- \*  $\sim$   $\rightarrow$  Bitwise NOT
- \*  $\ll$   $\rightarrow$  Bitwise Left Shift
- \*  $\gg$   $\rightarrow$  Bitwise Right Shift

\* Bitwise operators need or performs operations in bit level.

\* Whereas normal operators like  $++$ ,  $++$ ,  $!$  need minimum memory space of 8 bits or 1 byte to perform operation.

\* In bitwise operators, the data first converted to binary form and we get a output rather than true or false.

Bitwise AND (&) [performs bitwise AND]

Ex: int a=10, b=5;

c = a & b

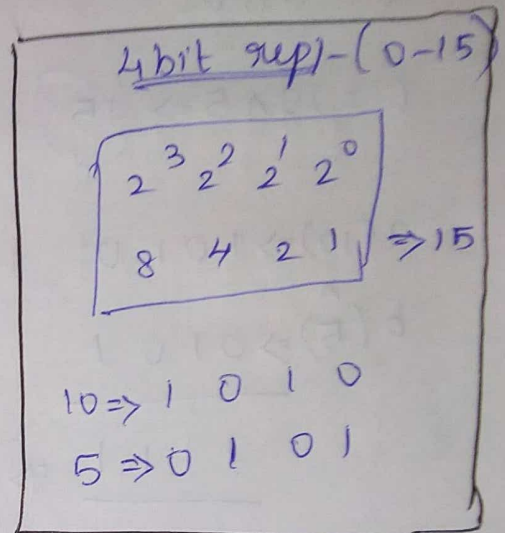
c  $\Rightarrow$  10 & 5  $\Rightarrow$  0

10  $\Rightarrow$  1 0 1 0

&

5  $\Rightarrow$  0 1 0 1

0 0 0 0  $\Rightarrow$  0



Bitwise OR (|) [performs bitwise OR]

Ex:

int a=10, b=5;

c = a | b

c = 10 | 5  $\Rightarrow$  15

10  $\Rightarrow$  1 0 1 0

|

5  $\Rightarrow$  0 1 0 1

1 1 1 1  $\Rightarrow$  15

## Bitwise XOR (^)

[performs logical XOR]

Ex: int a=10, b=5;

c = a & b

c = 10 & 5  $\Rightarrow$  15

a(10)  $\Rightarrow$  1010

b(5)  $\Rightarrow$  0101

1111  $\Rightarrow$  15

## Truth Table

A	B	A ^ B
0	0	0
0	1	1
1	0	1
1	1	0

Difference  
B/w

A	B	A ^ B
1	1	1

OR & XOR

## Program 1

main()

{

int a=10, b=5;

clrscr();

printf("%d", a & b);  $\Rightarrow$  2

printf("%d", a | b);  $\Rightarrow$  14

printf("%d", a ^ b);  $\Rightarrow$  15

printf("%d", a & b & b+1 || 0);  $\Rightarrow$  1

}

2 & 5 || 0

$\downarrow$   
1 || 0  $\Rightarrow$  1



$$\begin{array}{r}
 a \& b \Rightarrow \\
 10 \& 6 \\
 \hline
 0010
 \end{array}$$

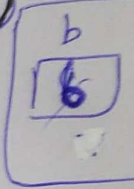
$$\begin{array}{r}
 0010 \\
 \hline
 \Rightarrow 2
 \end{array}$$

$$\begin{array}{r}
 a | b \Rightarrow \\
 10 | 6 \\
 \hline
 1110 \Rightarrow 14
 \end{array}$$

$$\begin{array}{r}
 a \wedge b \Rightarrow \\
 10 \wedge 6 \\
 \hline
 0110 \\
 \hline
 1100 \\
 \hline
 \downarrow \\
 12
 \end{array}$$

Set 1 Program 2:

printf ("%d", a & b && b + 1 || 0 || b++);  $\Rightarrow$  ①



$$\begin{array}{r}
 2 \&\& 7 || 0 || 6 \\
 \downarrow \\
 1 || 0 || 6
 \end{array}$$

First is  
1 is Logical OR;  
So it will not  
check next.

printf ("%d", b);  $\Rightarrow$  ⑥

Set 2 Program 3:

printf ("%d", a & b && b + 1 || 0 && b++); ①



$$\begin{array}{r}
 2 \&\& 7 || 0 \&\& 6 \\
 \downarrow \\
 1 || 0 \Rightarrow 1
 \end{array}$$

It won't evaluate since false

Program 4: Set (3)

a = 10    b = 6

*(2 88 7) → 1 (true, so work with high priority)*  
printf("%d", a & b    88    b + 1    ||    1    88    b ++); ⇒ ①  
printf("%d", b); ⇒ ⑥

Program 5: set (4)

*(2 88 0) → 0*  
printf("%d", a & b    88    0    ||    1    88    b ++); ⇒ ①  
printf("%d", b); ⇒ ⑦  
*0 || 1 88 b ++  
          ↓  
          higher precedence*

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int a=10,b=6;
7      printf("%d\n",a&b);
8      printf("%d\n",a|b);
9      printf("%d\n",a^b);
10     printf("%d\n",a&b && b+1 || 0);
11 }
12

```

"D:\COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART - 5- Jennys Lectures\BITWISE OPERATORS\BITWISE 1\bin\Debug\BITWISE 1...

```

2
14
12
1
Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.

```

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int a=10,b=6;
7      printf("%d\n",a&b);
8      printf("%d\n",a|b);
9      printf("%d\n",a^b);
10     printf("%d\n",a&b && b+1 || 0 || b++);
11     printf("%d\n",b);
12 }
13

```

"D:\COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART - 5- Jennys Lectures\BITWISE OPERATORS\BITWISE 2\bin\Debug\BITWISE 2...

2  
14  
12  
1  
6

Process returned 0 (0x0) execution time : 0.057 s  
Press any key to continue.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int a=10,b=6;
7      printf("%d\n",a&b);
8      printf("%d\n",a|b);
9      printf("%d\n",a^b);
10     printf("%d\n",a&b && b+1 || 0 && b++);
11     printf("%d\n",b);
12 }
13
```

"D:\COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART - 5- Jennys Lectures\BITWISE OPERATORS\BITWISE 3\bin\Debug\BITWISE 3...

2  
14  
12  
1  
6

Process returned 0 (0x0) execution time : 0.016 s  
Press any key to continue.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int a=10,b=6;
7      printf("%d\n",a&b);
8      printf("%d\n",a|b);
9      printf("%d\n",a^b);
10     printf("%d\n",a&b && b+1 || 1 && b++);
11     printf("%d\n",b);
12 }
13

```

"D:\COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART - 5- Jennys Lectures\BITWISE OPERATORS\BITWISE 4\bin\Debug\BITWISE 4...

```

2
14
12
1
6
Process returned 0 (0x0)   execution time : 0.016 s
Press any key to continue.

```

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int a=10,b=6;
7      printf("%d\n",a&b);
8      printf("%d\n",a|b);
9      printf("%d\n",a^b);
10     printf("%d\n",a&b && 0 || 1 && b++);
11     printf("%d\n",b);
12 }
13

```



"D:\COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART - 5- Jennys Lectures\BITWISE OPERATORS\BITWISE 5\bin\Debug\BITWISE 5... -

2  
14  
12  
1  
7

Process returned 0 (0x0) execution time : 0.047 s  
Press any key to continue.