1. ( **i** ) Bitwise operators are used for **changing individual bits in an operand.** The operators are similar to logical operators, except that they work on a smaller scale i.e. on bits instead bytes.

|  |  |
| --- | --- |
| **OPERATORS** | SYMBOLS |
| & | Bitwise AND |
| | | Bitwise OR |
| ^ | Bitwise XOR |
| ~ | Bitwise complement |
| << | Shift left |
| >> | Shift right |

Syntax: int a = 12, b = 25, c ;

c = a&b;

printf(“output is =”+c) ;

Example: 12 = 00001100 (In Binary)

25 = 00011001 (In Binary)

Bit Operation of 12 and 25

00001100

& 00011001

\_\_\_\_\_\_\_\_

00001000 = 8 (In decimal)

So output is 8.

**( ii )** Ternary operator is like if\_then. It takes three operands.

The first one is condition, second is the value to be assigned if condition turns to be true, third is also the value to be assigned if condition is false.

Syntax: *datatype variablename= condition? value1: value2;*

Example: int age=20;

boolean iseighteenover =

(age of client>18)?true:false;

**2.**

Source code:

#include<stdio.h>

int main()

{

int num1,num2;

printf("Enter your first number: ");

scanf("%d",&num1);

printf("Enter your second number: ");

scanf("%d",&num2);

int sum,difference,product,division,remainder;

sum=num1+num2;

difference=num1-num2;

product=num1\*num2;

division=num1/num2;

remainder=(num1%num2);

printf\n("sum of the numbers you entered is: %d ",sum);

printf\n("difference of the numbers you entered is: %d",difference);

printf\n("product of the numbers you entered is: %d",product);

printf\n("division of the numbers you entered is: %d",division);

printf\n("remainder of the numbers you entered is: %d",remainder);

return 0;

}

Output:

Enter your first number: 4

Enter your second number: 5

sum of the numbers you entered is: 9 difference of the numbers you entered is: -1product of the numbers you entered is: 20division of the numbers you entered is: 0remainder of the numbers you entered is: 4

Process returned 0 (0x0) execution time : 2.972 s

Press any key to continue.