OPERATORS

# Question :1

## 1.Bitwise operators:

Bitwise operators are special operator set provided by C.

They are used in bit level programming. These operators are used to manipulate bits of an integer expression. These are used to change individual bits in an operand.

These operators are used to perform bit operations. Decimal values are converted into binary values and bitwise operators work on these bits.

Bitwise operators in C language are & (bitwise AND), | (bitwise OR), ~ (bitwise NOT), ^ (XOR), <<(left shift), >> (right shift).

## Example:

#include<stdio.h>

int main ()

{

int a=12, b=25:

print f (“output=%d”, a & b);

return 0;

}

## Output:

Output =8

# 2.Ternary operators:

A ternary operator allows you to assign one value to the variable if the condition is true, and another value if the condition is false.

It is used to execute code based on the result of a binary condition. The name ternary refers to the fact that operator takes three operands.

This operator takes three arguments.

1.The first argument is a comparison argument.

2.The second one is the result upon a true comparison.

3.The third one is the result upon a false comparison.

Example:

int a=10, b=20, c;

if(a<b) {

c=a;

}

Else {

C= b;

}

Print f (“%d”, c);

The above example takes more than 10 lines.

By using ternary operator we can write this program in just 3 lines.

Syntax:

Value -if-true: value-if-false

Using ternary operator the above example can be rewritten as:

Int a=10, b=20, c;

C = (a<b) a: b;

Print f (“%d”, c);

## Output:

10

# Question2:

#include<stdio.h>

Main () {

Int a=21;

Int b=10;

Int c;

C=a +b;

Print f (“sum of 21 and 10 is %d/n”, c);

C=a-b;

Print f (“difference of 21 and 10 is %d/n”, c);

C=a\*b;

Print f (“multiple of 21 and 10 is %d/n”, c);

C=a/b;

Print f (“division of 21 and 10 is %d/n”, c);

}

return 0;

## Output:

Sum of 21 and 10 is 31

Difference of 21 and 10 is 11

Multiple of 21 and 10 is 210

Division of 21 and 10 is 2