OPERATOR (O/P)

1. AIRTHMATIC O/P

+ , -, * , /, % (mod) (remainder)

2. RELATIONAL O/P

< , > , <= , >= , != , == (equal to) compare a == 5

3. ASSIGNMENT O/P

= (copy) a = 5

4. INCREMENT AND DECREMENT O/P

++ --

int a = 3 ;

a++ ---> a = a + 1 -- > a = 4

a-- ---> a = a - 1 ---> a = 3

5. AIRTHMATIC ASSIGNMENT O/P

+= , -= , *= , /= , %=

int a = 3 ; a = a + 5 ---> a += 5 ; a = 8

int a = 3 ; a = a - 5 ---> a -= 5 ; a = -2

int a = 3 ; a = a * 5 ---> a *= 5 ; a = 15

int a = 3 ; a = a / 5 ---> a /= 5 ; a = 0

int a = 3 ; a = a % 5 ---> a %= 5 ; a = 3

---

6. LOGICAL O/P

&& :- and

|| :- or (| :- pipe or vertical bar)

! :- not

5 --> 000000101 (8 BIT) --> BYTE

7. BITWISE O/P

& :- bitwise and

| :- bitwise or

~ :- one's complement (~ :- tilde)

8. SPECIAL OPERATOR

a) , (COMMA) OPERATOR

b) sizeof() operator

c) ?: ternary operator

---

a + b * c

OPEREND --> a , b , c = n

OPERATOR --> + , * = (n - 1)

---

a + b infix

+ab prefix

ab+ postfix

CATEGORY OF OPERATOR

1. **UNARY OPERATOR**

   NEED ONE OPEREND

   + , - , ~ ,++ , -- e.g. - 5

2. **BINARY OPERATOR**

   NEED TWO OPEREND e.g. 5 - 3

   +, -, * , < , > , && , += , ...

3. **TERNARY OPERATOR**

   NEED THREE OPEREND

   ?: (ternary) operator

SPECAIL OPERATOR :- , (comma) operato

ADDITION OF TWO NOS USING COMMA OPERATOR

```
#include<stdio.h>
  int   main()
  {     int   a , b , c ;   // DATA SEPERATION
        c = ( a = 3 , b = 2 , a + b ) ;// MULTIPLE STATEMENTS
        printf(" SUM =  %d \n " , c);
  }
```

2.

```
#include<stdio.h>
int    main()
{          int    a = 19, 20 , 30  ;
           printf(" a = %d \n " , a);
}
```

1. int a ;
   a = (10 , 20 , 80) ;
   printf("%d" , a);
   ans :- a = 80

2. int a ;
   a = 10 , 20 , 80 ;
   printf("%d" , a);
   ans :- a = 10

3. int a = 10 , 20 , 80 ; // = first then , second
   printf("%d" , a);
   ans :- error

4. int 20, 30 ;
   error

sizeof() operator

syntax

1. sizeof(data type);
2. sizeof(variable);
3. sizeof(constant);

FIND SIZE OF THE DATA TYPE

```
#include<stdio.h>
  int main()
  {
      printf(" sizeof integer = %d \n" , sizeof(int)  );   // 2 or 4
      printf(" sizeof float   = %d \n" , sizeof(float)); //  4
      printf(" sizeof char    = %d \n" , sizeof(char) ); //  1

printf(" sizeof integer constant    = %d \n" , sizeof(12));   // 2 or 4
printf(" sizeof float    constant   = %d \n" , sizeof(3.456)); // 8
printf(" sizeof float    constant   = %d \n" , sizeof(3.456f)); // 4

printf(" sizeof  char     constant   = %d \n" , sizeof('A')); // 1
  }
```