C_14 => operators in C - Part 2

① Arithmetic operators :-

* used to perform arithmetic operations.

+, -, /, *, %

* % operator (MOD) → to get remainder.

↓

(*) used only for integer values

* not for float

Eg:- int a = 4 b = 4

```
    0 - Q
4 ) 4
    4
   ---
    0 → REM
   ---
```

```
#include <stdio.h>
#include <conio.h>
Void main()
{
int a=10; b=7;
clrscr();
printf("a+b = %d", a+b);   → 17
printf("a-b = %d", a-b);   → 3
printf("a*b = %d", a*b);   → 70
printf("a/b = %d", a/b);   → 1
printf("a%b = %d", a%b);   → 3.
getch();
}
```

Ex①

7) 10 (1.
7
3 →

Ex②.

```
float a=10; b=7;
printf("a+b = %f", a+b);   → 17.000000
                              6 digits after point
printf("a/b = %f", a/b)    → 1.4----
```

Note ③

~~printf("a%b = %f", a%b)~~

% operator won't work in float values.

Note ① → [% operator depends on sign.]

Ex ③ a = 10, b = -7

+a % b ⇒ 7 | 10 | 1 | 7 | (+3)

Ex ④ a = -10, b = 7

-a % b ⇒ 7 | -10 | 1 | 7 | (-3)

Note ②

Rem & Result

* Sign depends on the sign of first operand

Note: 4 ⇒ we can use one int datatype & other float datatype; but the result should be in float (%.f) should not be in (%.d).

Ex ⑤

```
int a = 1;
float b = 7;
c = ("a+b = %.f", a+b, c);
```

Precedence of operators : [overview - separate topic]

→ * / % — ① Higher Precedence

→ + − — ② Lower Precedence

Associativity:

→ Same precedence; so left to right

a + b − c

(left to right)

* L Value should be a variable.
* R Value should be expression

In assignment operator

Eg ①

L.H.S = R.H.S

a = 5 * 4; ✓

a ← 20

a + b = 0 ✗

a = a + b ✓

Eg ②

a = (b = (c = (d = 10)))

← Right to left.

Eg ③

a = a + 1 ⇒ a += 1;

a = a - 2 ⇒ a -= 2;

a = a % 10 ⇒ a %= 10;

a = a / 10 ⇒ a /= 10;

Shorthand operator.

a = b + (c * 10); ✗ can't applied

a = a + (c * 10);
a += (c * 10); ✓ allowed