```
int a=10, b=10;
int c;
```

① ✓ `c = a = b;`
10 10

② `c = a = 'b';`
98

③ ✗ `c = 'a' = 'b';`

④ ✗ `c = 'a' = b;`

b ⟶ variable

v/s

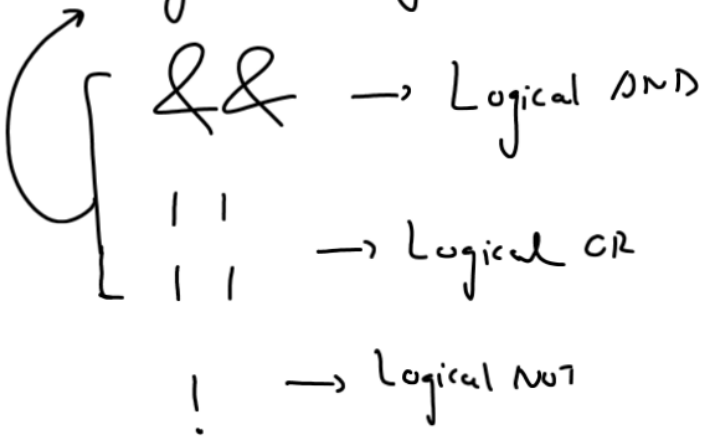
'b' ⟶ char const with ASCII code 98

v/s

b( ) ⟶ might be a fn

Logical Operators

(Used for checking multiple conditions)



X $0 < n < 10$

X $a > b < c$

Cond 1	&&	Cond 2	Result
T	&&	T	T
T	&&	F	F
F	&&	T	F
F	&&	F	F

Cond 1	||	Cond 2	Result
T	||	F	T
F	||	T	T
T	||	T	T
F	||	F	F

```
int a = 10, b = 5, c = 15, d;
```

```
d = a > b && b > c;
    1       0
0
```

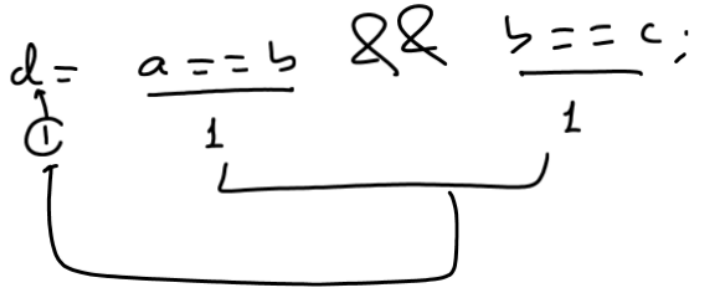
```
d = a > b || b > c;
    1       0
1
```

```
int a = 10, b = 10, c = 10, d;
```

Non Sense!

```
d = a == b == c;
    1    == 10
0
```

```
int a = 10, b = 10, c = 10, d;
```



Precedence Table — Precedence and Associativity

Grp 0: ( )

Grp 1: /, *, %

Grp 2: +, -

Grp 3: >, <, >=, <=

Grp 4: ==, !=

Grp 5: &&

Grp 6: ||

Grp 7: =

a + b * c
② ①

a / b * c
→ ① ②

a * b / c
→ ① ②

a = b = c = 10;
←

① a = b + c * d - e ;
④ ② ① ③

② a = b * c / d - e + f ;
⑤ ① ② ③ ④

⑧ a = b ./. c + d / e ;
④ ① ③ ②

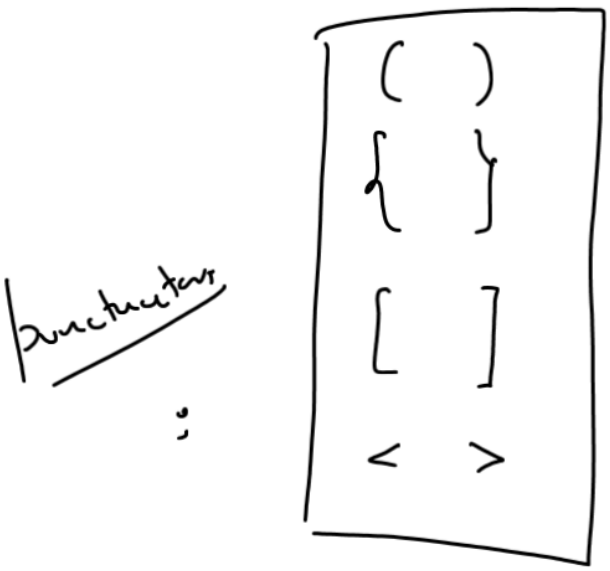
④ ✗ a + b = c - d ;

⑤ a = b == c * d + e ;
④ ② ① ③

⑥ a = b + c * d - e ;
④ ② ① ③

⑦ a = (b + c) * (d - e) ;
④ ① ③ ②

⑧ a = b * (c + d / e) ;
④ ③ ② ①



Developing basic program
==========
WAP to accept 3 integers from the user and calculate and print their sum and average.

SAMPLE OUTPUT 1
==========
Enter 3 integers: 3 4 5
Sum is 12
Avg is 4.000000

SAMPLE OUTPUT 2
==========
Enter 3 integers: 3 4 6
Sum is 13
Avg is 4.333333