

CS & IT ENGINEERING

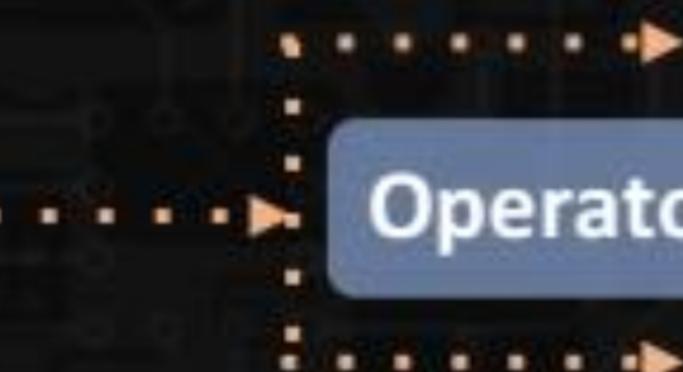


Programming in C
Chapter-1
Data types and Operators
Lec- 05



By- Pankaj Sharma sir

TOPICS TO BE COVERED



Operators-II

Unary +, -

Arith $\left[\begin{matrix} \times, /, \% \\ +, - \end{matrix} \right]$

L to R

Rel. $\left[\begin{matrix} <, \leq, >, \geq \\ ==, != \end{matrix} \right]$

L to R

Assignment =

R to L

High

1) Unary +, -

2) $\times, /, \%$

3) +, -

4) $<, \leq, >, \geq$

5) ==, !=

6) =

↓
low

High

↓
low

int a;

$\boxed{2 > 10}$ | = 3 == 8 > -1 > 5 ;

printf("%d", a); 0

① >

② |= == .

③ =

$2 > 10 \Rightarrow$ Is 2 greater than 10 ?

a = 0 | = 3 == $\boxed{8 > -1 > 5}$;

a = 0 | = 3 : = $\boxed{1 > 5}$;

a = $\boxed{0 | = 3} = = 0$

a = $\boxed{1 == 0}$

a = 0

```
int i;  
i = 3; | valid
```

```
int i;  
i = printf("Pankaj");  
6  
printf("./d", i);
```

i = $3 \times 2 + 1$
Evaluate

Lvalue = Rvalue

Pankaj6

~~printf("/d", 3x2 + 5 - 6/2); printf("/d", printf("Hello Bachho"))~~

$$\underline{3 \times 2 + 5 - 6/2}$$

$$6 + 5 - 6/2$$

$$6 + 5 - 3$$

$$11 - 3$$

$$8$$

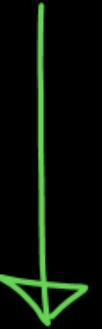


Hello Bachho12

जैसे क्षेत्रीय operators की एक value होती

%, C में printf की एक एक value होती

%



No. of symbols or counting of symbols
printed by printf.

Logical Operators

- 1) Logical AND (&&)
 - 2) Logical OR (||)
 - 3) Logical NOT (!)
- binary
- unary

Logical AND (And)

AND : और

(No choice)

a and b

Just like relational

Operators, the result/
output/value of a

logical operator is
Either 0 or 1

$a \times b = \text{non-zero}$

{When both are
non-zero}

a and b

a

F

F

T

T

b

F

T

F

T

$a \wedge b$

F

F

F

T

0

0

0

1

Result/value/output

12 → True

13.78 → True

&& * If both operands are non-zero
then the output is 1.
Otherwise, the output is 0

T T

12 && 13.78

1

```
printf ("%d", 12 && 13.78); 1  
printf ("%d", 1 && 1.07); 1  
printf ("%d", 0 && 3.2); 0  
printf ("%d", 0.0 && 0.0); 0
```

1.

int a ;

a = printf("Pankaj") && 6 ; ① Pankaj

printf("./d",a); ②

non-zero & non-zero
6 && 6 \Rightarrow True && True
True 1

2.

int a ;

③ a = 1
④ 1

Pankaj1

Pankaj0

a = printf("Pankaj") && (6/2 - 3) ;
printf("./d",a);

① Pankaj
② 6 && (6/2 - 3)
③ 0

True False 0
6 && 0 \Rightarrow False

1) Unary + -

2) X, /, .

3) +, -

4) <, <=, >, >=

5) ==, !=

6) &&

7) =

Logical OR (||)

OR : choice
अन्यथा

a	b	$a \parallel b$	O/P
F	F	F	0
F	T	T	1
T	F	T	1
T	T	T	1

The output is 1 (true), if at least one operand is non-zero (true), otherwise, the o/p is 0.

O/P

printf("%d", 2 || -12); 1

printf("%d", -378 || 30); 1

printf("%d", 0 || 7); 1

printf("%d", 7 || 0.0); 1

printf("%d", 0 || 0); 0

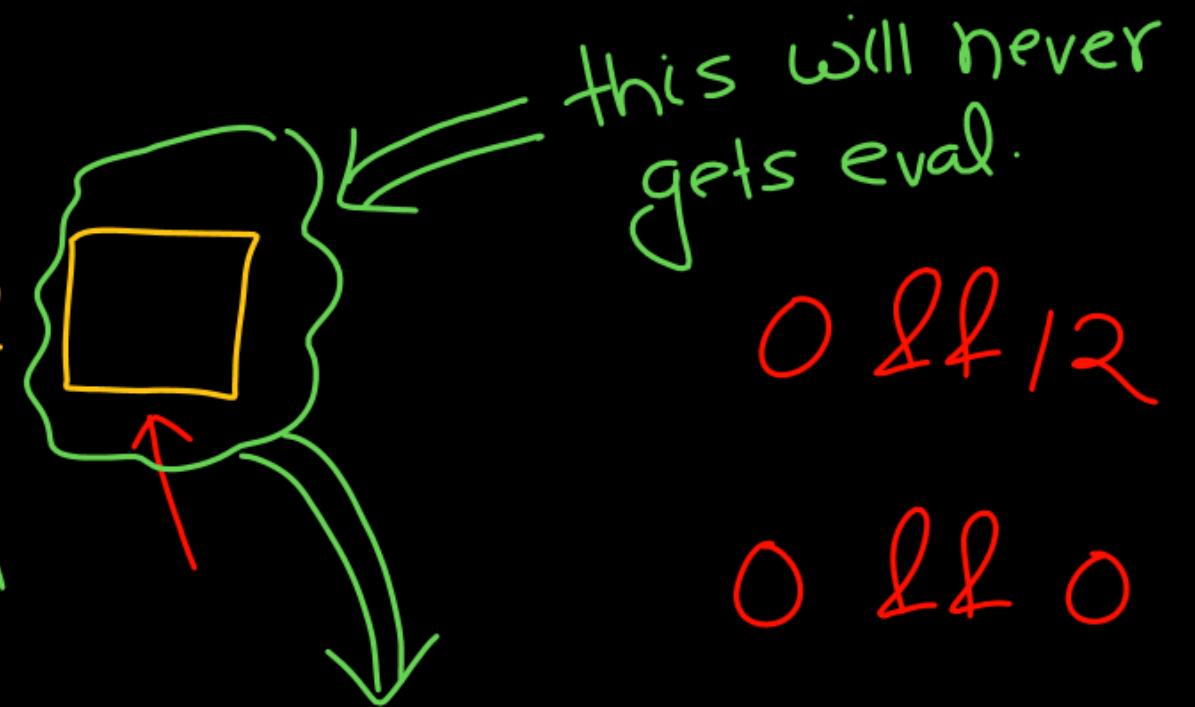
12 & -1
true & true
true

① ✓

If the 1st
Operand of logical
& operator is
0 & 1
0 & 0

0 other
is not
evaluated

Short-Circuit Evaluation



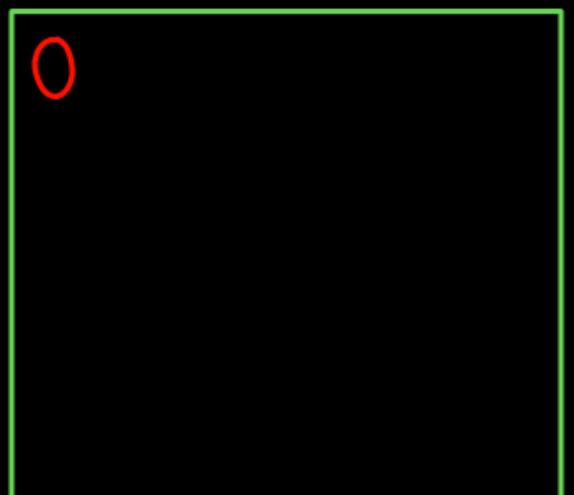
$$0 \& 1 = 0$$

$$0 \& 0 = 0$$

The result is 0, irrespective
of whatever be 2nd operand.

```
int a;  
a = 0 && printf("Hello");  
printf("%d", a);
```

Never eval.



1st operand is 0

2nd operand is {value of}
{printf}

Q

int a ;

a = ~~printf("Pankaj") - printf("Neeraj")~~ 6
~~printf("./d",a);~~ 6

ff printf("Gate");

6-6 ff printf("Gate");
a = 0 ff printf("Gate").
 X

Pankaj Neeraj 0

Q

int a ;

a = printf("Pankaj") = printf("Neeraj") && printf("Gate");
printf("./d", a);

① printf("Neeraj") && printf("Gate")
a = printf("Pankaj") = 1

Lvalue = Rvalue

variable
↳

literal
var
Expression

Q

1. If so -

Compiles

int a; 6 == 6

a = printf("Pankaj") == printf("Neeraj")
printf("./d", a);

if printf("Gate");

PankajNeerajGate1

= =

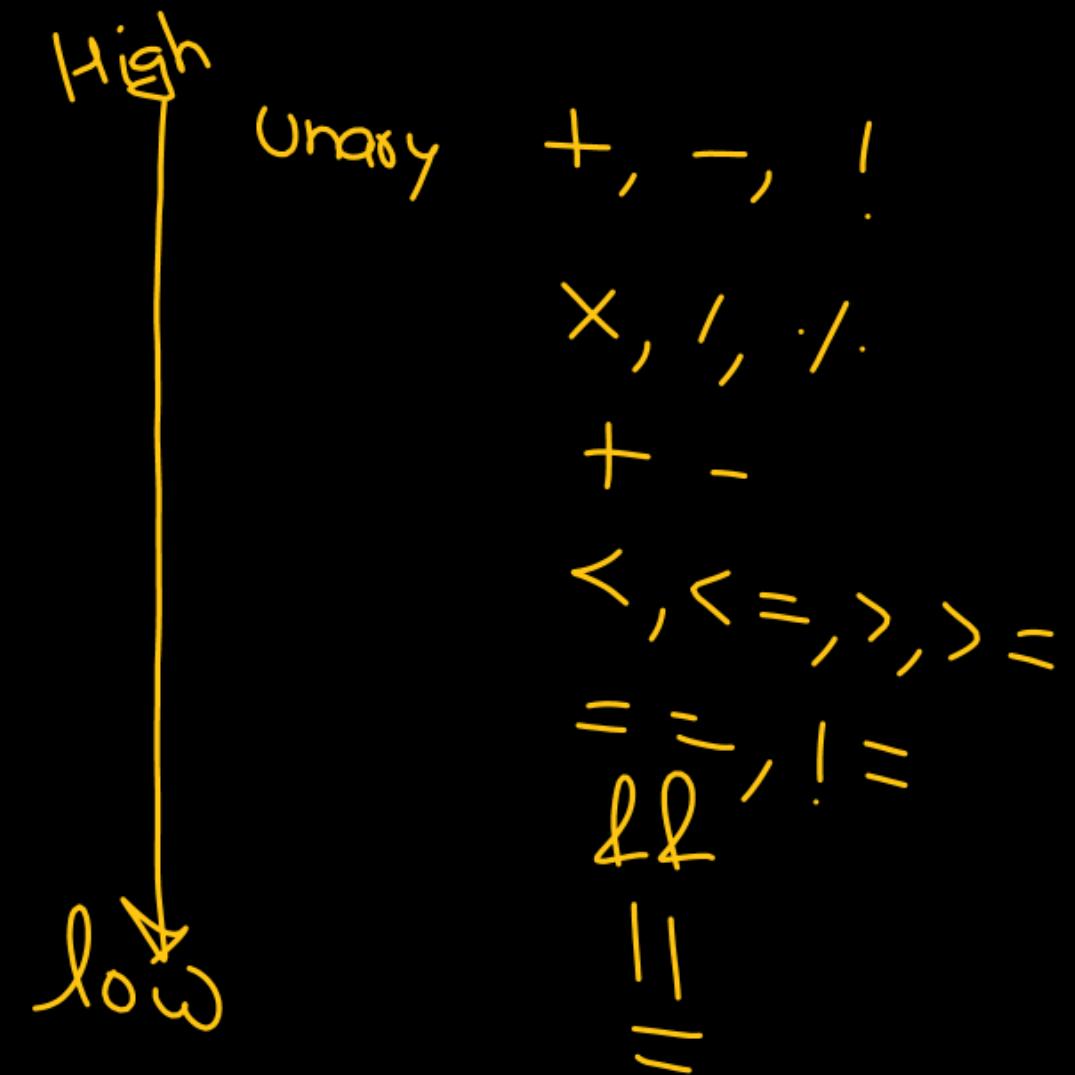
if

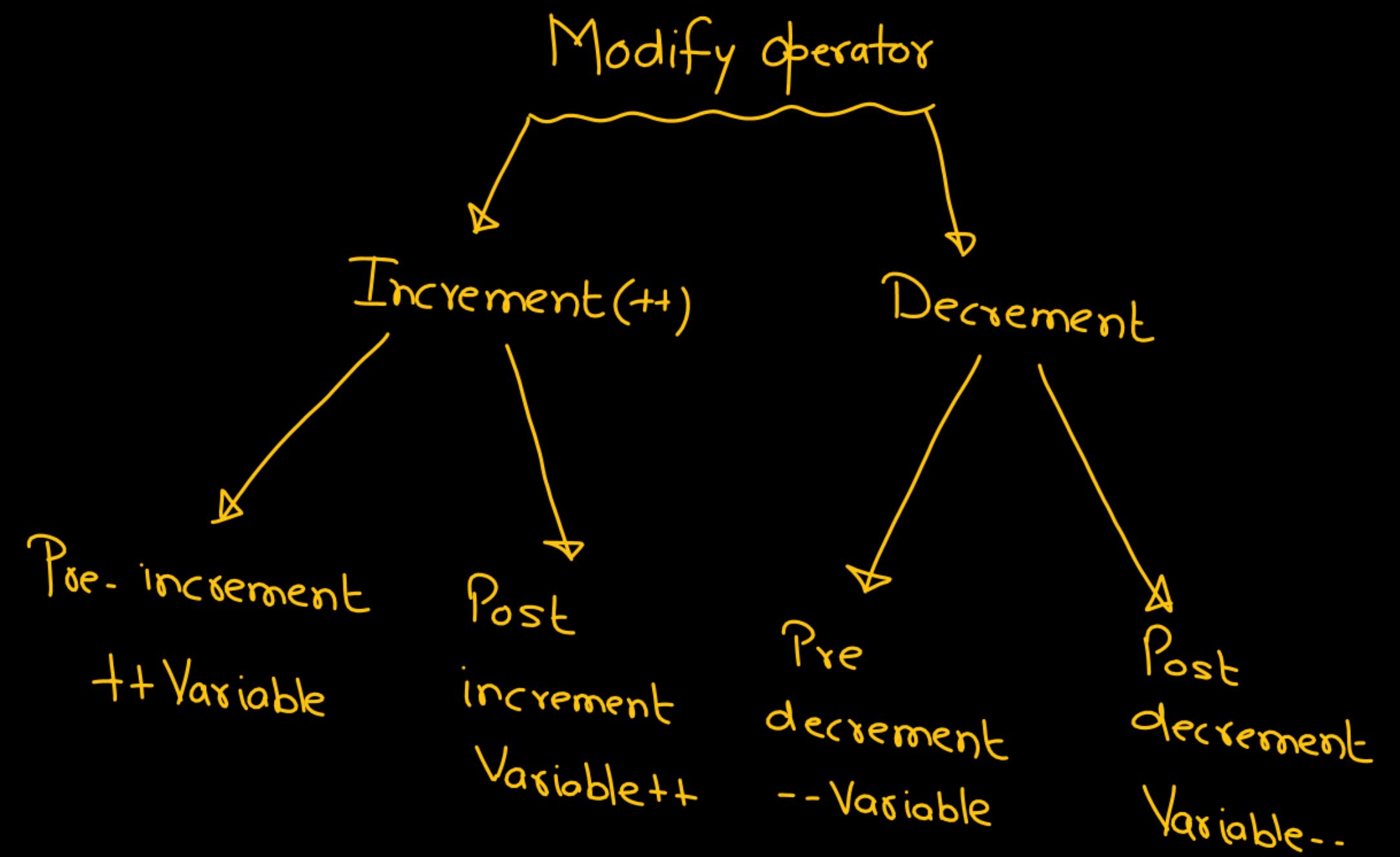
Logical NOT(!) \Rightarrow

$\text{NOT}(2) = \text{NOT}(\text{Non-Zero}) = \text{NOT}(\text{True}) = \text{False}$

$!(\text{non-Zero}) = 0$

$!(0) = !(\text{False}) = \text{True} = 1$



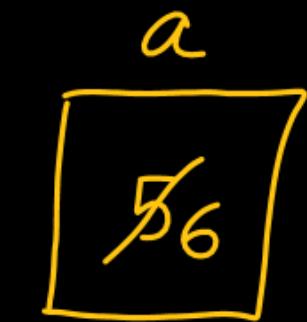


```
int a = 5;
```

```
++a;
```

```
printf("%d",a);
```

O/P: 6

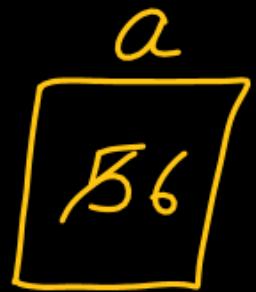


```
int a=5;
```

```
a++;
```

```
printf("%d",a)
```

O/P: 6



doubt?

A hand-drawn diagram of an assignment statement. It consists of a rectangular box containing the expression 'a = a + 1'.

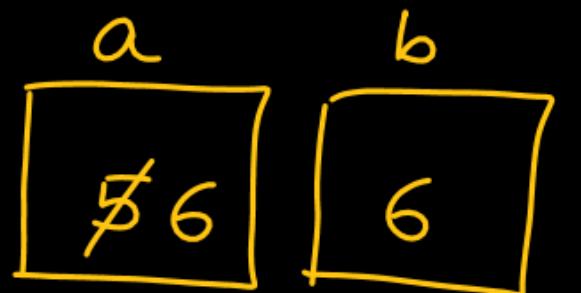
$++a \rightarrow$ (i) First, increase the value of variable by 1

(ii) Use the updated value

int a = 5, b;

b = ++a; $\xrightarrow{(i)} a = a + 1$;
 $\xrightarrow{(ii)} b = a$;

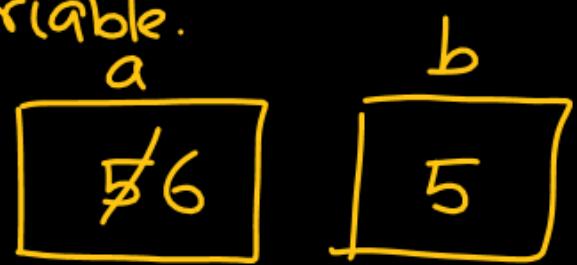
printf("%d %d", a, b);



66

$a++ \rightarrow$ (i) Use the value of variable

(ii) Then, increase the value of variable.



int a = 5, b;
b = a++; $\xrightarrow{(i)} b = a$;
 $\xrightarrow{(ii)} a = a + 1$
printf("%d %d", a, b);

65

int a = 5, b();

b = ++a + a++ + a++;()

C Standard

(i) Sequence point

Between 2 succ.

Sequence points, we
can modify the
value of a variable
atmost once.

Compiler Dependent

```
Void main() {  
    12;  
    13.78;  
}
```

Statement

exp \Rightarrow statement with some value

1) Decimal Number System (10 symbols
0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

2) Binary Number System (2 symbols
0, 1)

3) Octal Number System (8 symbols
0, 1, 2, 3, 4, 5, 6, 7)

4) Hexadecimal Number System (16 symbols
(0-9, A, B, C, D, E, F)
(0-9, a, b, c, d, e, f))

Decimal Number System

 = x

 [y]

New value = $\underbrace{10x}_\text{decimal} + \underline{y}$

value

place 10^1 10^0

old-value = 32

$3 \quad 2$

$3 \times 10^1 + 2 \times 10^0 = 30 + 2 = 32$

32 [6] \downarrow New symbol

10^2 10^1 10^0

New value = $3 \times 10^2 + 2 \times 10^1 + 6$

= $(3 \times 10^1 + 2 \times 10^0) \times 10 + 6$

= Old-value $\times 10 + 6$

Binary (0,1)

decimal to binary

decimal \rightarrow binary

(i) $(24)_{10} = (?)_2$

11000

$$\begin{array}{r} 2\sqrt{1} \\ \hline 0 \\ \hline 1 \end{array}$$

2	24	Rem
2	12	0
2	6	0
2	3	0
2	1	1
	0	1

Stop

$$\begin{array}{r} 2\sqrt{24} \\ \hline 12 \\ \hline 0 \\ \\ 2\sqrt{3} \\ \hline 1 \\ \hline 1 \end{array}$$

rem

$$(67)_{10} = (?)_2$$

$$= 1000011$$

2	67	Rem
2	33	1
2	16	1
2	8	0
2	4	0
2	2	0
2	1	0
	0	1

stop

binary to decimal

$$(11000)_2 = (?)_{10}$$

$$\begin{array}{r} 1 \ 1 \ 0 \ 0 \ 0 \\ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 \end{array}$$

$$\begin{aligned} &= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\ &= 16 + 8 \\ &= 24 \end{aligned}$$

$$(1000011)_2 = (\quad)_{10}$$

$$\begin{array}{r} \overbrace{1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1} \\ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 \end{array}$$

$$\begin{aligned} & \Rightarrow 1 \times 2^6 + 1 \times 2^1 + 1 \times 2^0 \\ &= 64 + 2 + 1 \\ &= 67 \end{aligned}$$

$$\begin{aligned} 2^0 &= 1 \\ 2^1 &= 2 \\ 2^2 &= 4 \\ 2^3 &= 8 \\ 2^4 &= 16 \\ 2^5 &= 32 \\ 2^6 &= 64 \\ 2^7 &= 128 \\ 2^8 &= 256 \\ 2^9 &= 512 \\ 2^{10} &= 1024 \end{aligned}$$

binary value

10

2

10 0

new value \Rightarrow Old-val $\times 2 + 0$
 $= 2 \times 2 + 0$
 $= 4$

1 0 0
2² 2¹ 2⁰

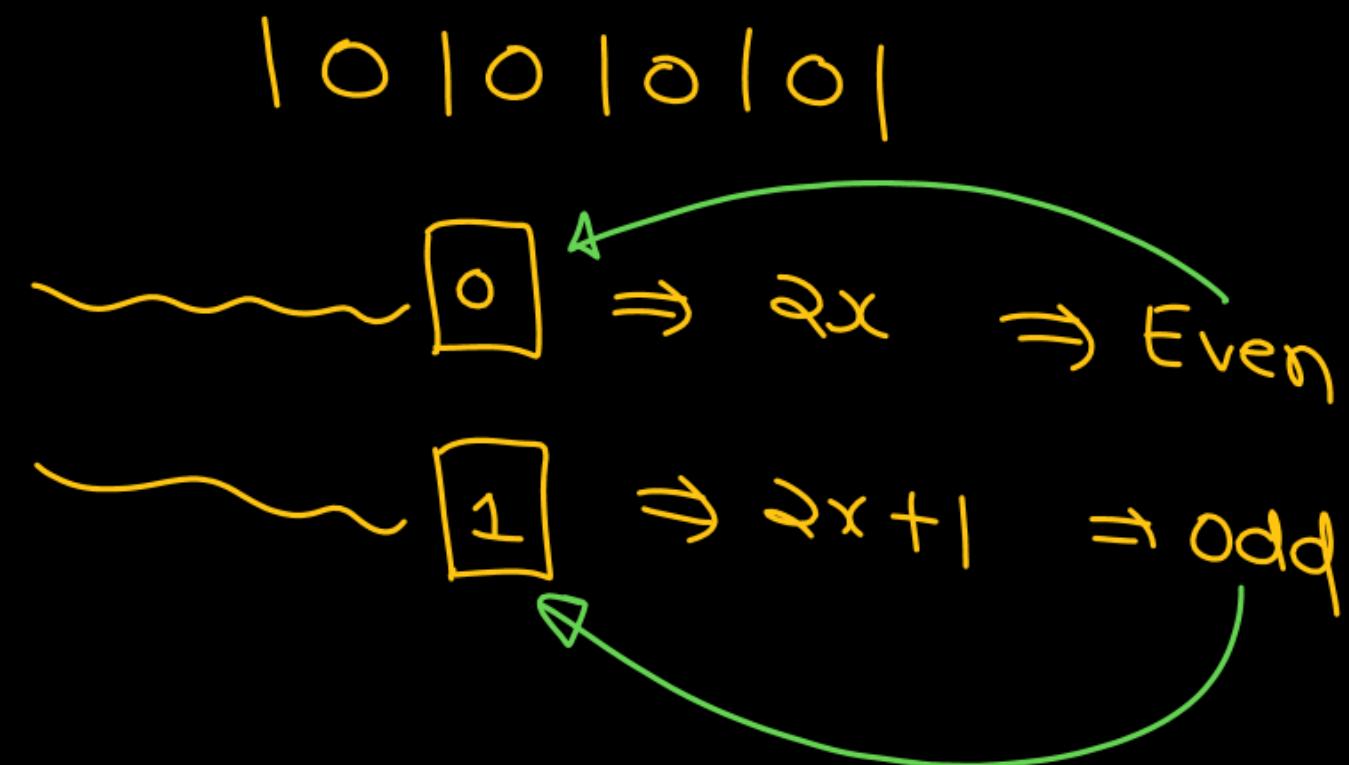
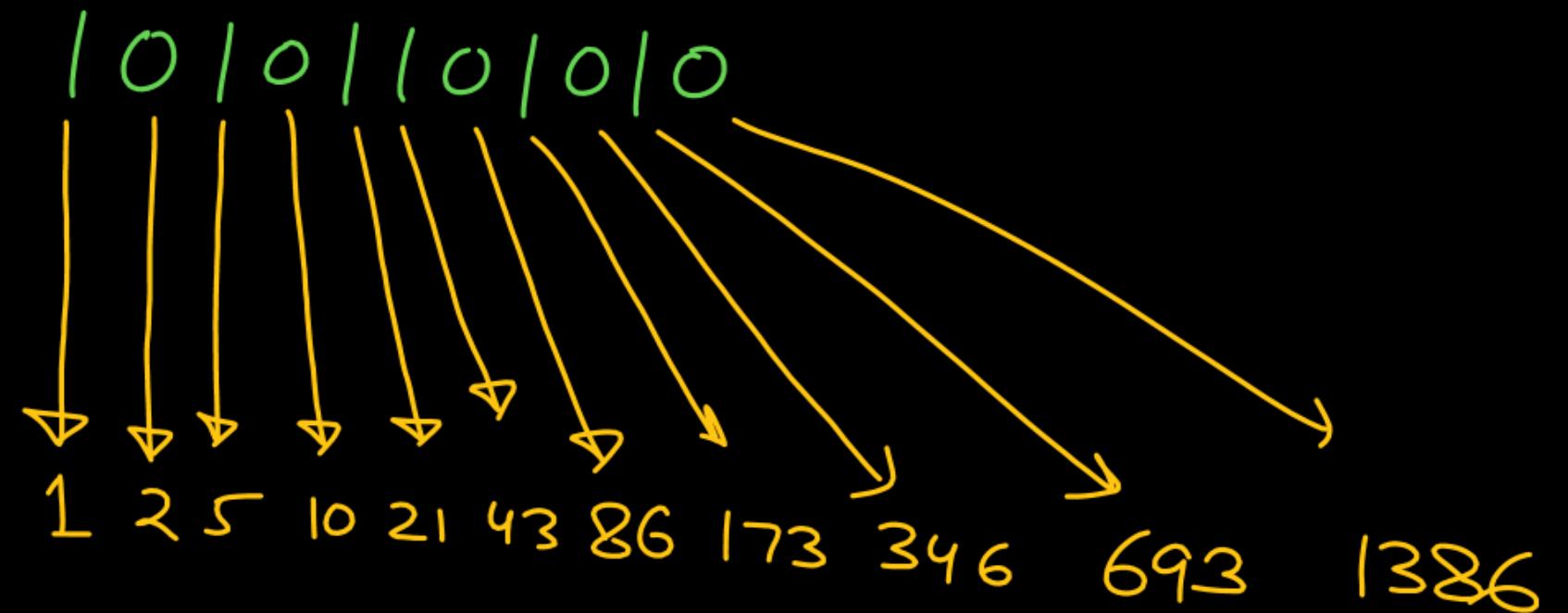
$$\Rightarrow 1 \times 2^2 = 4$$

binary NO $\Rightarrow x$

$\sim\sim\sim$ 0 $\Rightarrow 2x + 0 = 2x$

$\sim\sim\sim$ 1 $\Rightarrow 2x + 1$

value



Octal (0-7)

decimal \rightarrow octal

$$(35)_{10} = (?)_8$$

$$= (43)_8$$

8	35	Rem
8	4	3
0		4
Stop	↑	↑

$$(123)_{10} = (\quad)_8$$

$$= (173)_8$$

8	123	Rem
8	15	3
8	1	7
	0	1

stop ← ↑

$$(173)_8 = (\quad)_{10}$$

$$\begin{array}{r} 173 \\ 8^2 8^1 8^0 \end{array}$$

$$= 1 \times 8^2 + 7 \times 8^1 + 3 \times 8^0$$

$$= 64 + 56 + 3$$

$$= 123$$

$$(43)_8 = (\quad)_{10}$$

$$\begin{array}{r} 43 \\ 8^1 \ 8^0 \\ \hline = 4 \times 8^1 + 3 \times 8^0 \\ = 32 + 3 \\ = 35 \end{array}$$

YT → 1 min

Next class

- Hexadecimal
- Escape seq.
- bitwise operators
- Problems

8:30 PM

9/1

