CS & IT ENGINEERING

C Programming

Data Types and Operators

Lec- 04

By- Pankaj Sharma sir

TOPICS TO BE COVERED

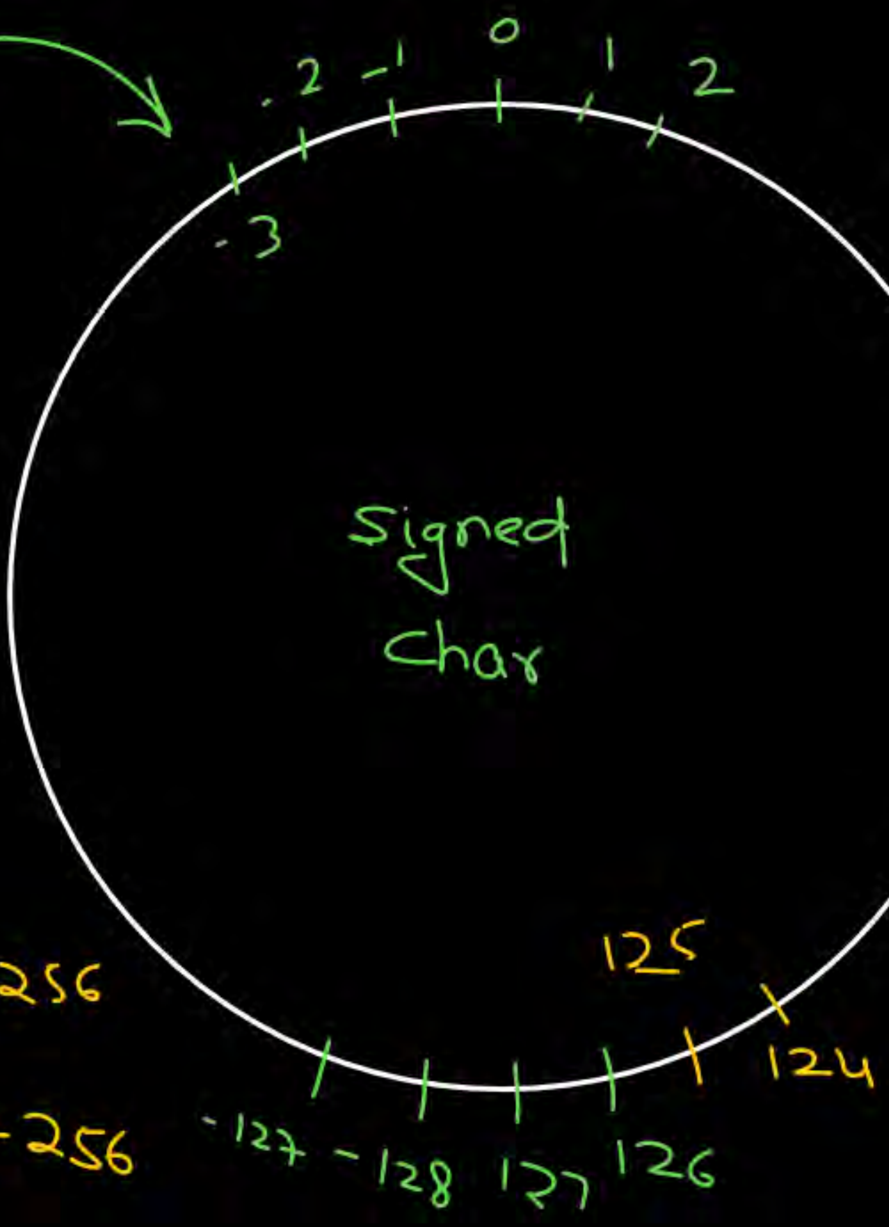
Data Types -1

1.

by default signed

```
char c = -191;
printf("%d", c);
```

65



c	store	
-130	126	-130+256
-132	124	-132+256
-191	65	-191+256

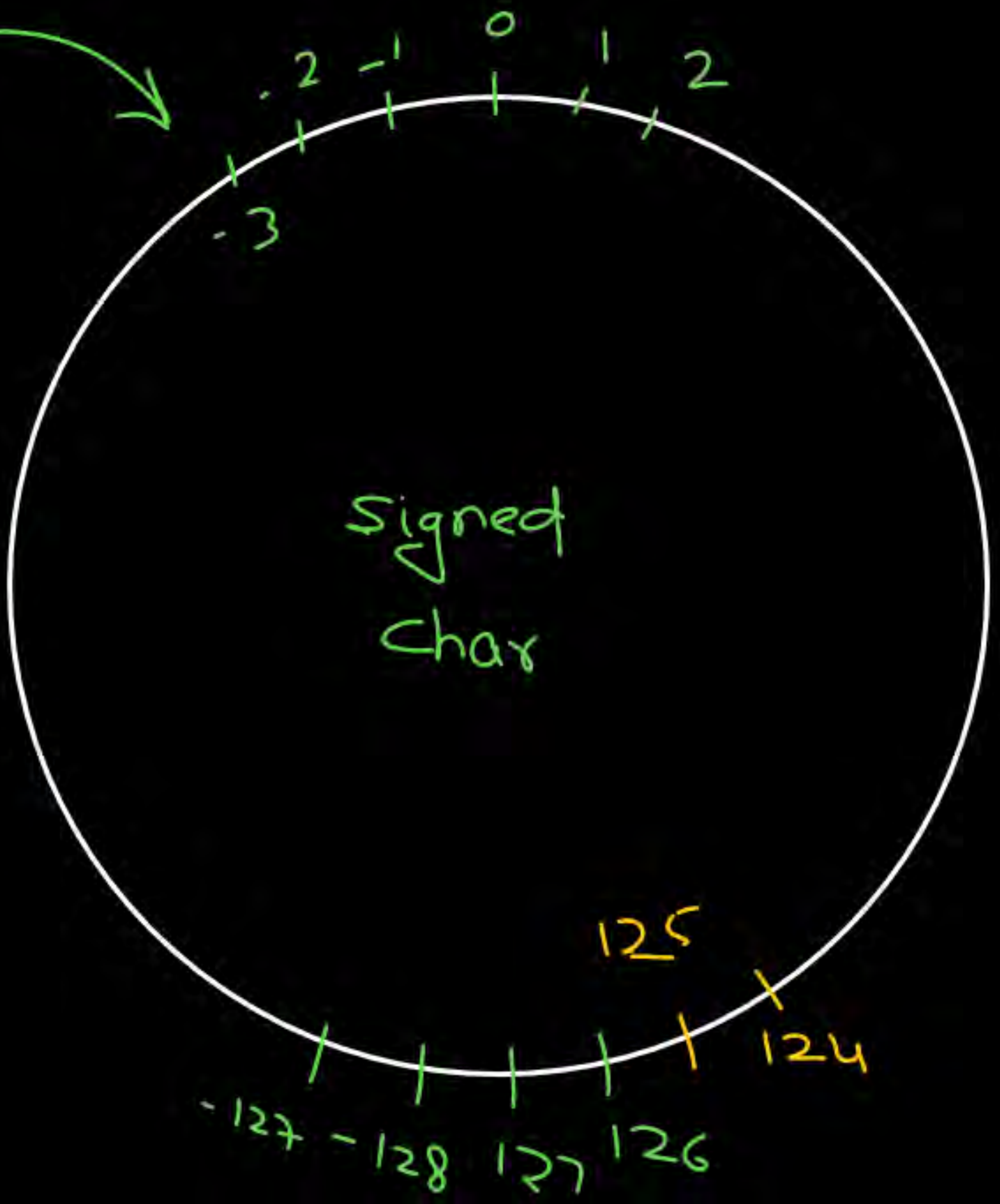
1.

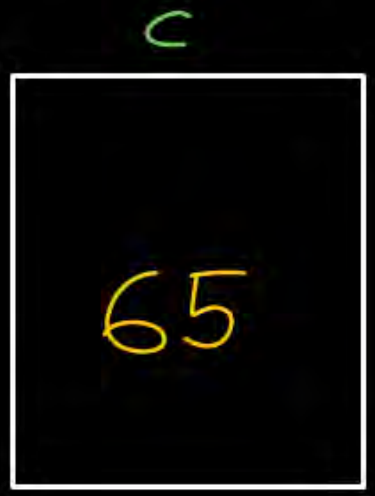
by default signed

```
char c = -191;
printf("%c", c);
```

char system

65 → A





Char system

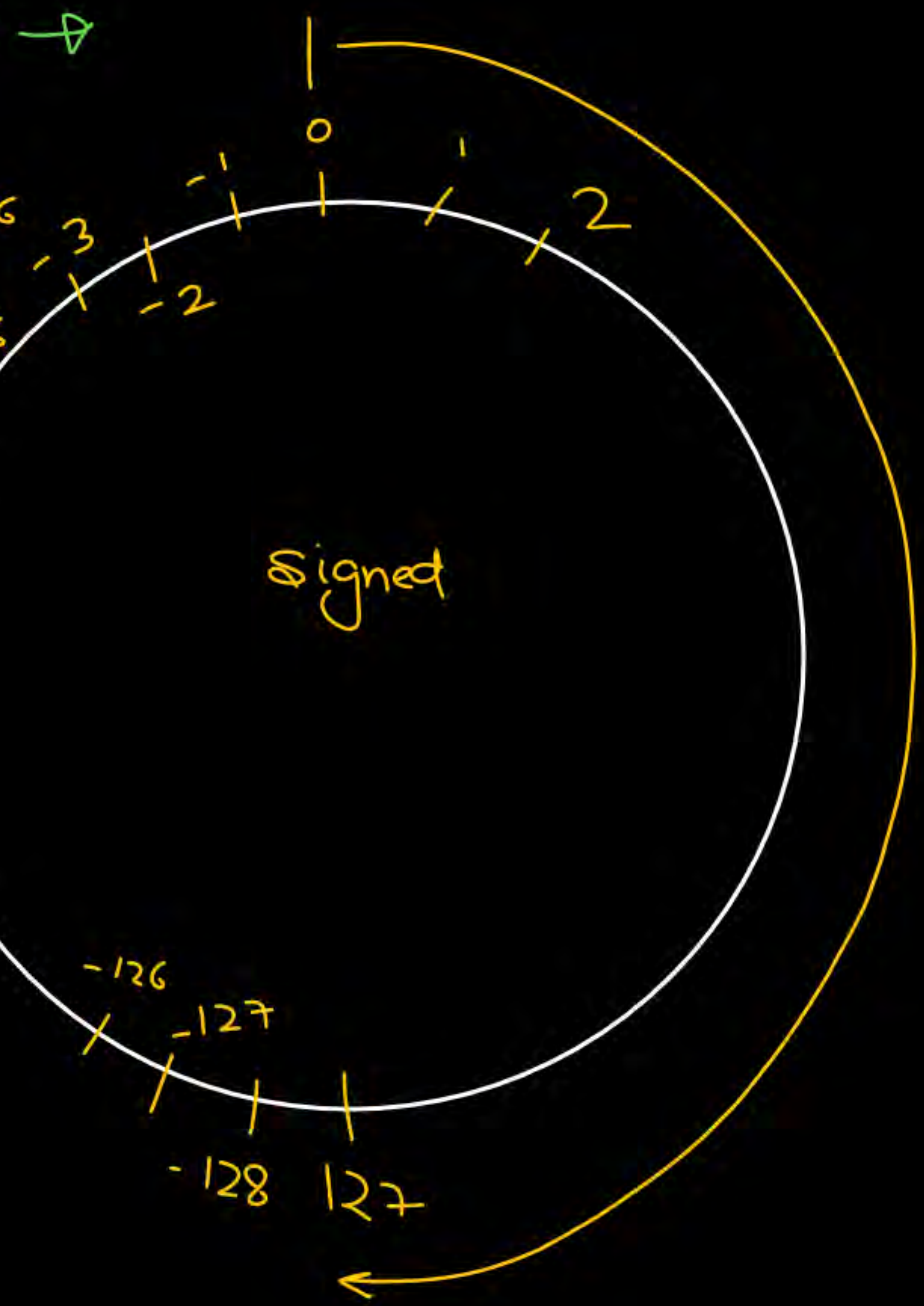
Symbol → +ve integer constant

unsigned

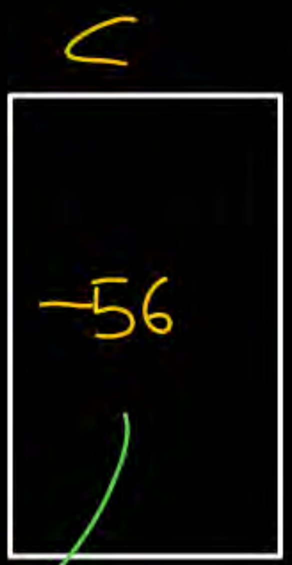
① store

c	store	
130	-126	130-256
200	-56	200-256

200 →



```
char c = 200 ;
printf("%c", c);
```



char system

-ve values ✗

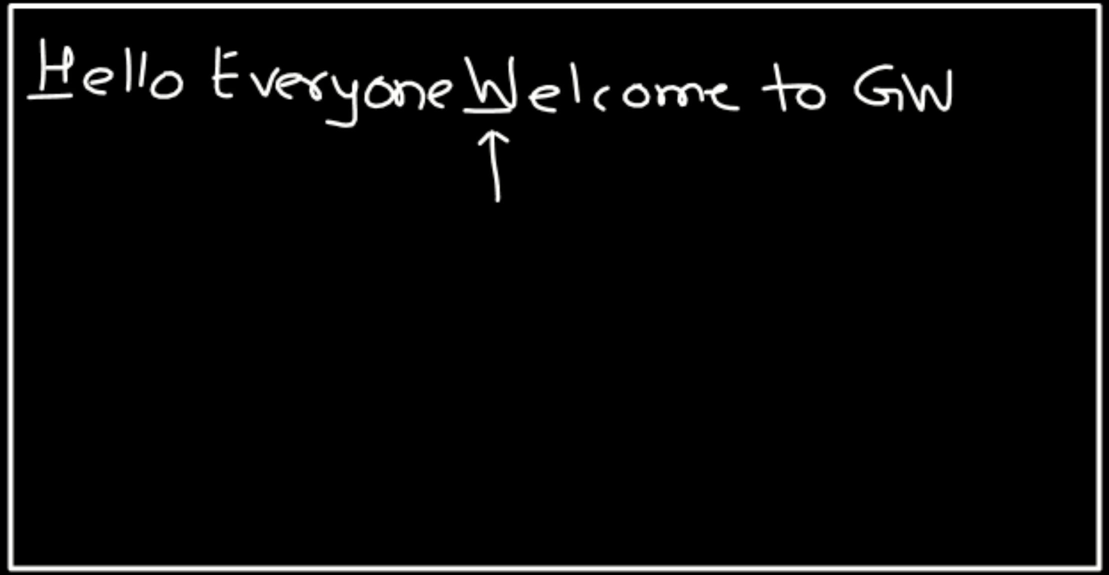
+ve values

%c   Char system   symbol   unsigned

1) \n

2) \t

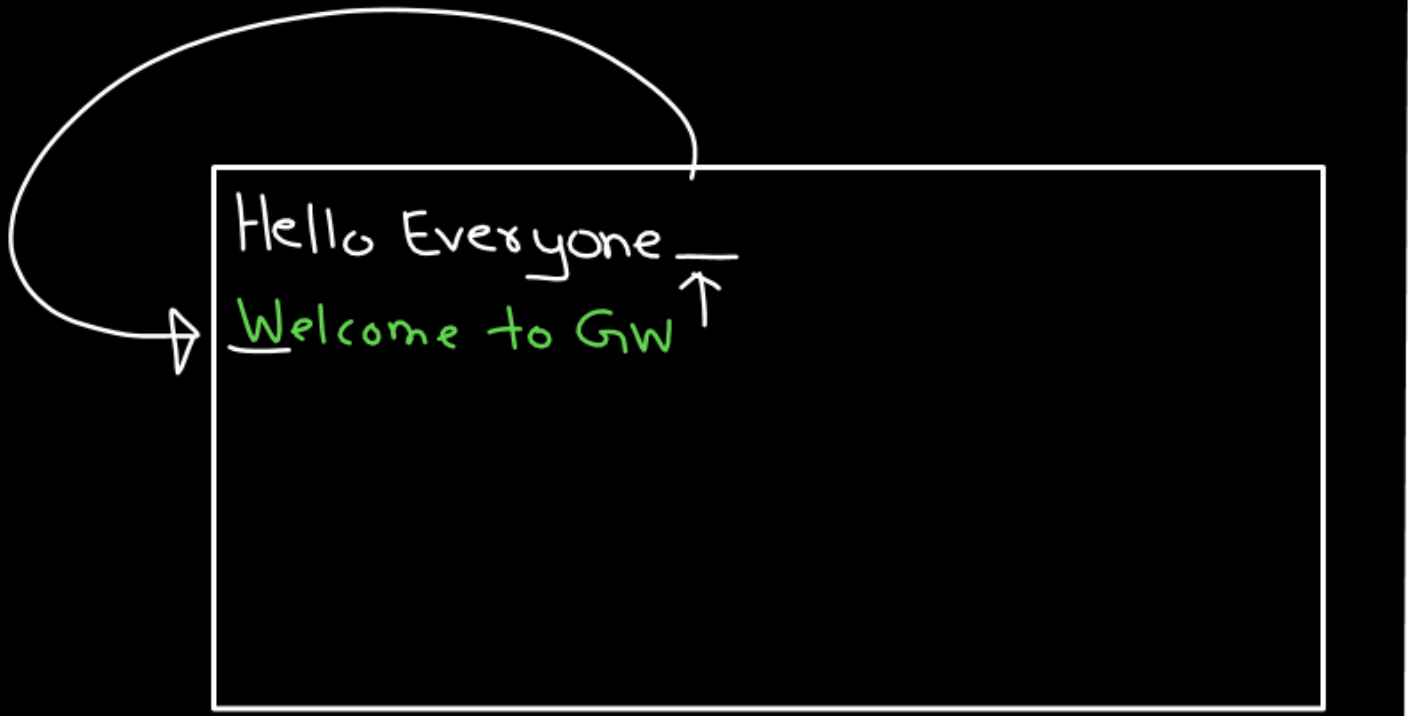
```
# ________
void main(){
    printf("Hello Everyone");
    printf("Welcome to GW");
}
```



```
Hello EveryoneWelcome to GW
```

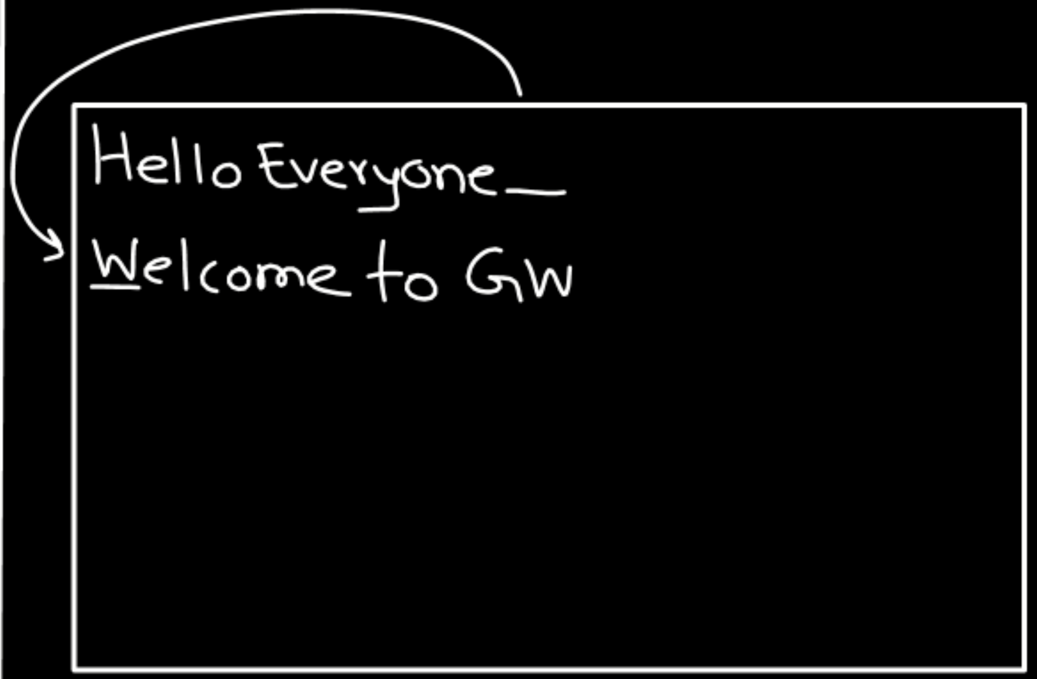
① \n → Move the cursor to the begining of next line

```
printf("Hello Everyone\n");
printf("welcome to GW");
```



```
Hello Everyone_
Welcome to GW
```

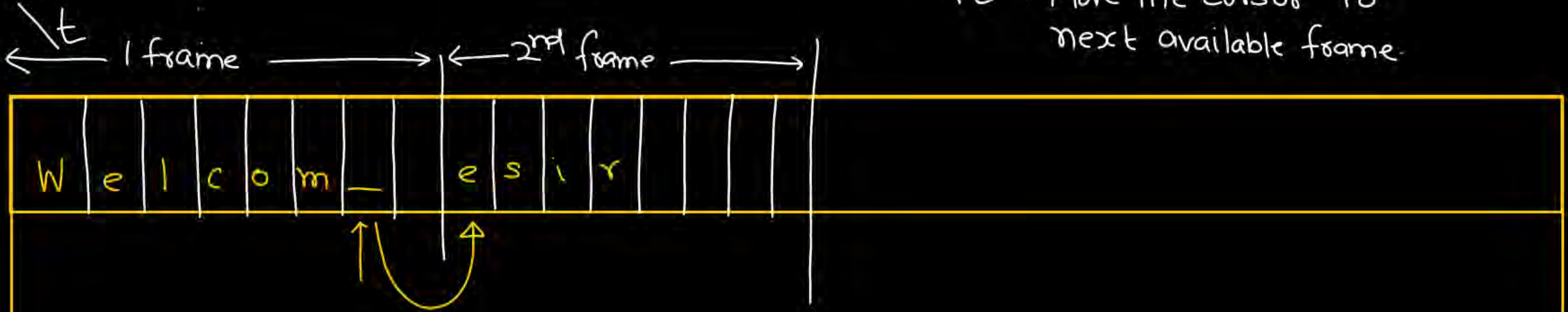
```
printf("Hello Everyone");
printf("\nWelcome to GW");
```



```
Hello Everyone_
Welcome to GW
```

②

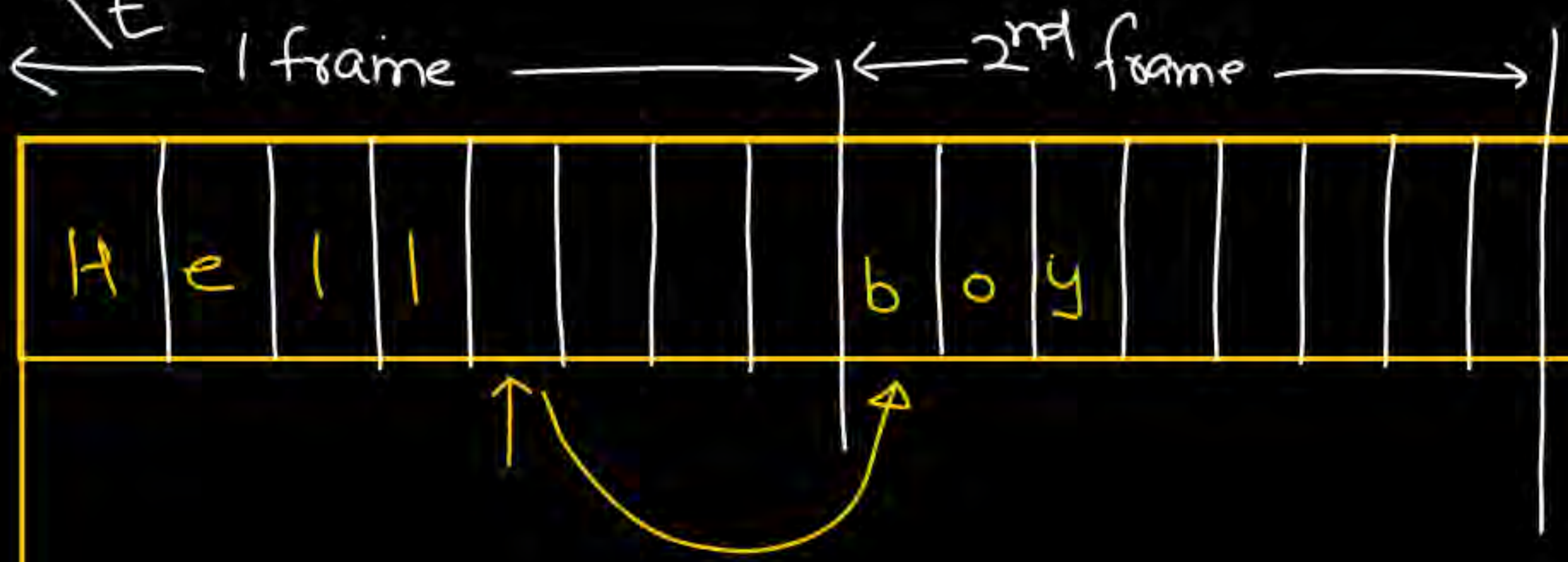
\t : Move the cursor to next available frame.



```
printf("Welcom\te");
printf("sir");
```

② \t

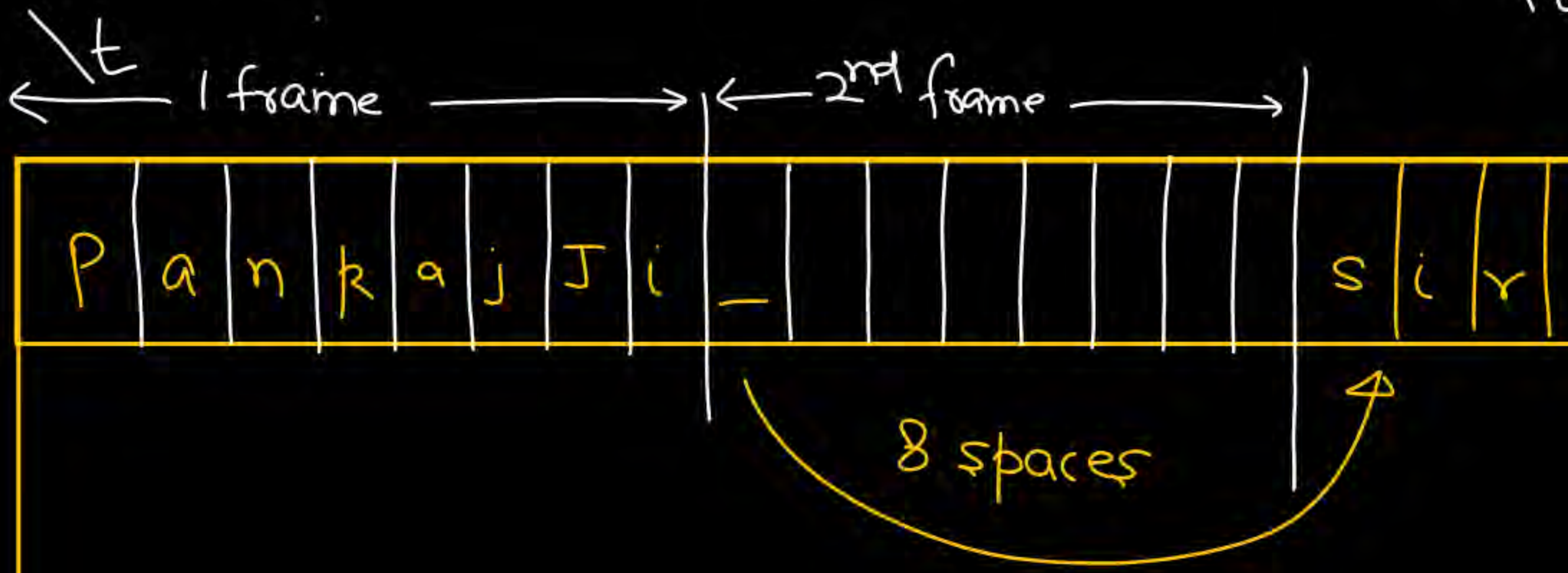
\t : Move the cursor to next available frame.



```
printf("Hell\t boy");
```

②

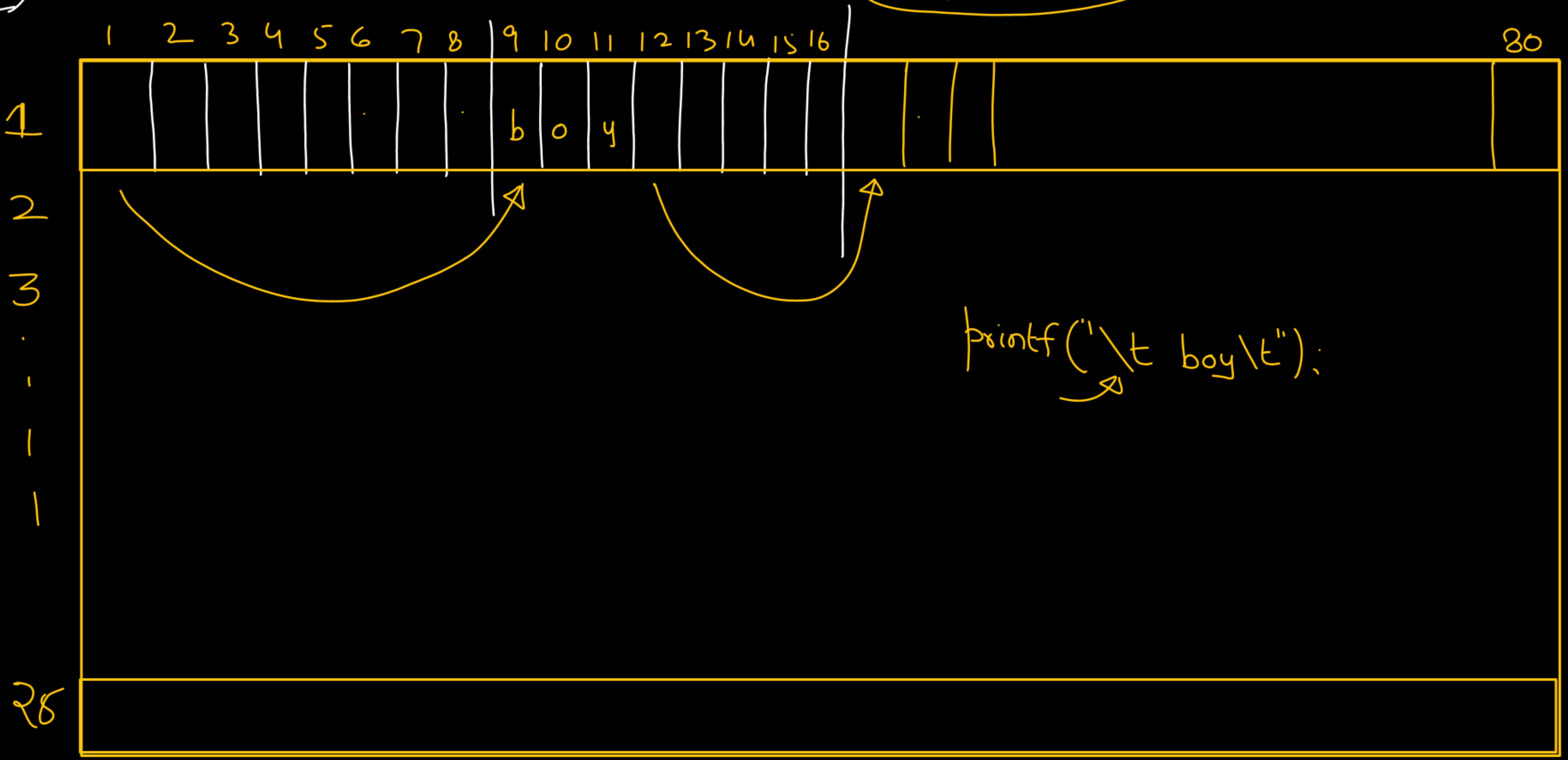
\t : Move the cursor to next available frame.



```
printf("Pankaj Ji\t");
printf("sir");
```

2

\r \v X



```
printf("\t boy\t");
```

Operators

$\underset{\text{operand}}{\boxed{10}} \overset{\text{operator}}{+} \underset{\text{operand}}{\boxed{20}} \Rightarrow$ 2 values (operands)

$10 \times 20 \Rightarrow$ 2 values (operands)

$-7 \Rightarrow$ - is working on 1 value (operand)

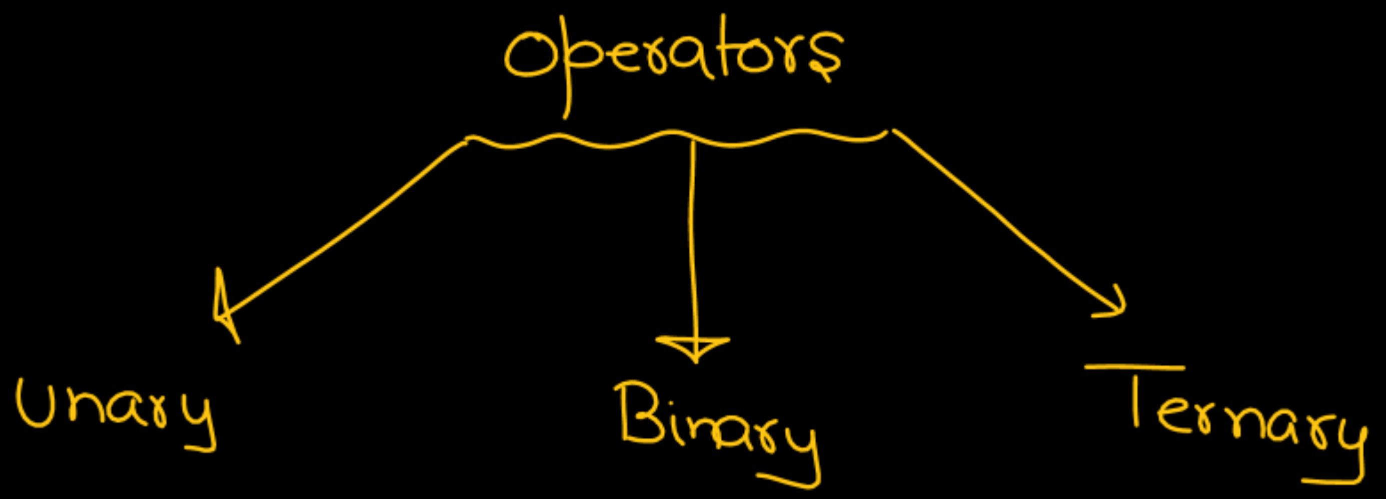
value/result

$10 + 20 \Rightarrow$

$10 \times 30 \Rightarrow$

$30/10 \Rightarrow$

हर operation की एक value होती है |



Operators

Unary

Binary

Ternary

1 operand

2 operand

3 operand

$-7, +8$

$10 + 20$

10×20

$30 / 10$

C Program → statements

int a ; declaration

a = 10 ;

Expression :

statement with some value

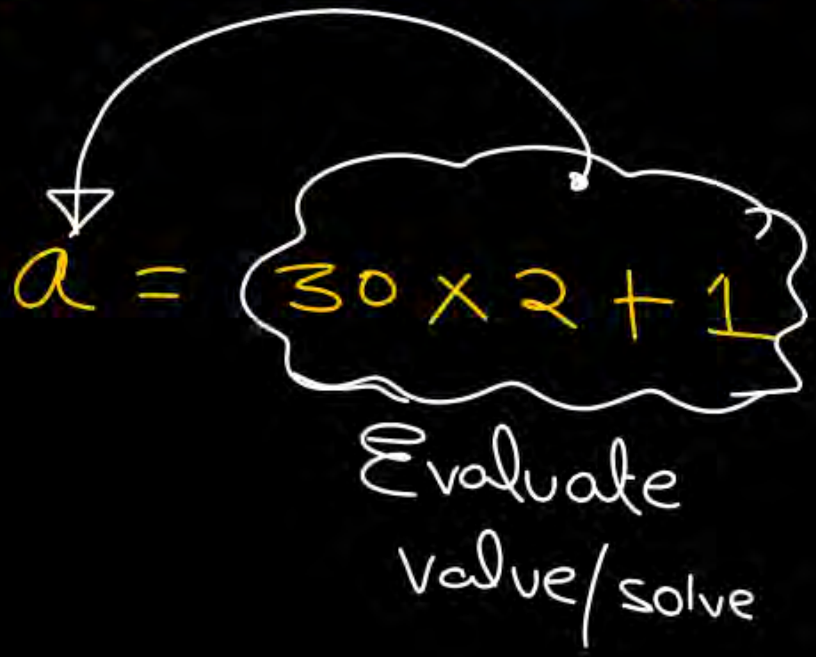
```
#include<stdio.h>
void main(){

        10 ;
        10+20 ; ✓
        30/2 ; ✓
        }
```

10 + 20 → Expression

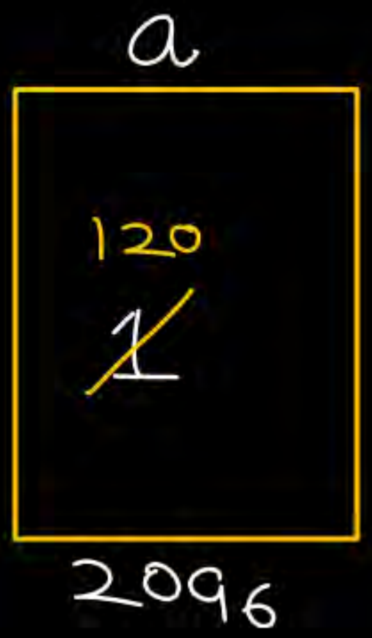
Assignment (=) Operator

1) Binary operator (2 operands)



update the value of a variable

```
int a = 1 ;
a = 120 ;
```



(i) LHS = RHS

int a=12;

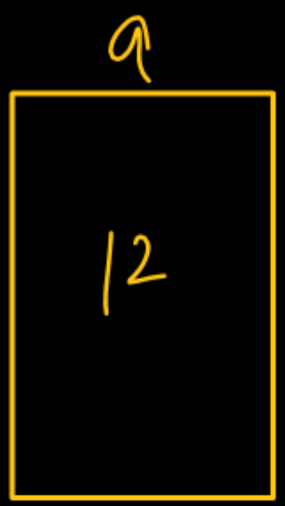
10 = a;

RHS → Expression

→ constant/literal

→ variable

LHS → constant/literal ✗



(ii) a = 10+3×1; ✓ valid

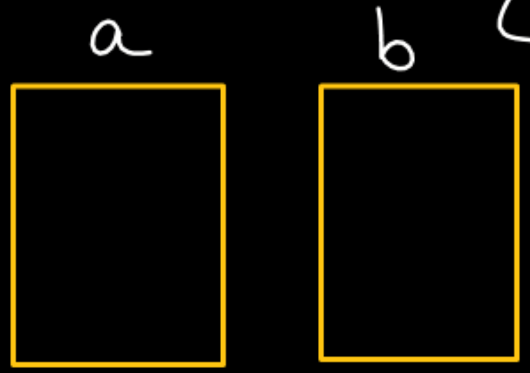
(iii) a = 100; ✓ valid

(iv) int a=10,b;

b=a; ✓ valid

(v) int a,b;

a + b = 100; ✗



Compiler ud ke laat marega

Lvalue = Rvalue → exp/constant/variable

↓

Can not be exp.
Can not be constant
Must be some variable

```
int  a ;
a =    ;
```

```
int  a ;
   = 10 ;
```

```
{
int  a ;
a = 10
}
```

before }

Arith. Operator

unary

+ , −

+7 , -21

Binary

$\underbrace{+, -, \times, /}_{\text{math}}$, % modulas operator

BODMAS

$$8/2 \times 4$$

16

1

Priority (precedence)

(i) %, ×, / high

(ii) +, - low

$$3+8/2$$

↓

$$3+(8/2)$$

$$\underbrace{3+4}$$

$$\boxed{7}$$

$$8/2 \times 4$$

$$\longrightarrow$$

both operator are of same priority.

Associativity

L to R

$$(8/2) \times 4$$

$$4 \times 4$$

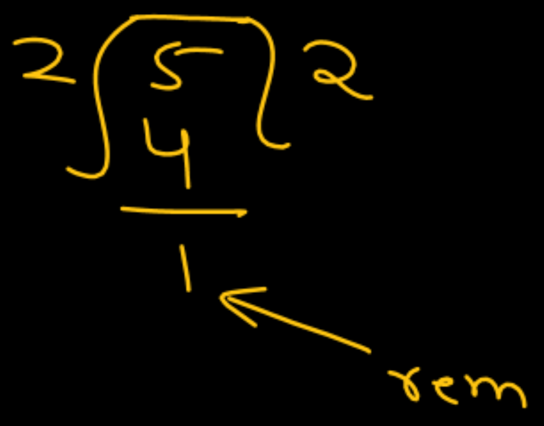
$$\boxed{16}$$

%

(i) binary

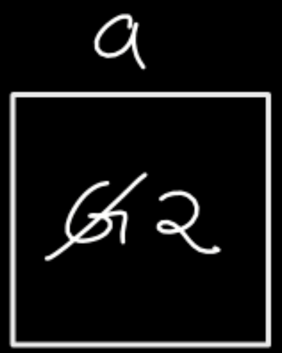
(ii) a%b : what is the remainder when a is divided by b

5 % 2



printf("%d", $\underset{①}{\overset{1}{\cancel{5\%2}}}$); 1

```
int a ;
a = 7 % 5 ;
printf("%d",a);
```



$a = 2;$

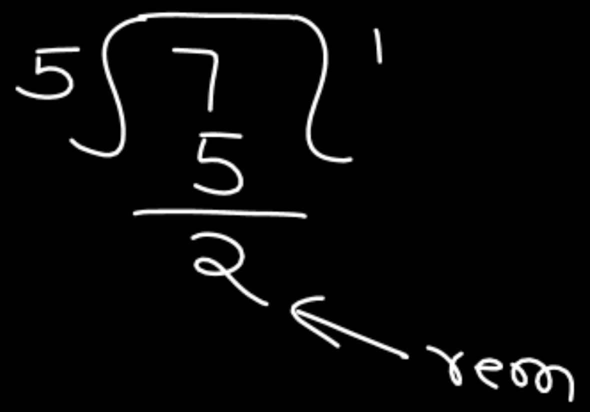
high ↓ low

%, /, ×

+, -

=

O/P : 2



** a % b

both operands must be integer type.

Otherwise → Error

Ud ke laat

a % b : The sign of result is as same as the sign of first operand.

-12 % 5 : -ve

12 % -5 : +ve

-12 % -5 : -ve

```
int a;             Eval
a = 13 % 4 x 3 + 5;
printf("%d",a);  8
```

$a = 13 \overset{①}{\%} 4 \times 3 + 5;$

→

$a = \underbrace{1 \times 3} + 5$

$a = \underbrace{3 + 5}$

$a = 8$

%, x

+

=

(i) printf("%d", $\underbrace{10 \times 2 + 3}$); 23

20 + 3

23 ⇒ value

value
30

(ii) printf ("%d", ~~10+20~~); 30

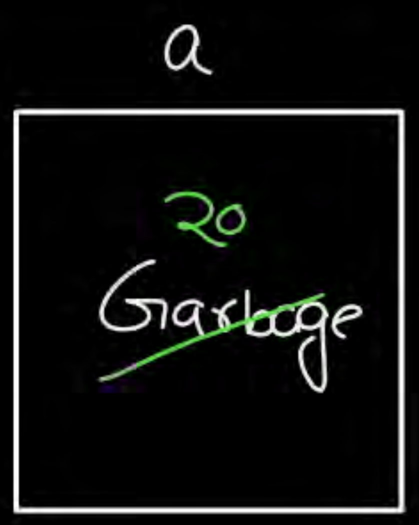
printf ("%d", 10×20); 200

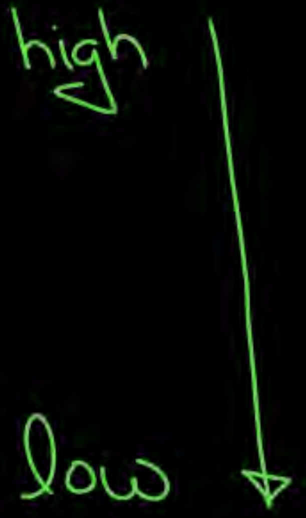
= is also an operator

int a;

printf("%d", ~~a = 20~~); 20

20

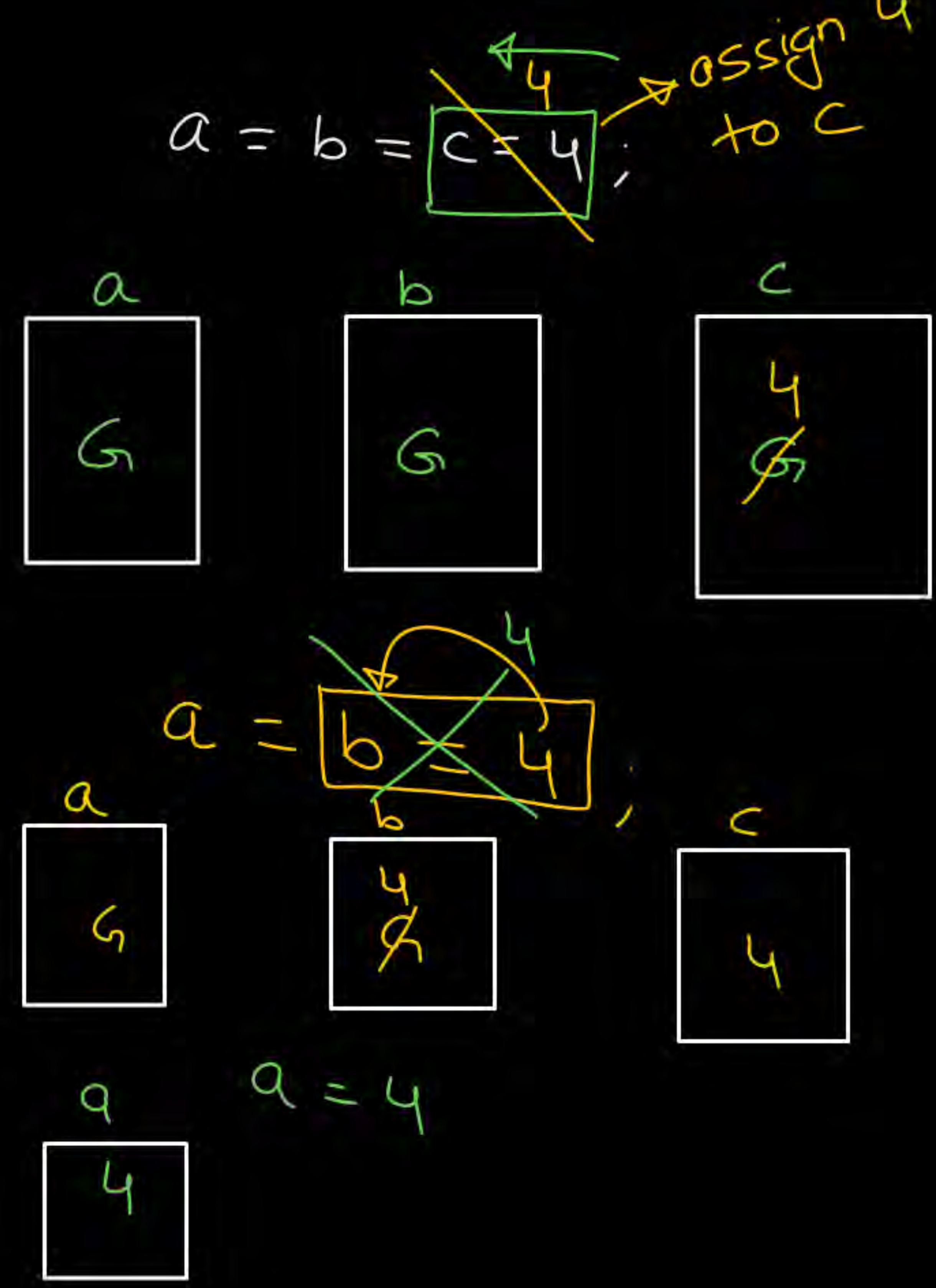




% × , /	L to R
+ , −	L to R
=	R to L

1.

```
int a,b,c;
a=b=c=4;
printf("%d %d %d",a,b,c);
```



2.

```
int a,b,c;
a = b = 4 = c;
printf("%d %d %d",a,b,c);
```

$a = b = \boxed{4 = c}$

4 = c ;

Lvalue Can not be Constant

Ud ke laat marega

THANK YOU GW SOLDIERS !