

CS & IT ENGINEERING



C Programming
Lecture No.3



Pankaj Sir



**TOPICS
TO BE
COVERED**

A yellow diamond-shaped sign with a black border and black text, mounted on a grey post. Below the sign is a white and orange striped construction barrier.

o1

Assignment Operator

o2

Arithmetic Operator

printf ("_____")
 └──→ Text

%d → short int, int

int a=10;

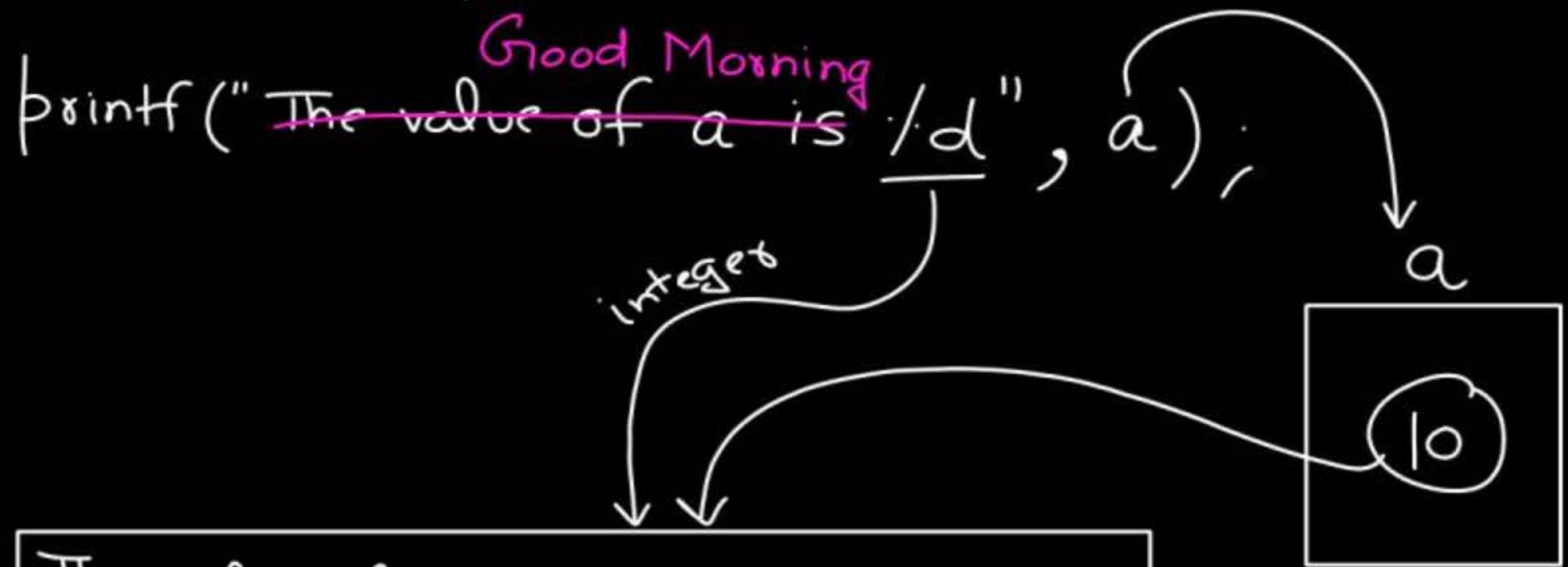
%c → char

printf("a"); a

%f → float

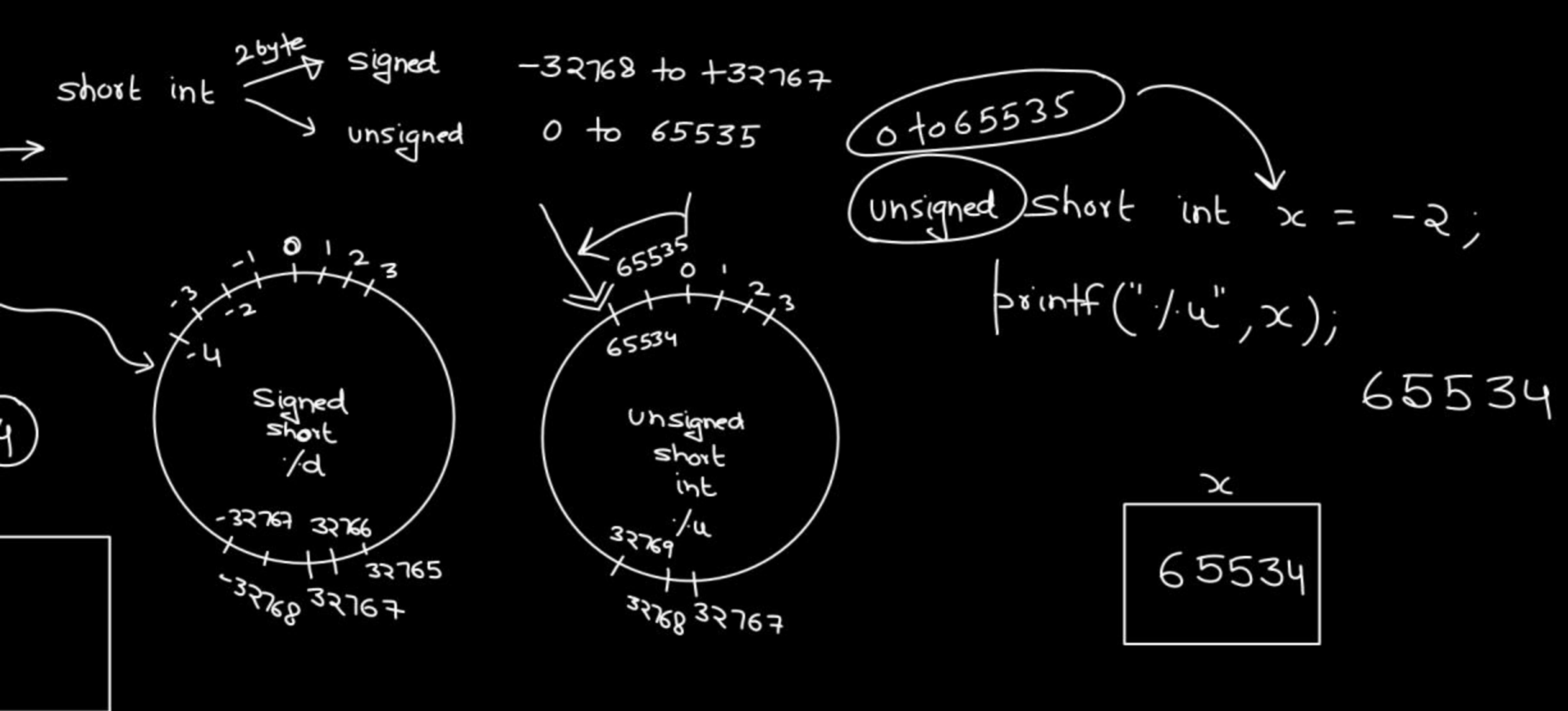
printf("10+10") 10+10

```
int a = 10;
```



The value of a is 10

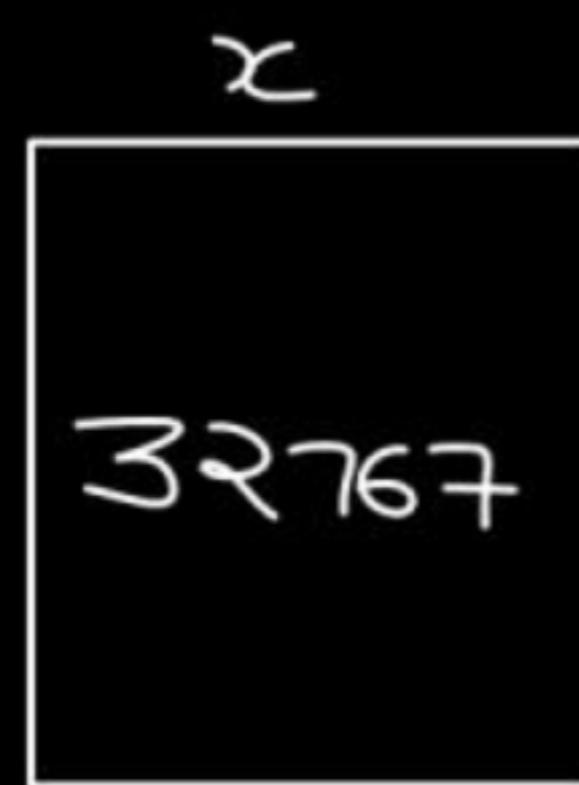
Good Morning 10



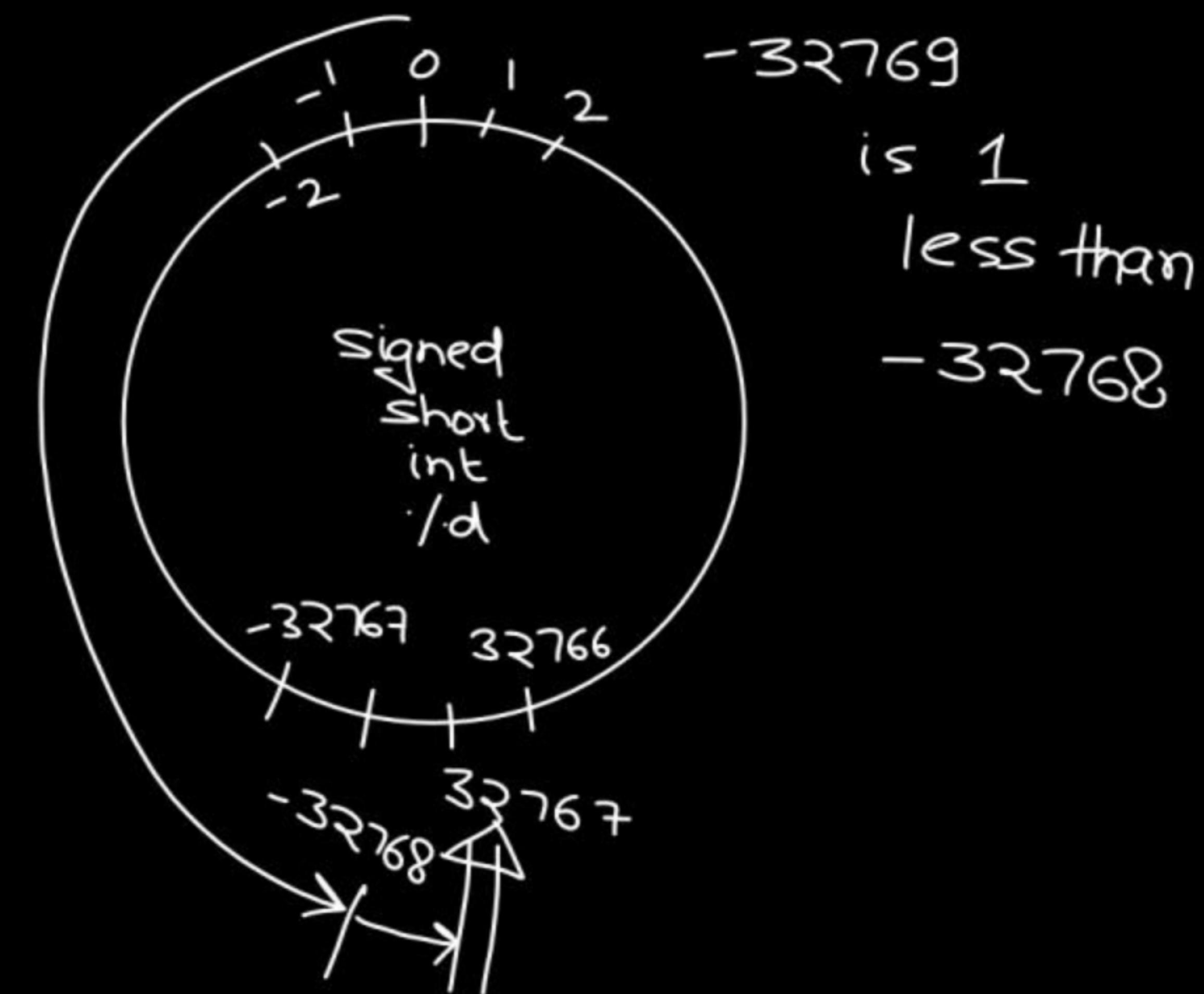
Q)

Short int x = -32769; { by default : x is signed }

printf("%d", x);



O/P: 32767



char data type

char ch = '@';

char → 1 byte

char ch = '#';

8 bits

1 symbol

Confusion?

2^8 values

Unsigned char : 0 to $2^8 - 1$

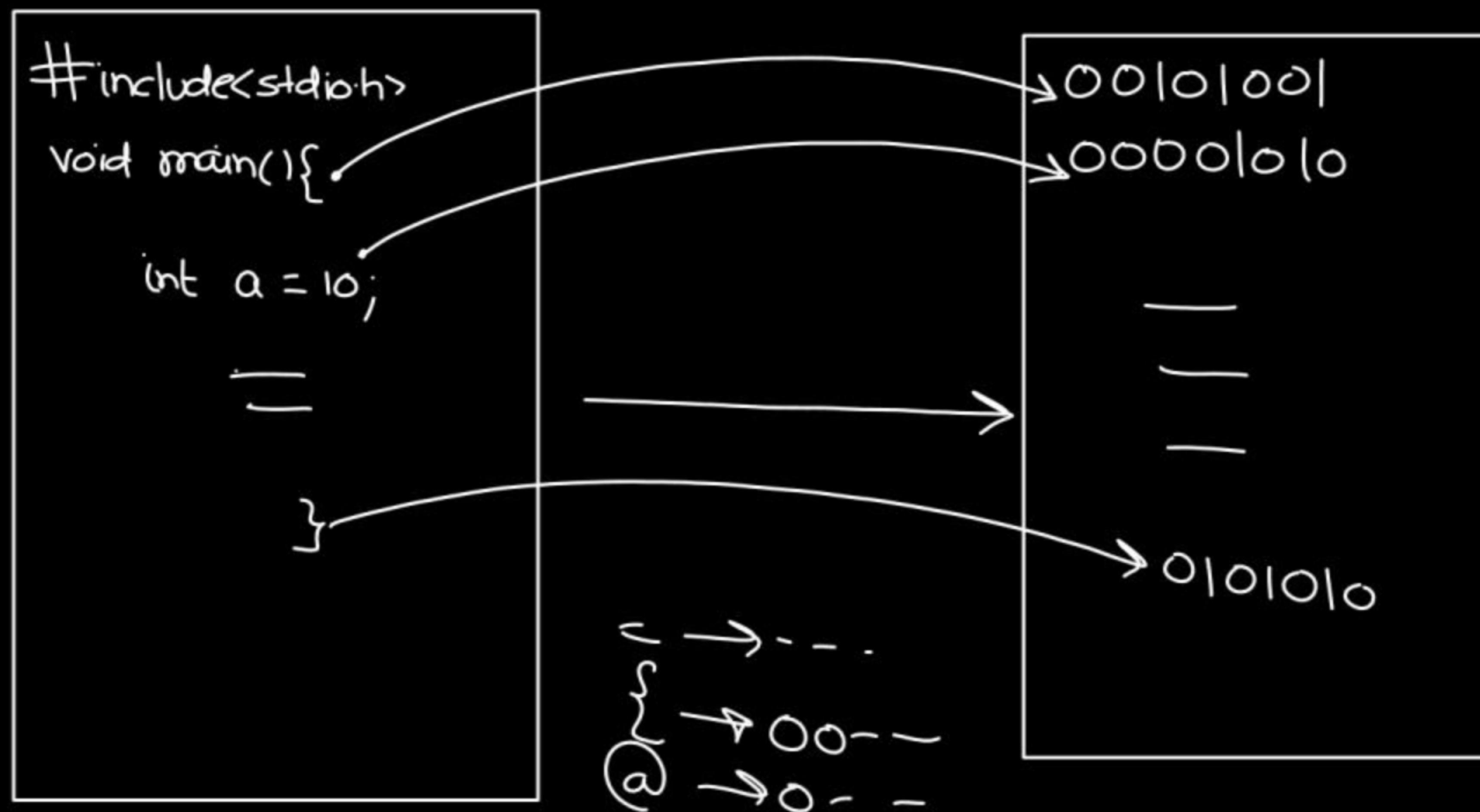
0 to 255

Signed char : -2⁷ to 2⁷ - 1

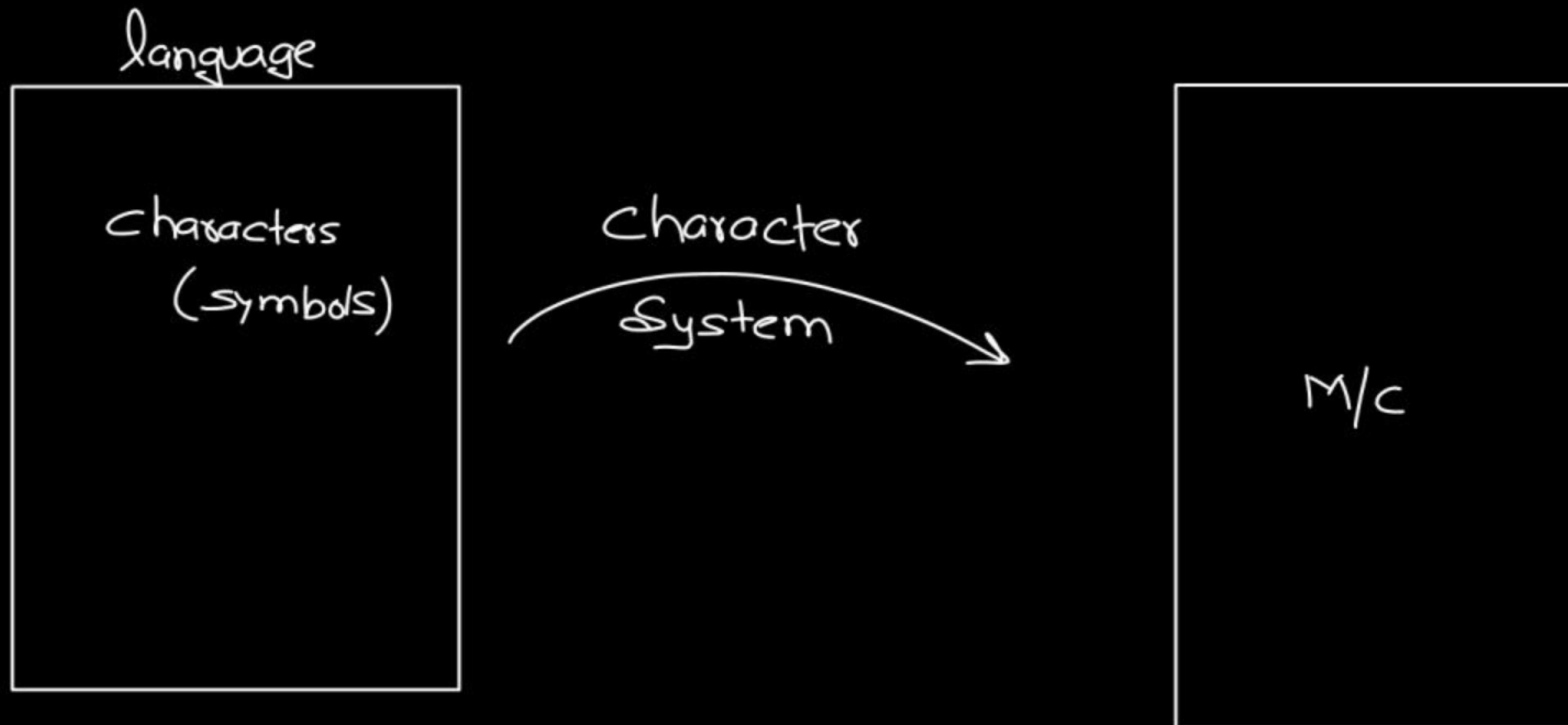
-128 to +127

CHARACTER SYSTEM

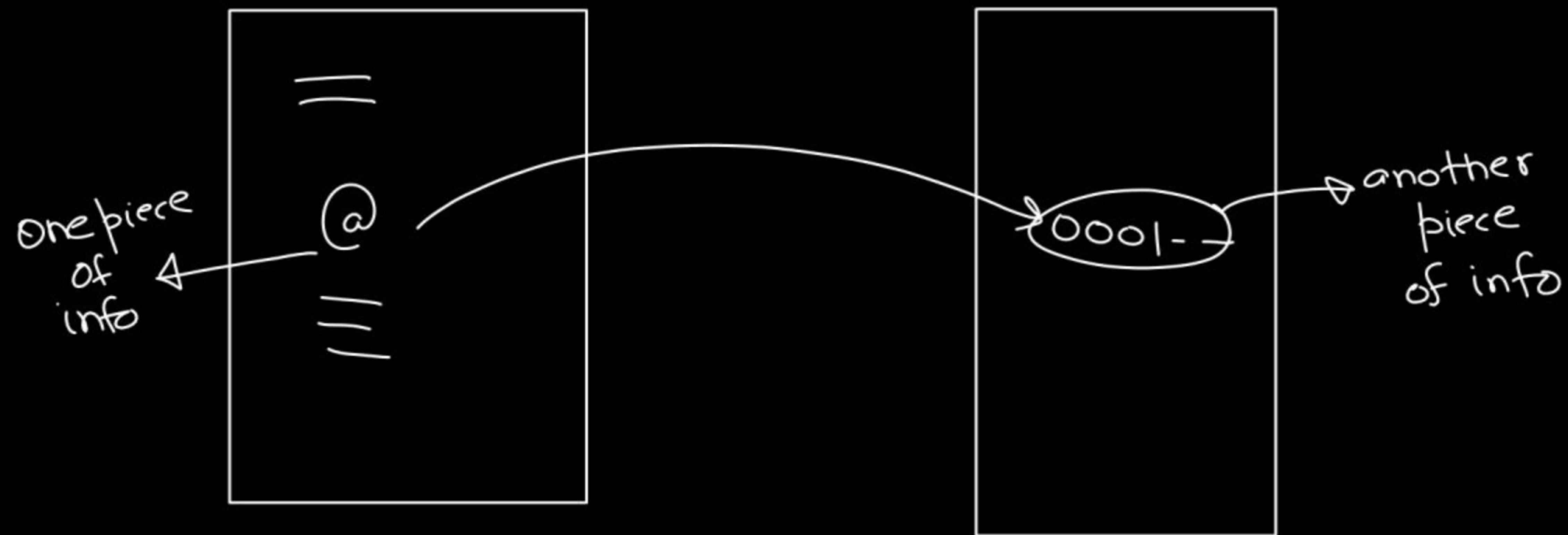
10 → 00001010



CHARACTER SYSTEM



ASCII

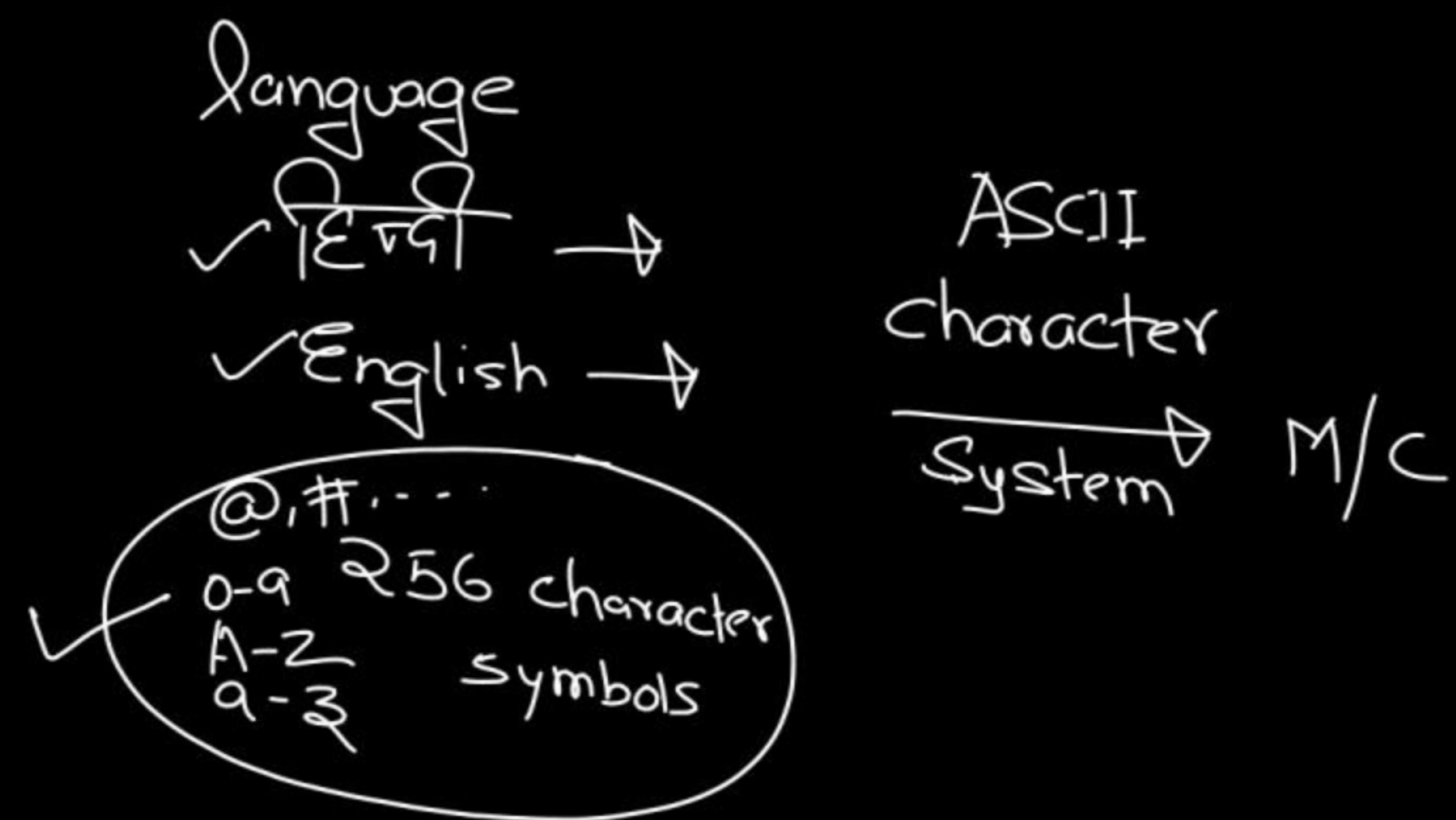


English Type

1 byte \Rightarrow 0 to 255

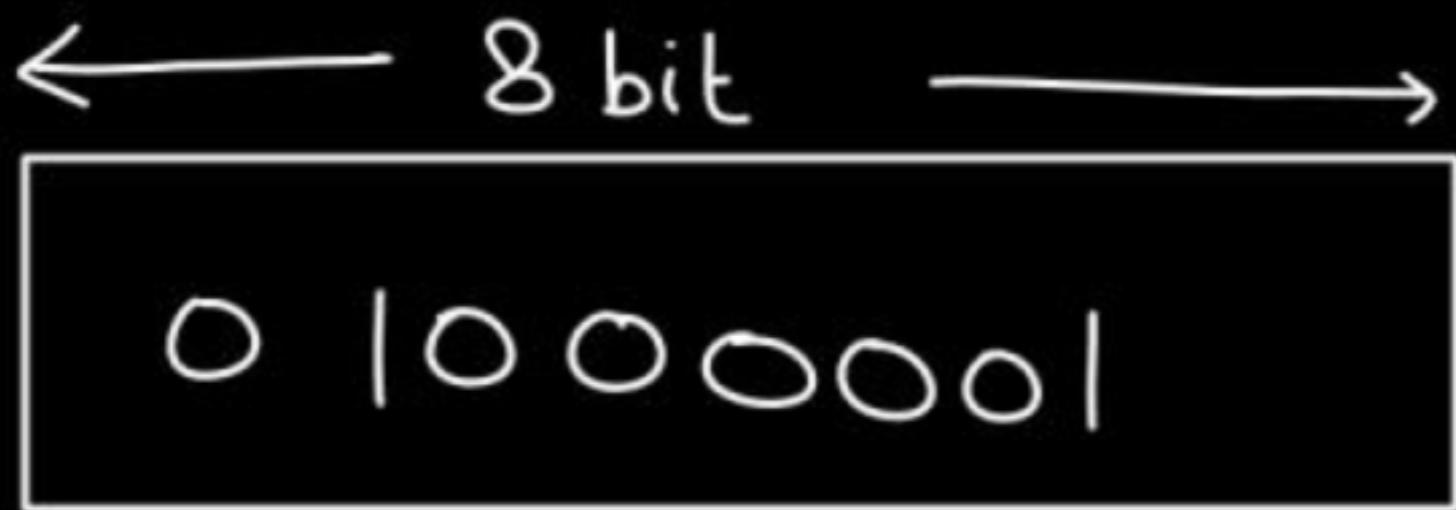
A	65	a	97	0	-	48
B	66	b	98	1	-	49
C	67	c	99	.	.	.
D	68	d	100	.	.	.
.
Z	90	ঃ	122			

1 byte



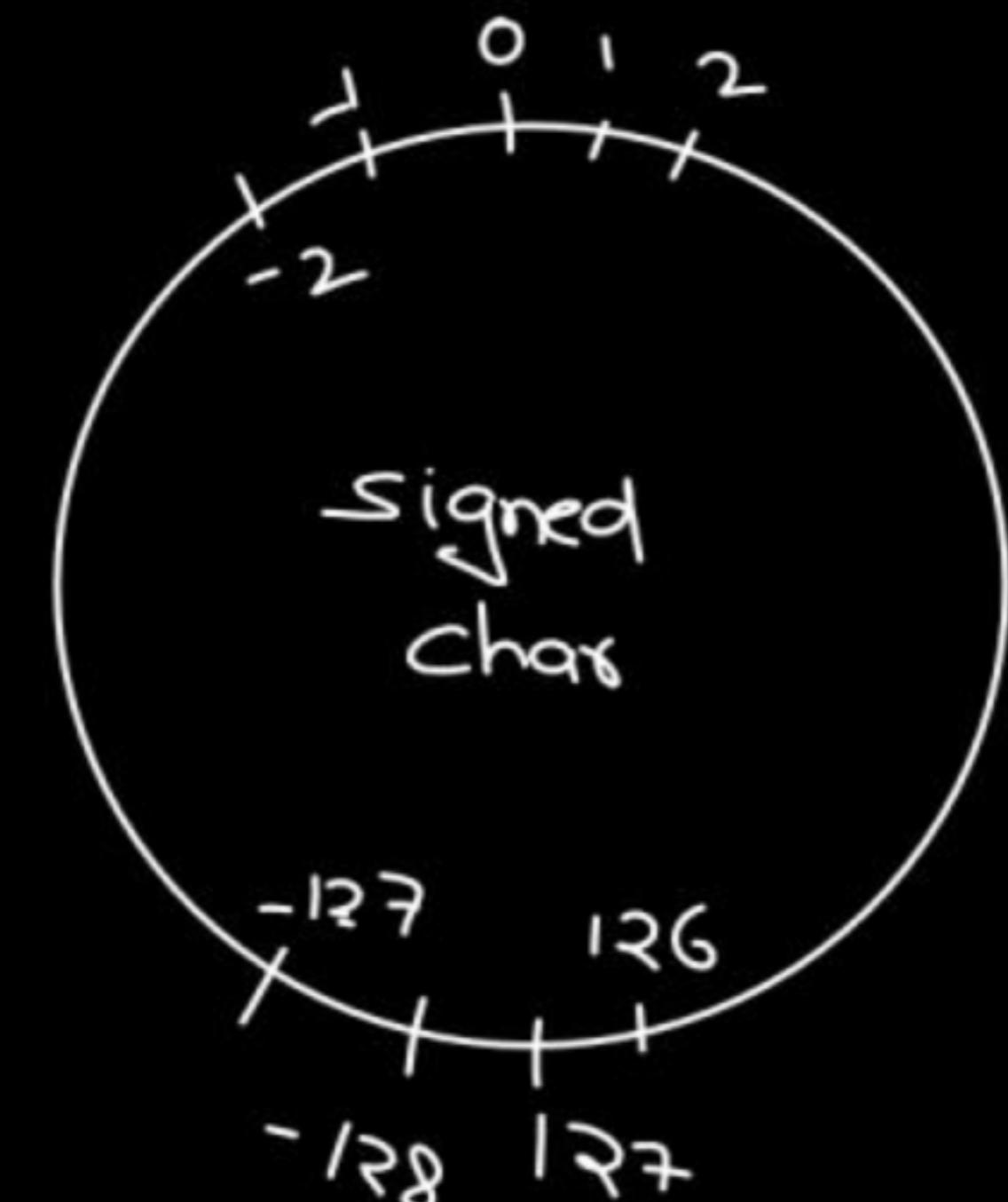
CHARACTER SYSTEM

char ch = 65; ✓



printf("%d", ch);

(65) ✓



$\%c \rightarrow$ character system (symbol)

2:

char ch = 129 ; { by default ch : signed }

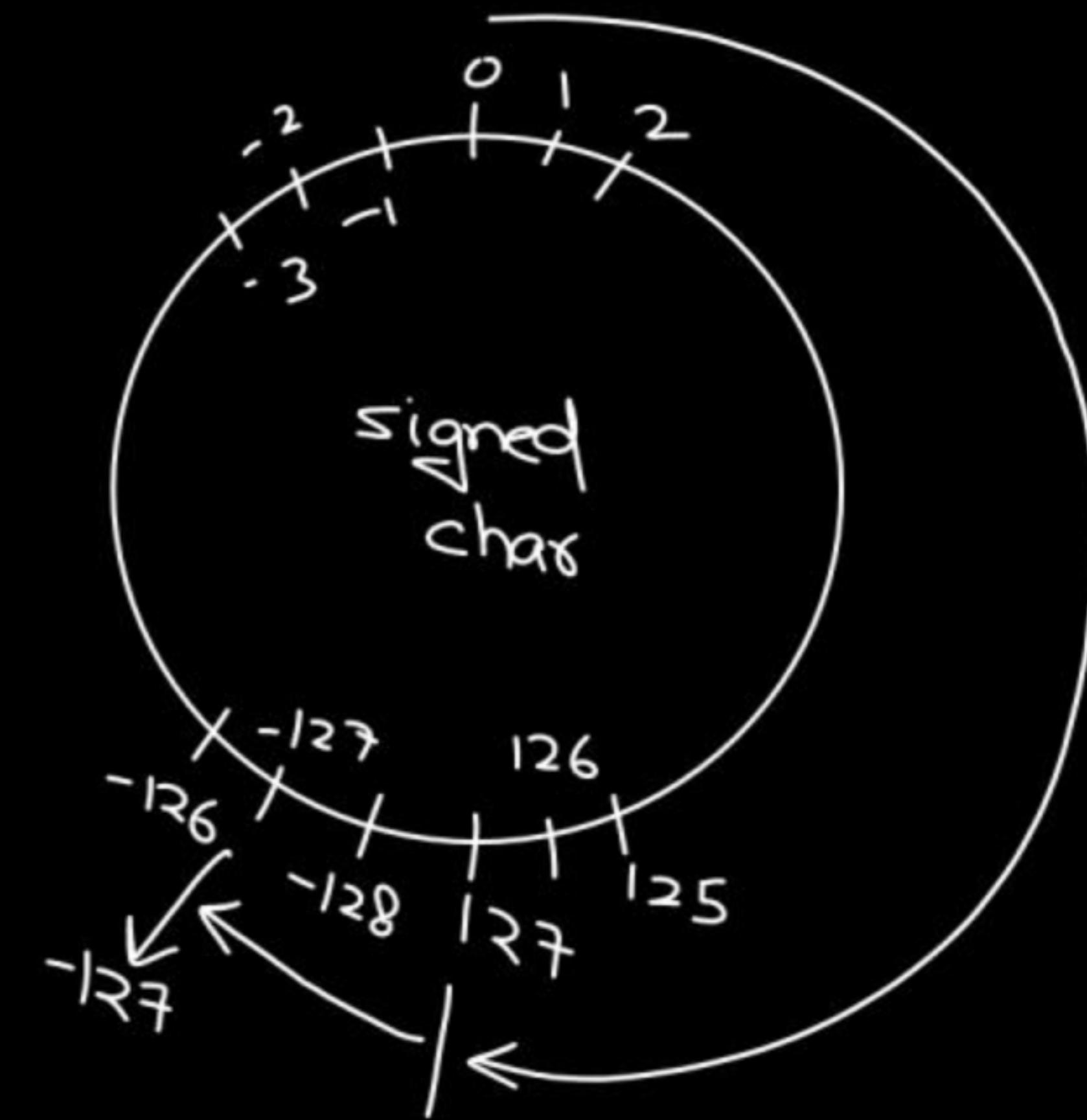
printf("%d", ch);

O/P:

-127



-128 to +127



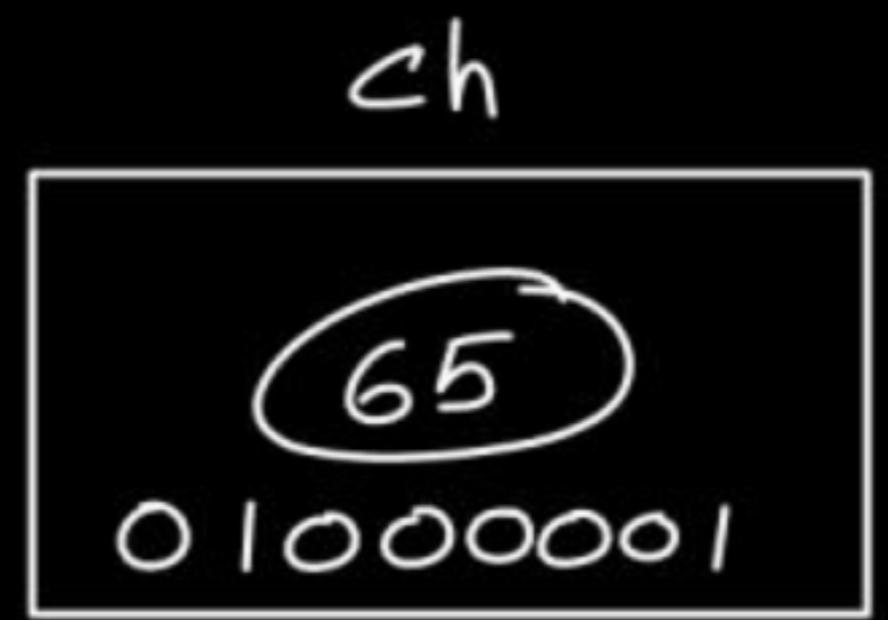
char ch = 65; ✓

printf("%c", ch);

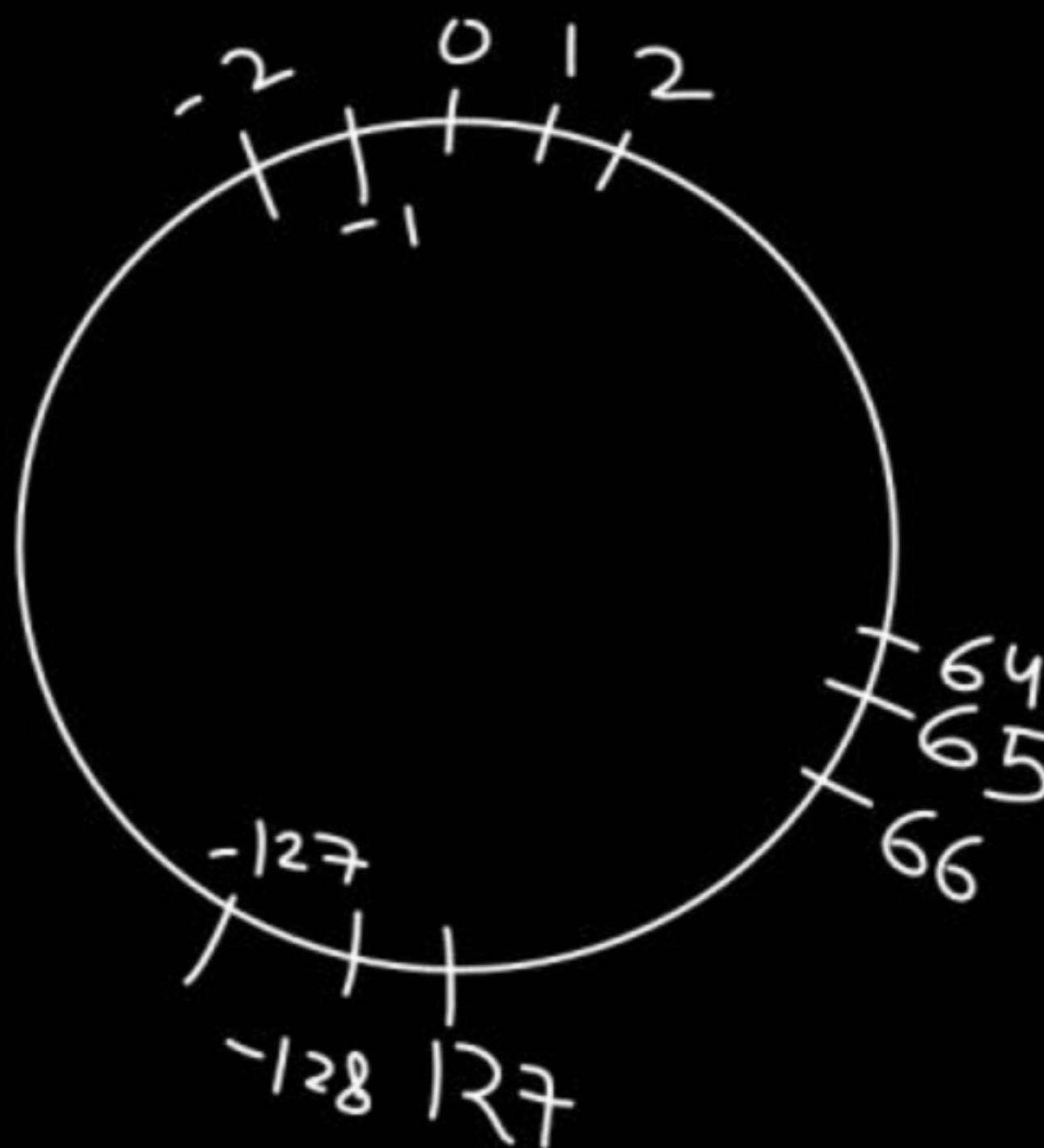


Character system
(symbols)

A



A ↦ 65



char ch = 'A' ;

printf("%d", ch); ⑥5
printf("%c", ch); ①A
↓
symbol

char ch = A X

Error

A ?

char ch = @ , X

@ ?

ch



```
char ch = -129;
```

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

127

```
char ch = -193;
```

```
printf("%c", ch);
```



Symbol

ASCII code

63

Signed
char

```
char ch = -191;
```

```
printf("%c", ch);
```

A



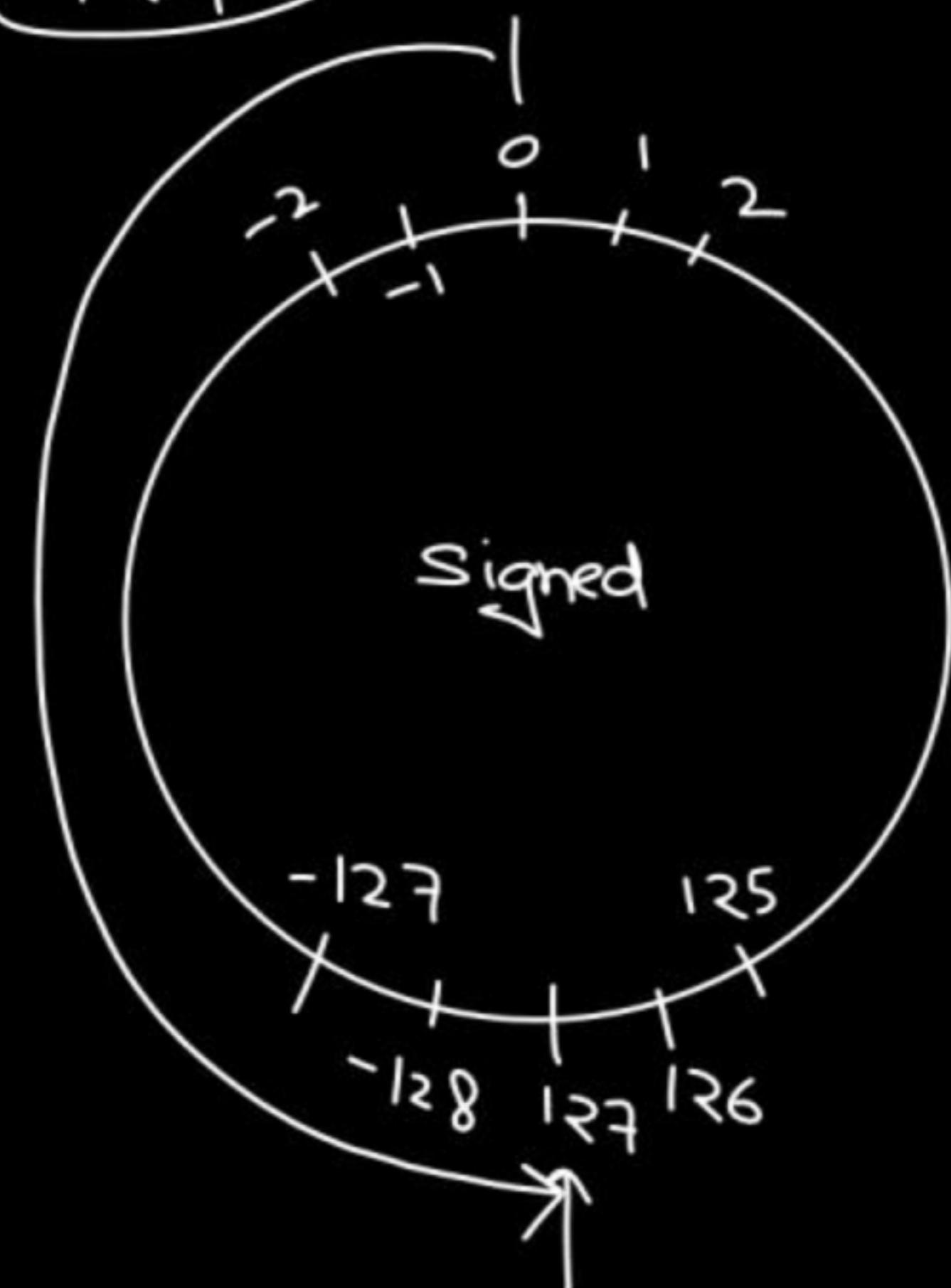
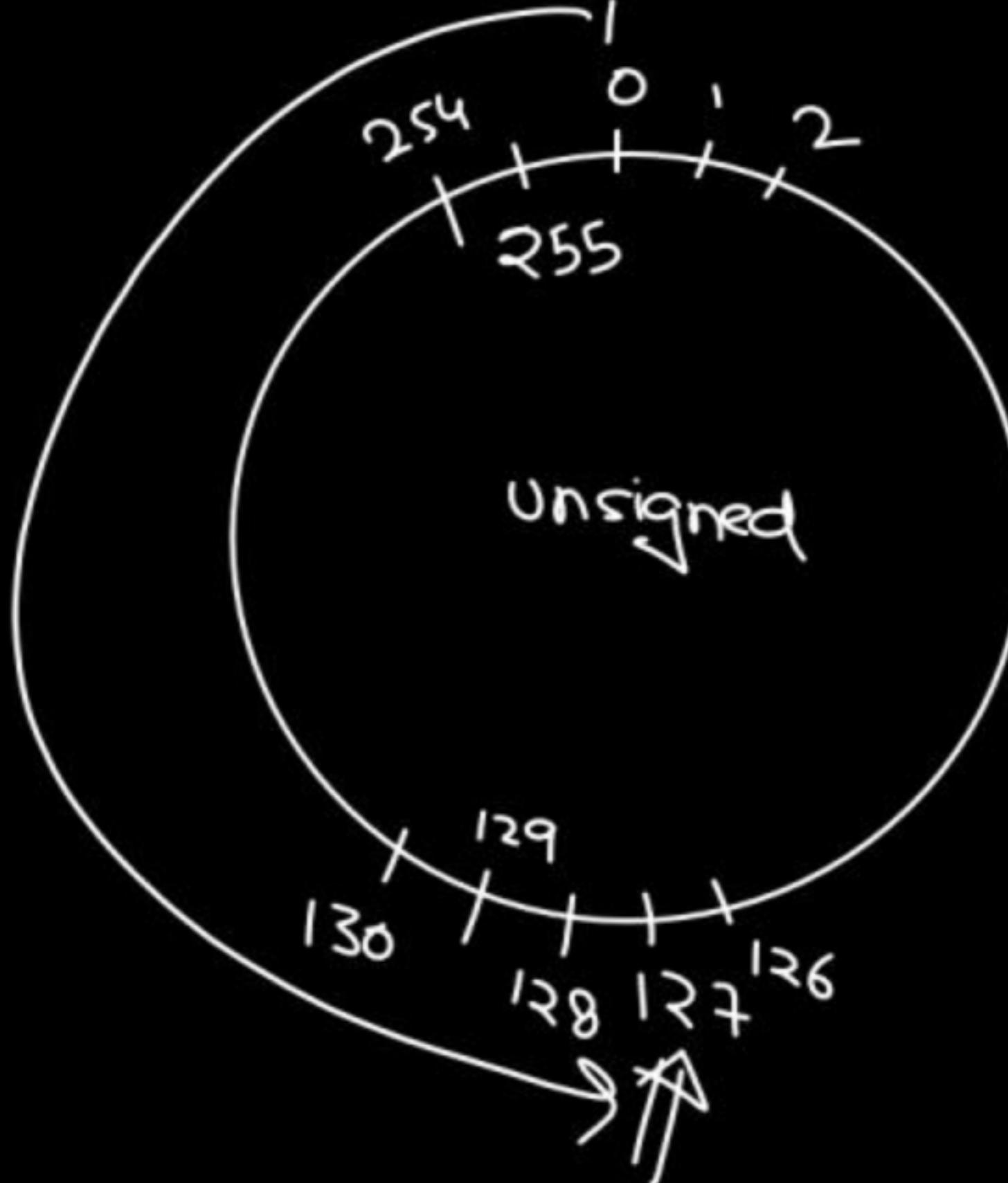
~~-129 = -128-1~~

-129 = 127

-130 = 126
-131 = 125

Unsigned char ch = -129;

printf("./d", ch); 127

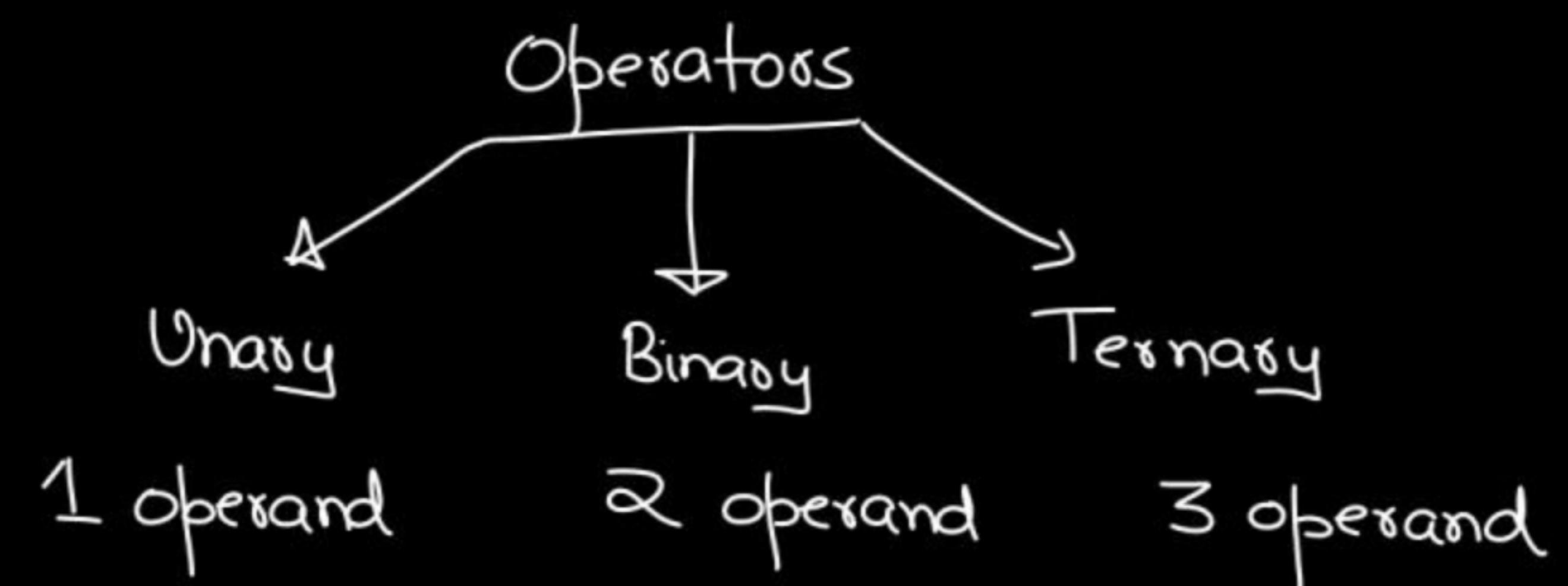


Operators

2 values
↓
operand $10 + 20 = 30$
↓
operator

-7 → value
↓
1 operand
↓
operator

(i) Every operation $\Rightarrow \{$
value
output
result $\}$



Assignment operators (=)

```
#include<stdio.h>
void main(){
    int a = 12;
    a = 200;
}
```



binary operator
2 operand
LHS = RHS

lo = a ; X
Error

Lvalue = Rvalue

int a,b,c;

a = 10; → literal

b = 20;

c = a + b * 2; ✓
 expression Exp X
 literal X

 c = a;

variable

Exp X

literal X

literal / constant ✓

variable ✓

expression ✓

a

10

b

20

c

10
~~50~~

$+,- \Rightarrow \text{unary}$ $+2$ -7.3

| Operand | Operand

binary : $+,-,\times,/, \%$ $\cancel{\times} \text{ BODMAS}$

$$\begin{array}{c} 6/2 \times 3 \\ \hline \swarrow \quad \searrow \\ 9 \quad 1 \end{array}$$

+,-,/,X

% Modulus operators :

$a \% b$: Remainder when a is divided by b.
value/output/result

12 % 5

2

printf ("%d", 12 % 5) 5 $\overline{12} \overline{10} \overline{2}$
 \ 2

$$5 \overline{)12} \{ 2.4$$

$\overline{0}$ ← remainder

% operator : both operands must be integer type otherwise Error.

$$a = 12.3 \% 2; \underbrace{\text{Error}}$$

Q/P

$$2 \% 7 : 2$$

$$3 \% 11 : 3$$

$$5 \% 6 : 5$$

$$\begin{array}{r} 7 \sqrt{2} \backslash 0 \\ \quad 0 \\ \hline \quad 2 \end{array}$$

remainder

(ii) What if operand is -ve

→ Standard : undefined ✓

→ a / b : sign of result is same as the sign of first operand i.e. a

$$-12 \% 5 \Rightarrow \begin{matrix} \text{sign} \\ -\text{ve} \end{matrix}$$

$$12 \% -5 \Rightarrow +\text{ve}$$

$$-12 \% -5 \Rightarrow -\text{ve}$$

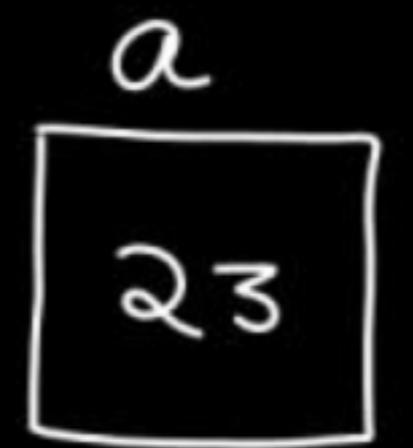
$$12 \% 5 \Rightarrow +\text{ve}$$

int a ;

$$a = 3 + \underbrace{4 \times 5}_{\textcircled{1}} ;$$

$$a = 3 + \underbrace{20}_{\textcircled{2}}$$

$$a = 23$$



Precedence : ① %, /, \times

② +, -

③ =

$$a = \underbrace{3 \times 12 / 6}$$

precedence same

① $\times, /, \%$

Associativity:

Left to right

$$a = \underbrace{3 \times 12 / 6}_\text{①}$$

(i) \times
(ii) $/$

} same

$$a = 36 / 6$$

$$a = 6$$

$$\boxed{6}$$

`printf("%d", 2+3); valid 5`

`printf("%d", 12*3); valid 36`

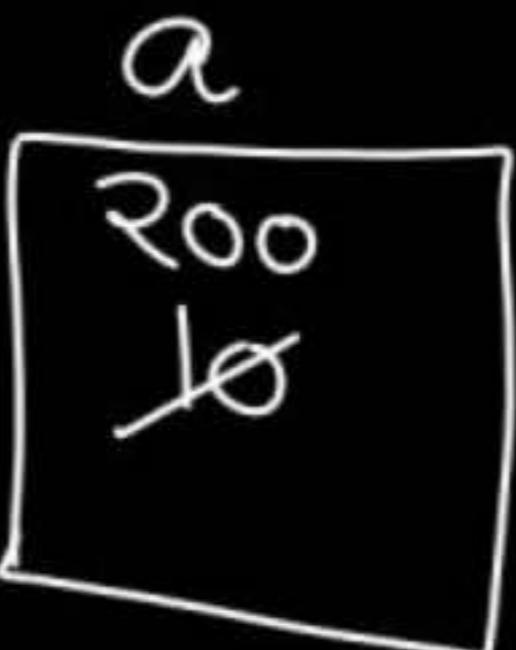
Evaluate

[
int a = 10;
`printf("%d", a = 200);`

= is an
operator

O/P: 200

↳ EQ operator
on result



int a, b, c;

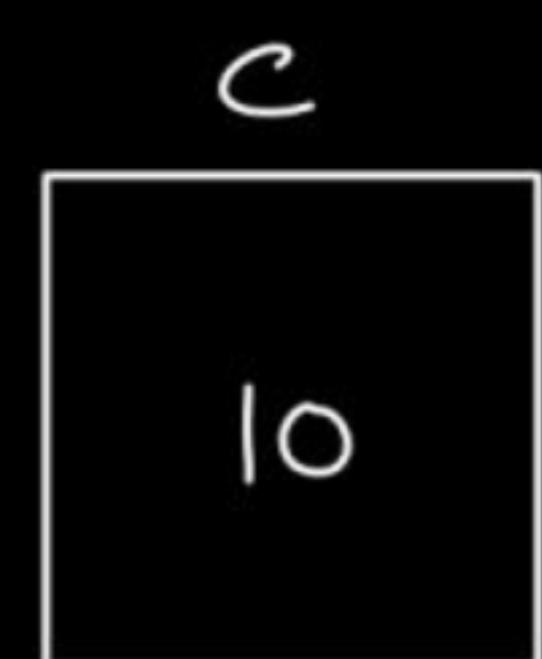
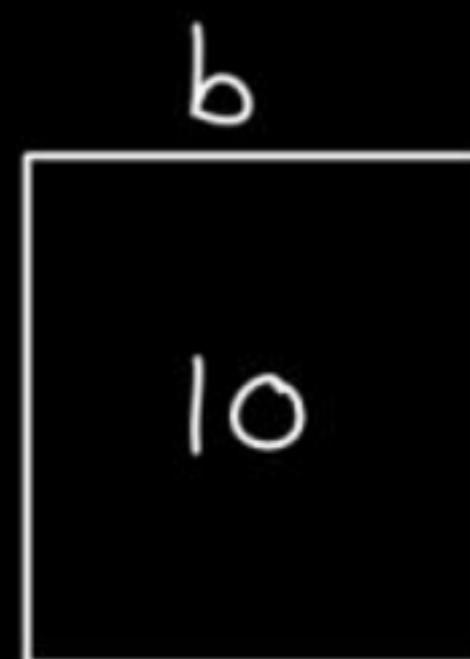
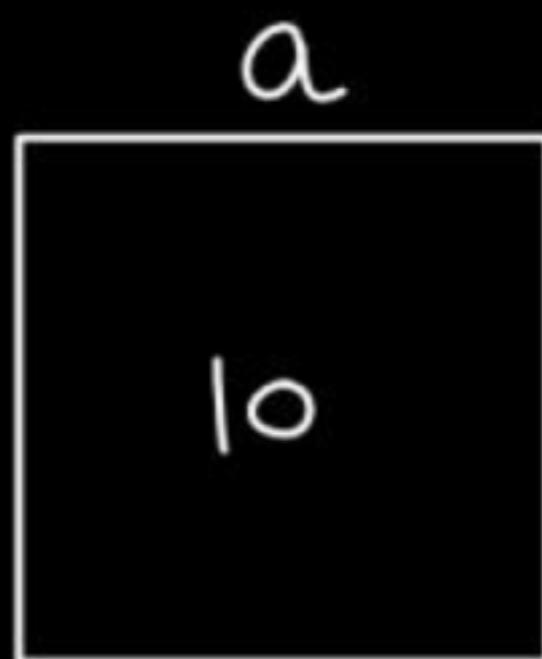
a = b = c = 10;

a = b = 10

a = 10

=

Right to left



$$\text{result} \rightarrow 30$$
$$10 + 20 - 30$$

```

int a,b,c;
a = b = 10 = c;           ↗ L value : Error
                           is literal
printf("%d %d %d",a,b,c);

```

O/P : ?

Result/behaviour of an operator : based on its operand

$$\textcircled{1} \quad 12/5 = 2$$

$\downarrow \quad \downarrow$
int int

int, int \Rightarrow int

$$\textcircled{2} \quad 12.0/5 \Rightarrow 2.4$$

$\downarrow \quad \downarrow$
float int float

$$\textcircled{3} \quad 12/5.0 \Rightarrow 2.4$$

int float float

$$\textcircled{4} \quad 12.0/5.0 \Rightarrow 2.4$$

float float

Q

int a;

$$a = 4.0 \times 3 / 5 + 6 ;$$

printf ("%d", a);

$$4.0 \times 3 / 5 + 6$$

Error

$$12.0 / 5 + 6$$

\downarrow
float

Relational operators

- (i) <
 - (ii) >
 - (iii) <=
 - (iv) >=
 - (v) ==
 - (vi) !=
- binary
operators

LHS < RHS

10 < 20 : Is 10 less than 20 ?

Statement is TRUE

1

30 < 15 : Is 30 less than 15 ?

No
False

O/P
value : 0
result

printf("%d", $10 \leq 20$), 1

printf("%d", $10 < 1$); 0
false

printf("%d", $10 > 30$); 0

printf("%d", $10 > 3$); 1

\leq : $a \leq b \rightarrow$ (i) either $a < b$
(ii) a OR
is equal to b

printf("%d", $10 \leq 20$); 1

printf("%d", $10 \leq 10$); 1

printf("%d", $10 \leq 1$); 0

$a \geq b \longrightarrow$ Either $a > b$
OR
 a is equal to b

1

printf("/d", 10>=3); 1

printf("/d", 1>=6); 0

printf("/d", 6>=6); 1

$10 == 10$: Is 10 equals to 10 ?

True 1

`printf("%d", 10 == 10)` 1

$a != b$: Is the value of a not equals to value of b ?

$10 != 20$: Is 10 not equals to 20 ? True O/P

`printf("%d", 10 != 20);` 1

$2 != 2$: Is 2 not equals to 2 ? False 0

`printf("%d", 2 != 2);` 0

relational operators

<, >, <= , >= . (high)

= = , != (low)

Left to right

1) unary +, -

2) /, *, /] L to R

3) +, -] L to R

4) <, <= , >, >=] L to R

5) = , !=] L to R

6) = (assignment)] R to L

