





# Data Types

Comprehensive Course on C- Programming

# CS & IT Engineering

C Programming  
Data Types

Lecture Number- 03

By- Pankaj Sir



# Topics

*to be covered*



- 1 Data Types

$n$  bits



$2^n$  possible values

$$\frac{a^m}{a^n} = a^{m-n}$$

unsigned  
0 to  $2^n - 1$

signed

$\frac{2^n}{2^1}$   
 $2^{n-1}$

$-2^{n-1}, \dots, -1$

0 to  $2^{n-1} - 1$

$-2^{n-1} \text{ to } 2^{n-1} - 1$

int  $\Rightarrow$  2 bytes



16 bits



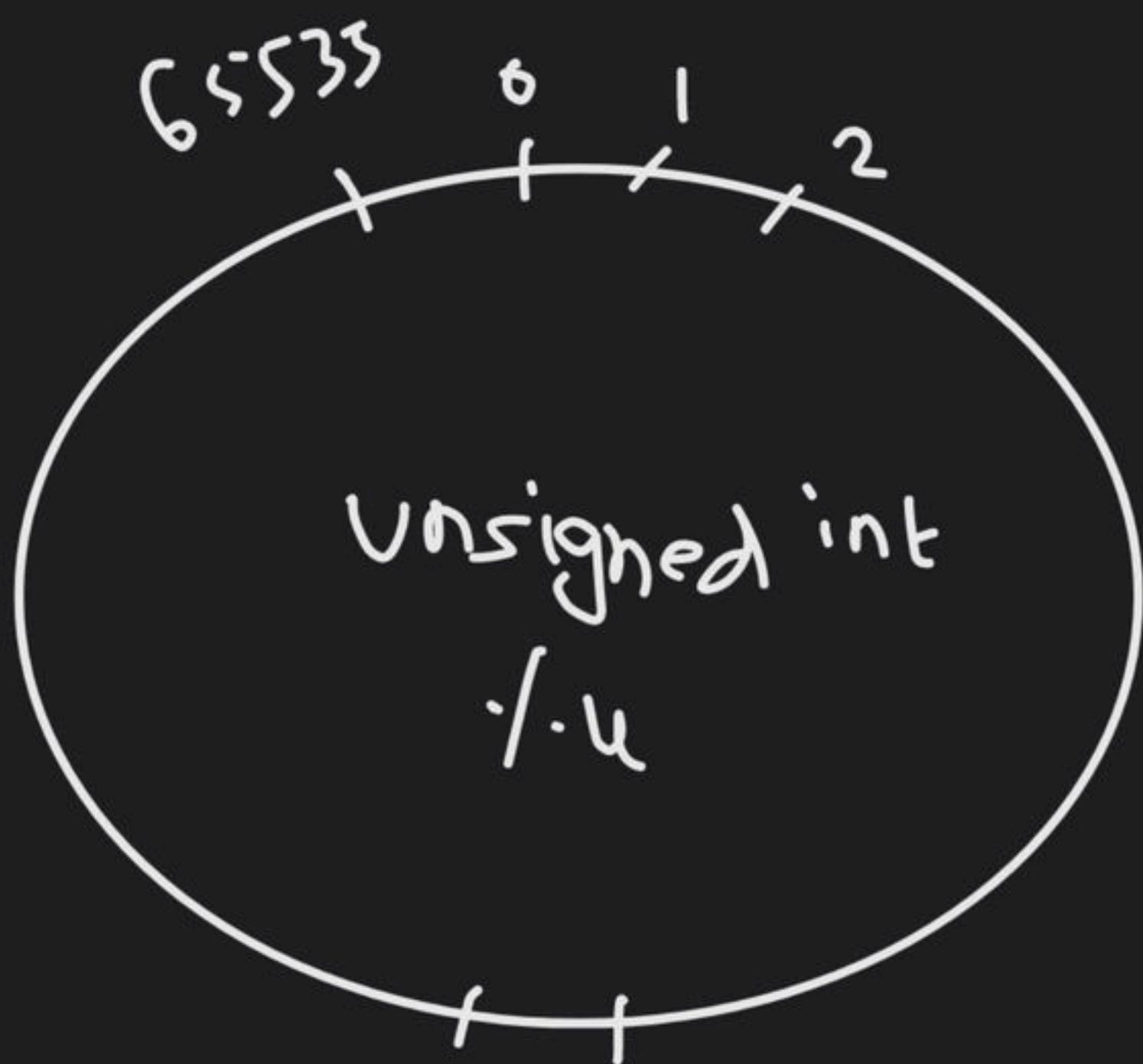
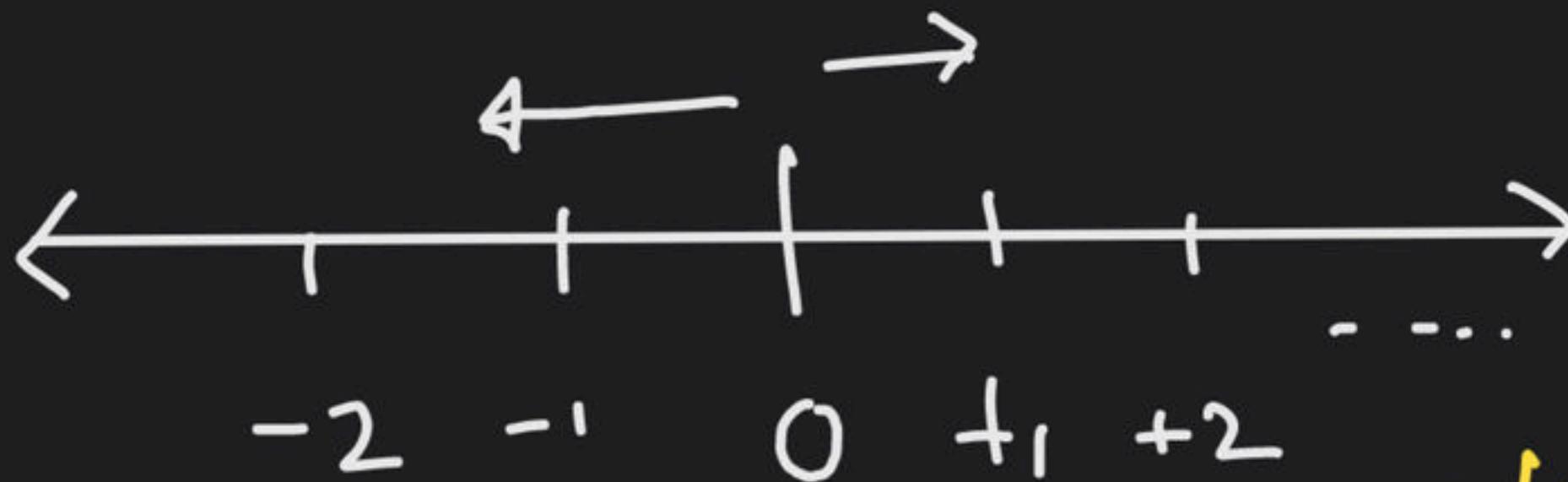
65536

unsigned

6 to 65535

signed

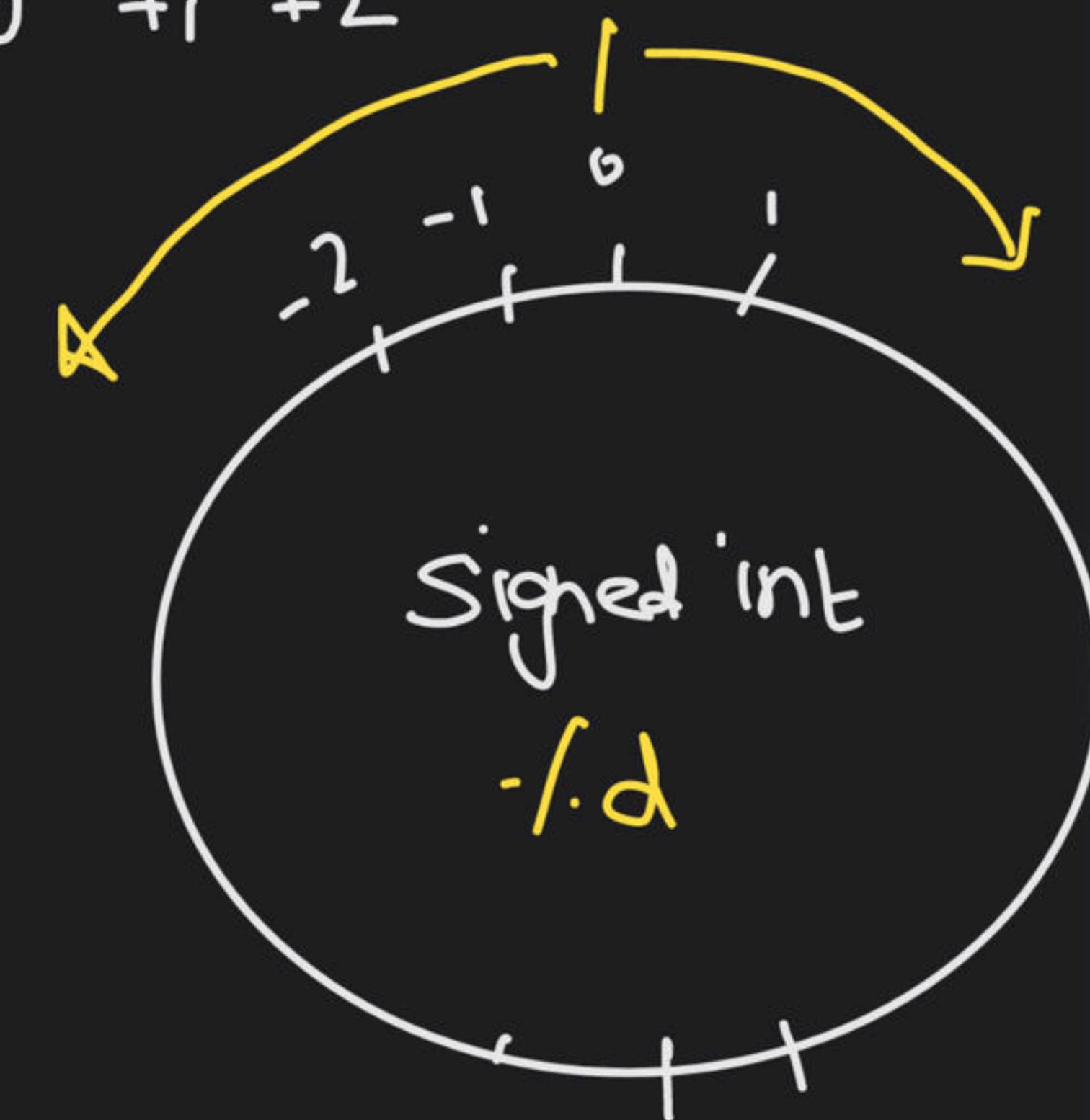
-32768 to +32767



Unsigned int

$\therefore u$

32768 32767



Signed int

$\therefore d$

-32768 32767

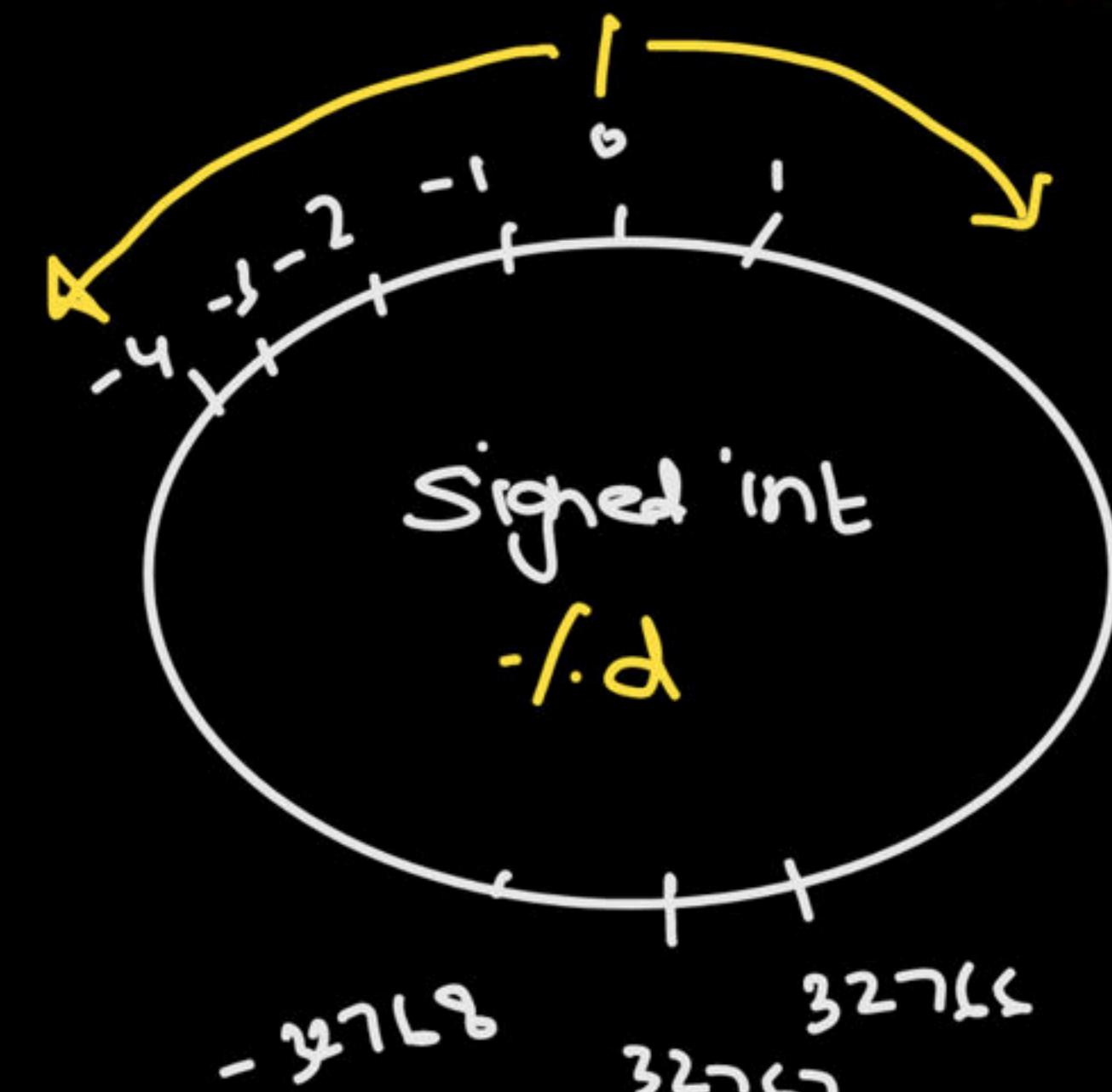
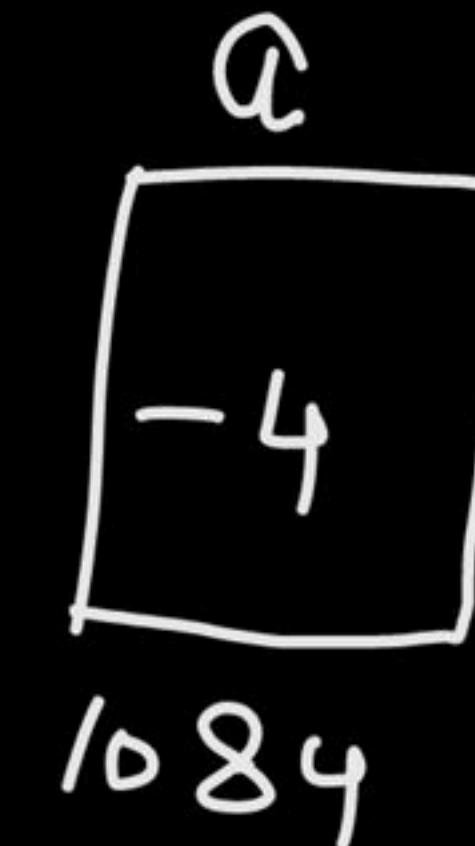
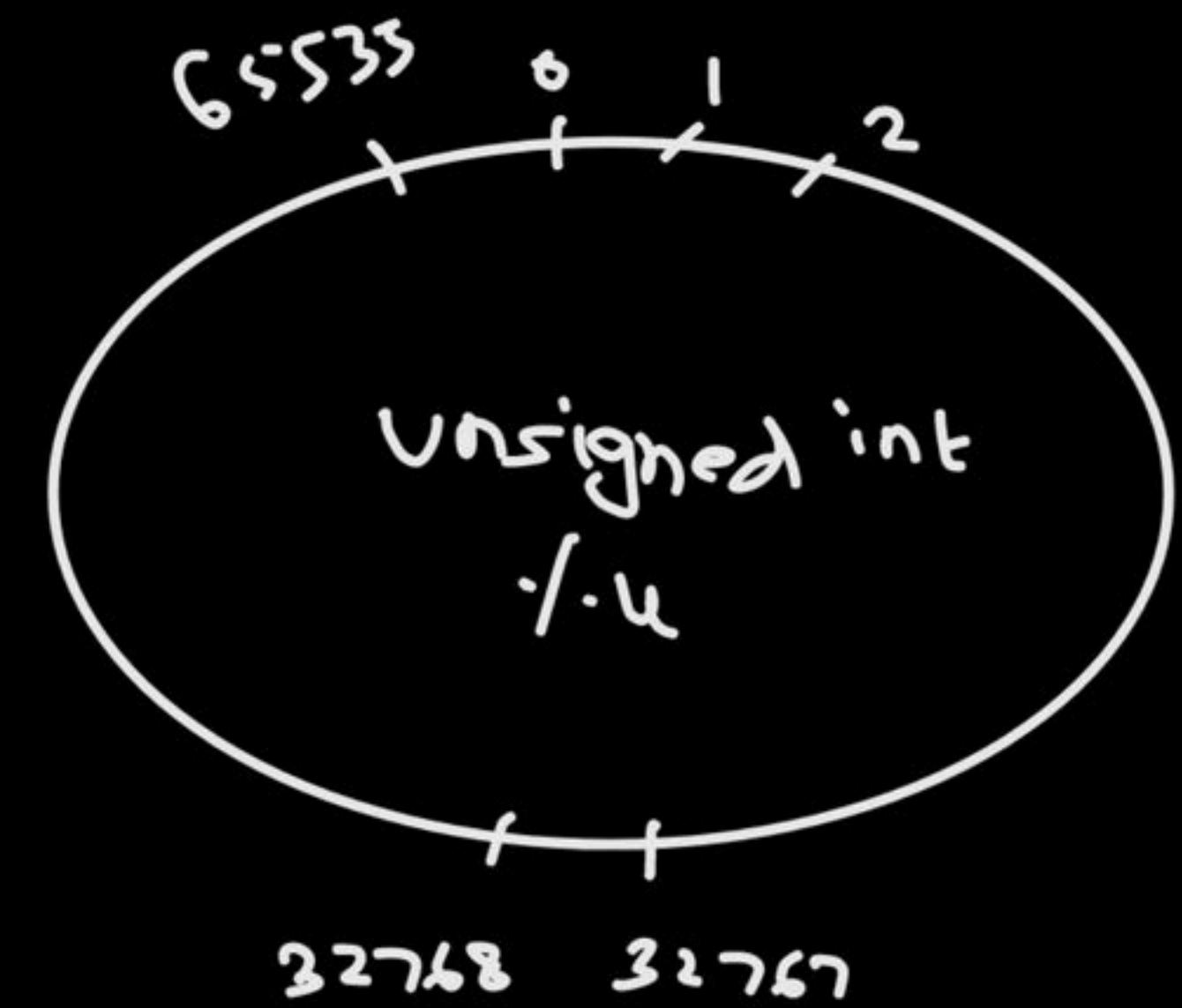
```
#include<stdio.h>
void main(){
    signed int a = -4;
```

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



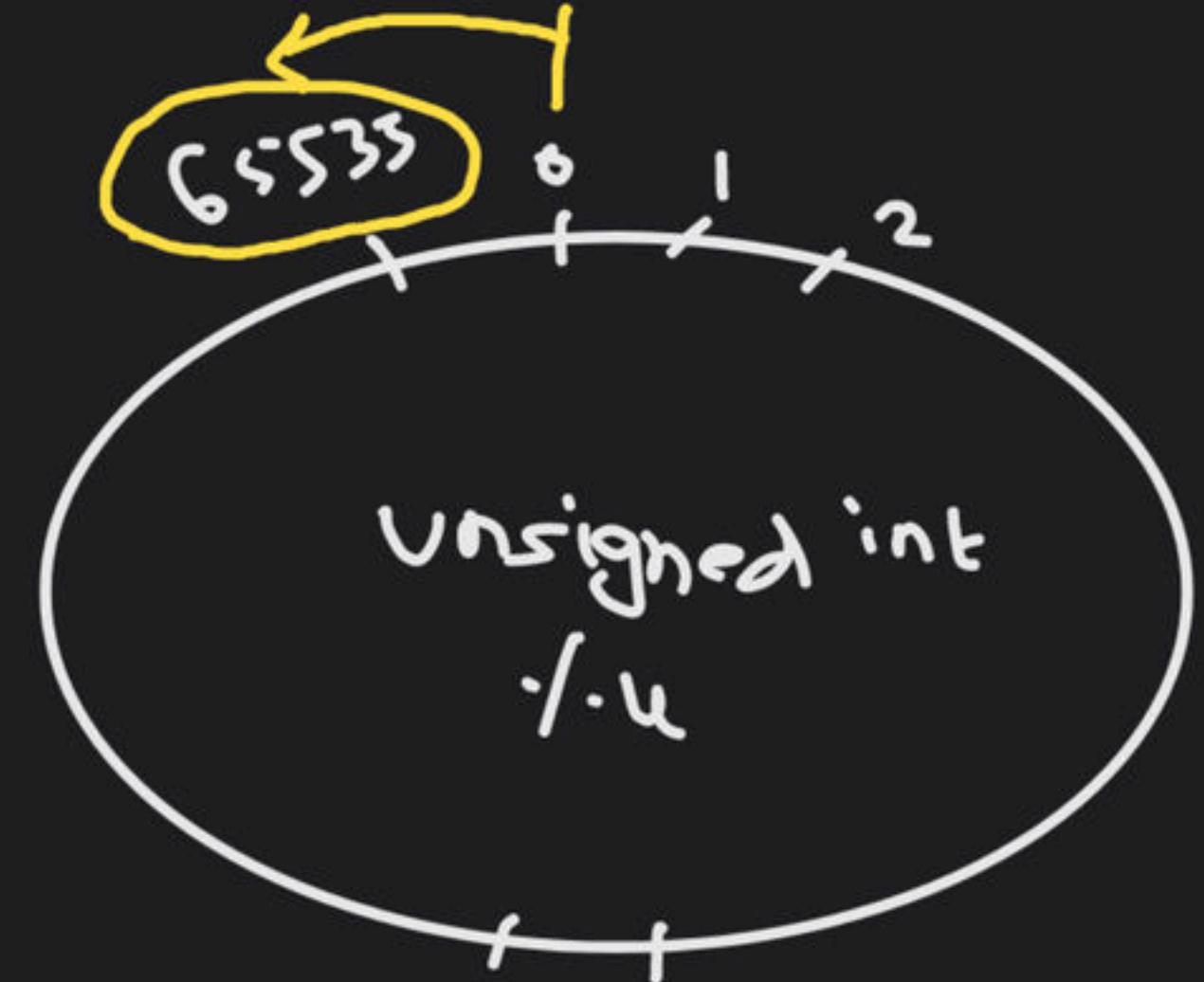
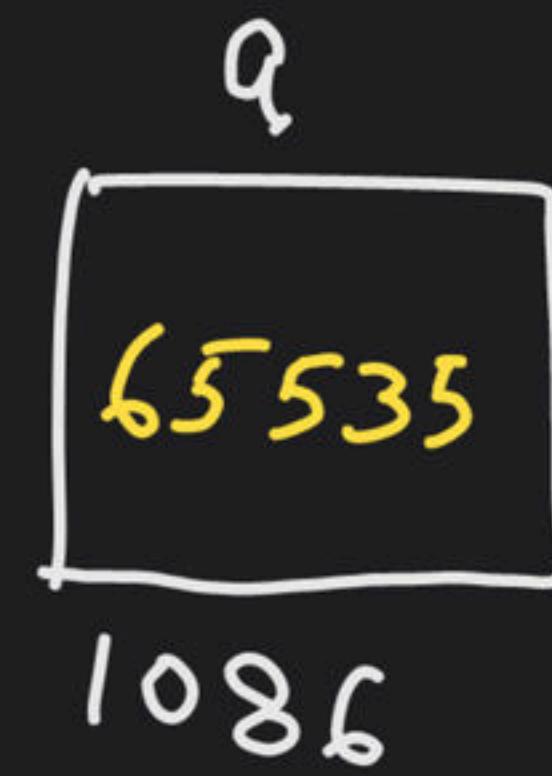
```
}
```

-4



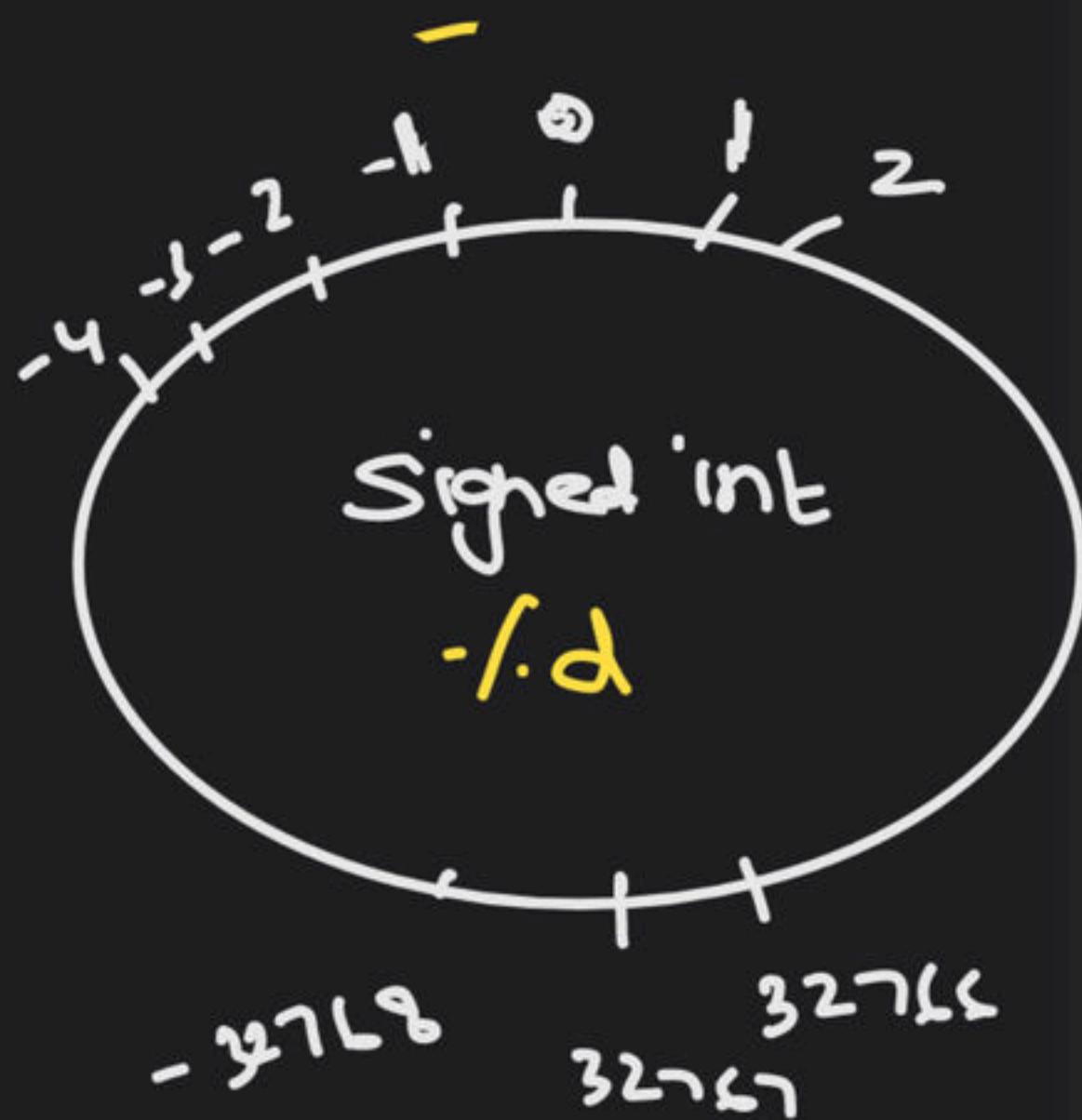
```
#include <stdio.h>
void main() {
    unsigned int a = -1;
    printf("./u", a);
}
```

O/P: 65535



a

-1  $\Rightarrow$  unsigned value  
large +ve value



2 byte

4 byte

↓

$2^{32}$

values

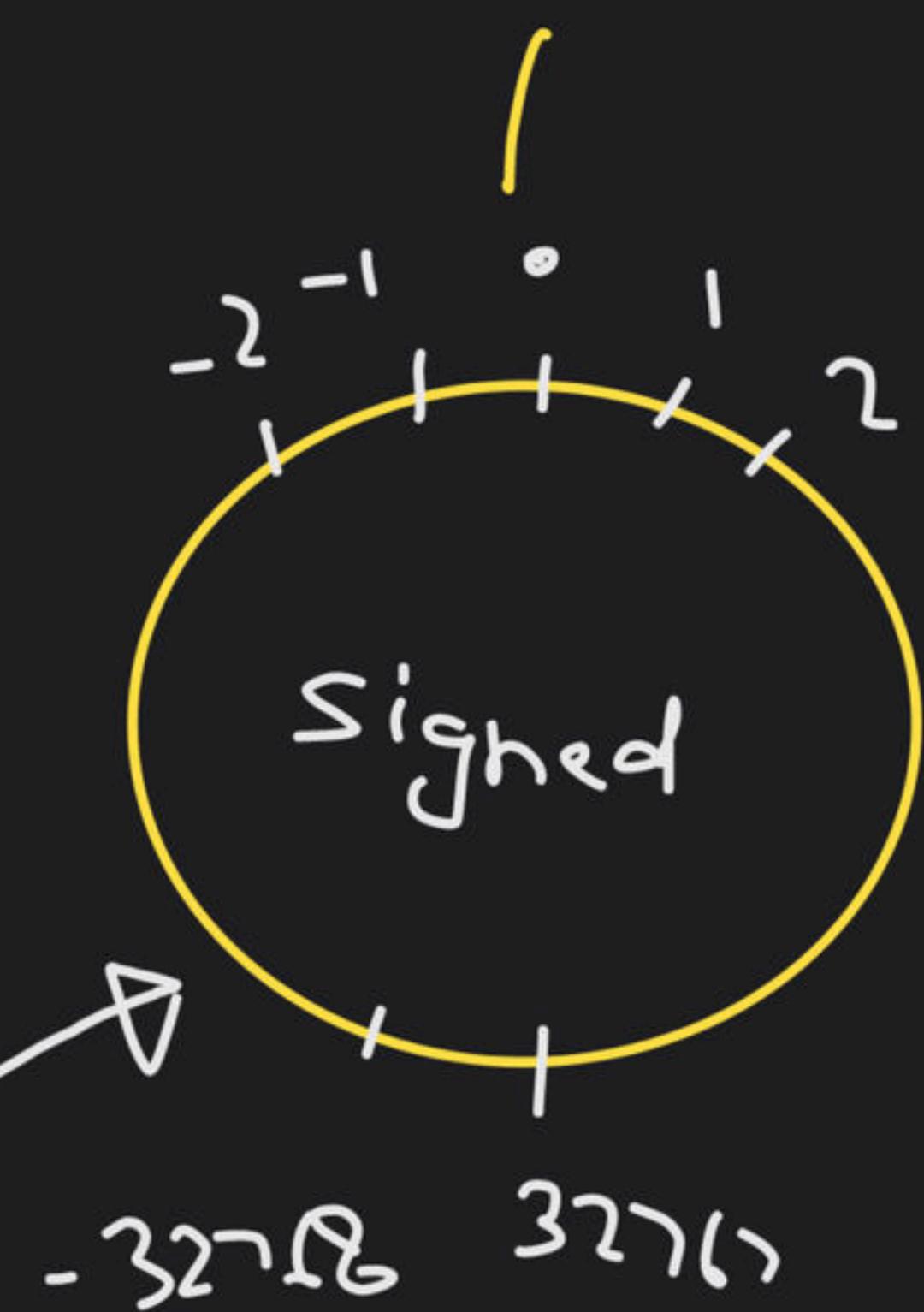
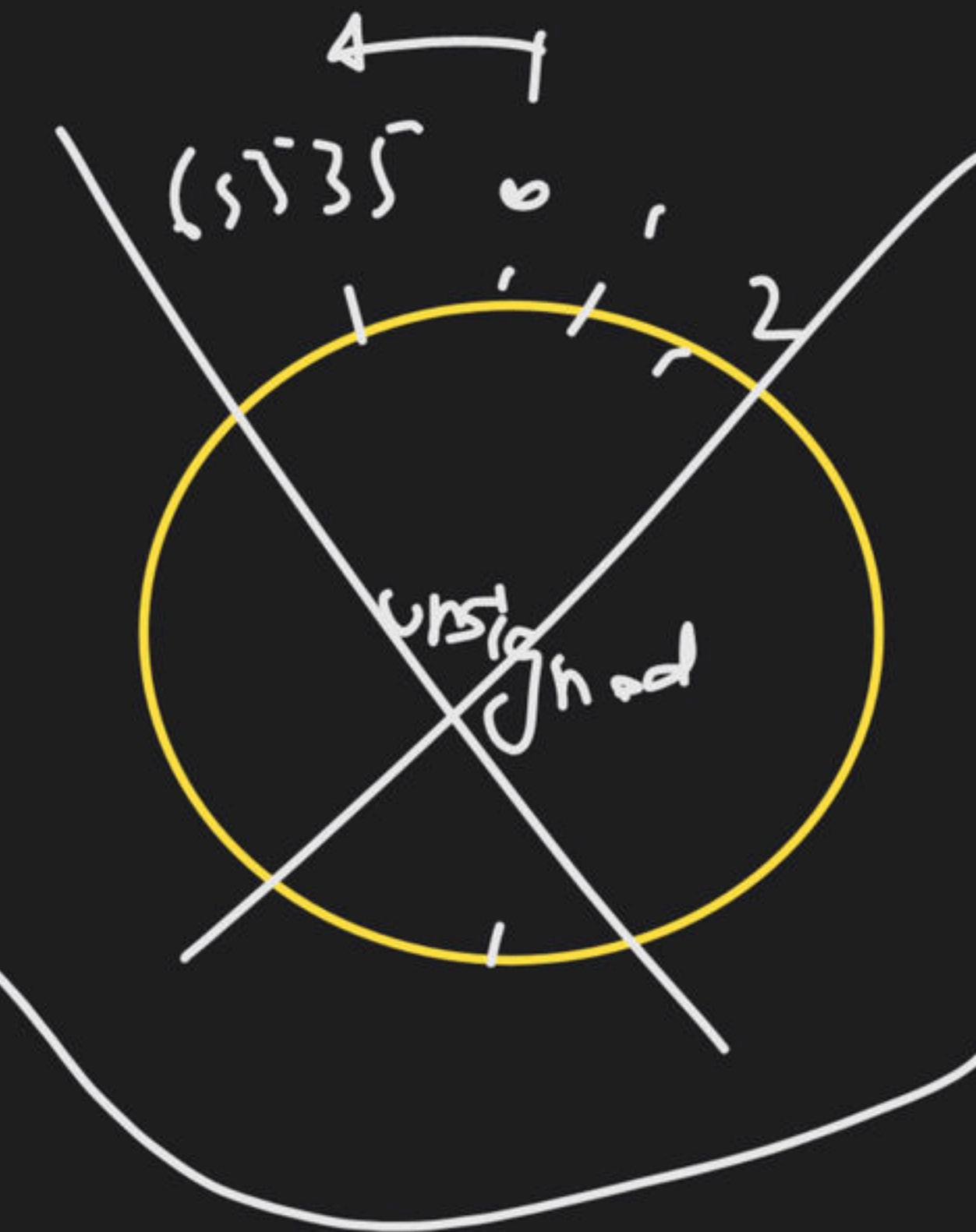
Cyclic property

int  
long int,  
char

```
#include <stdio.h>
Void main() {
    Unsigned int i=-1;
    printf("%d", i);
}
```

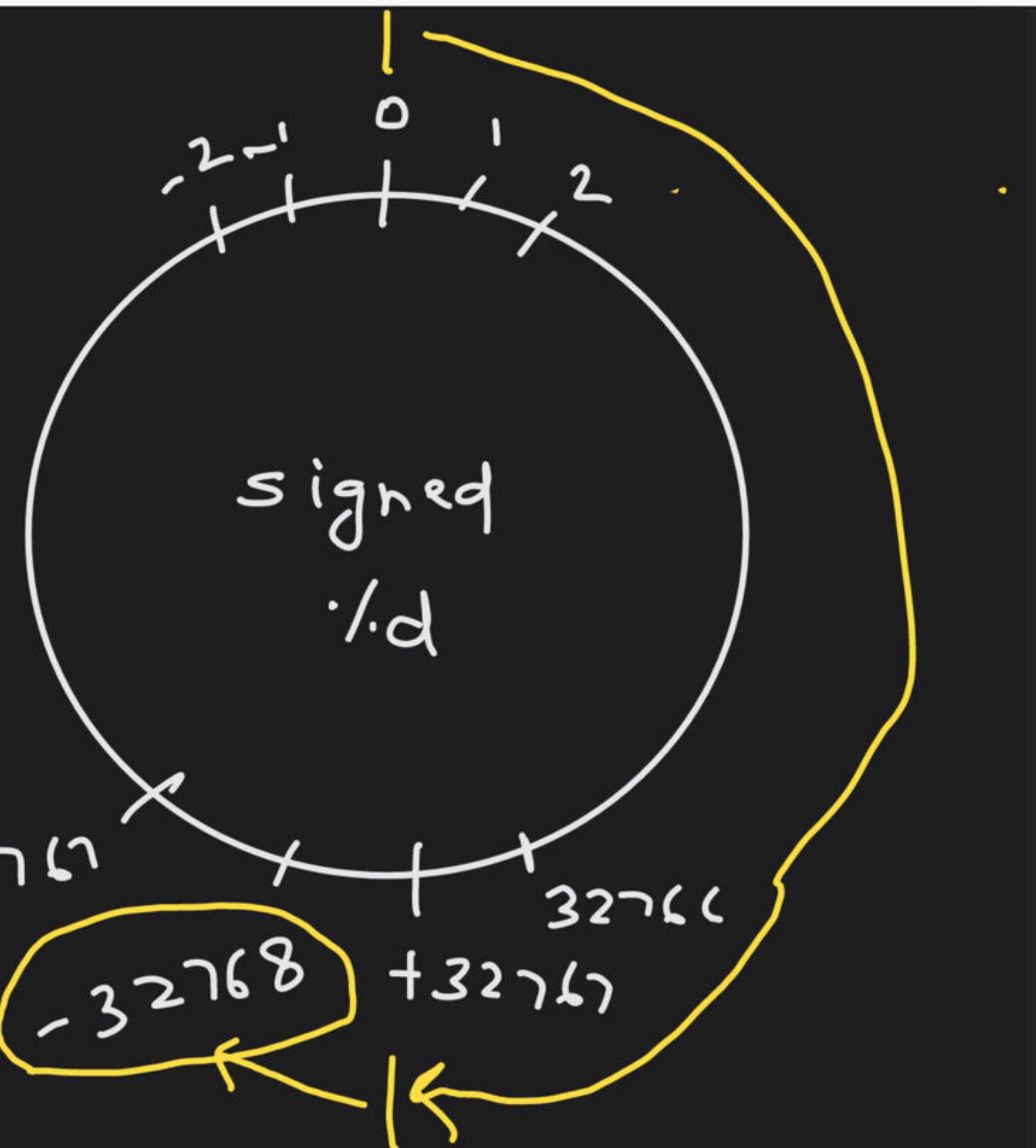
-1

65535



int  $\Rightarrow$  2 bytes

```
#include <stdio.h>
void main(){
    signed int a = -32768;
    printf("%d", a);
}
```



32768 is 1 more than

```
void main()
{
    int a = -32768;
```

```
printf("./d", a);
```

```
}
```

0/  
32768

a  
32768

int  $\Rightarrow$  2 byte



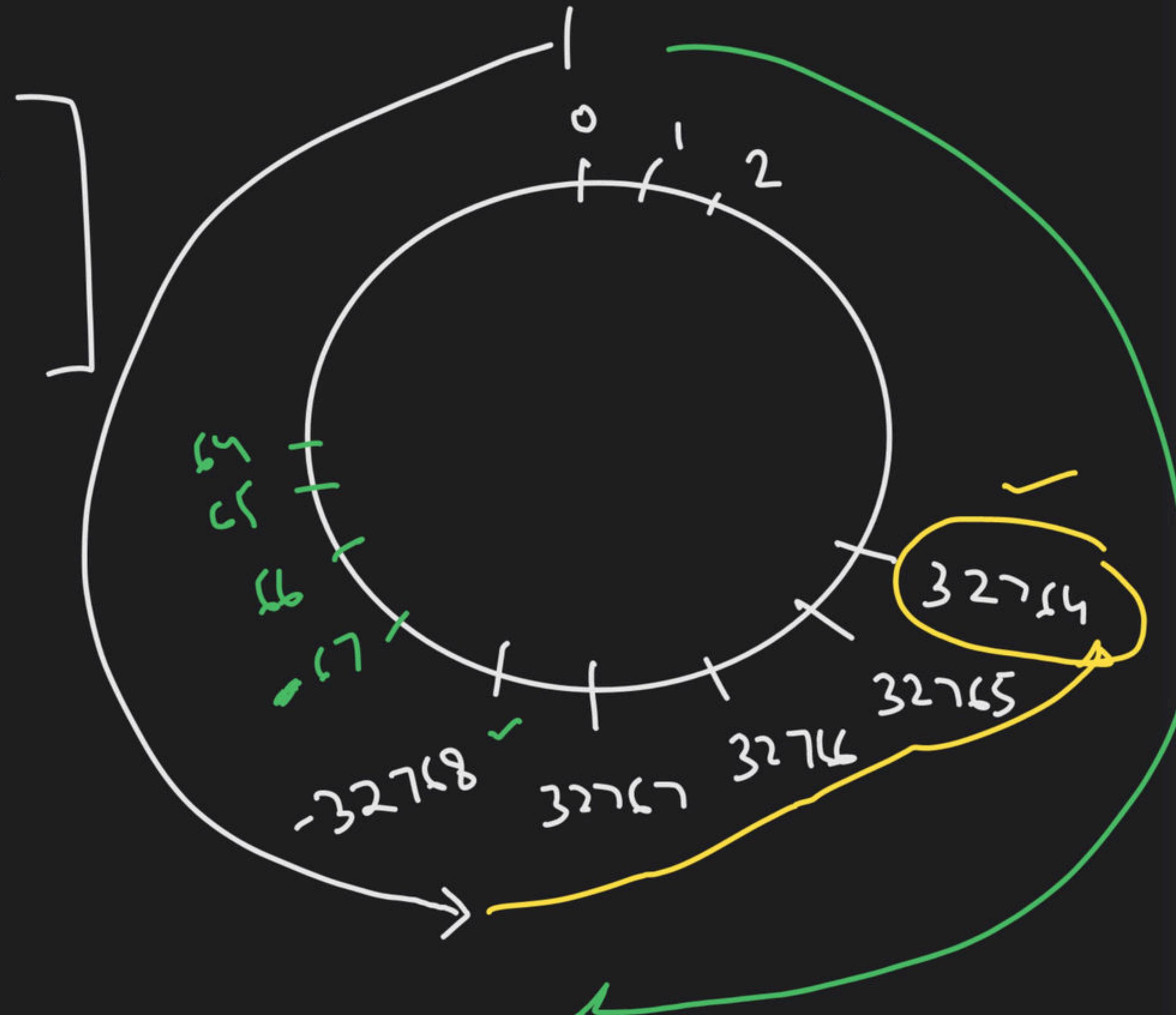
```
int a = -32772 ;
```

```
printf("%.1d",a);
```

+ 32764

```
int a = 32772
```

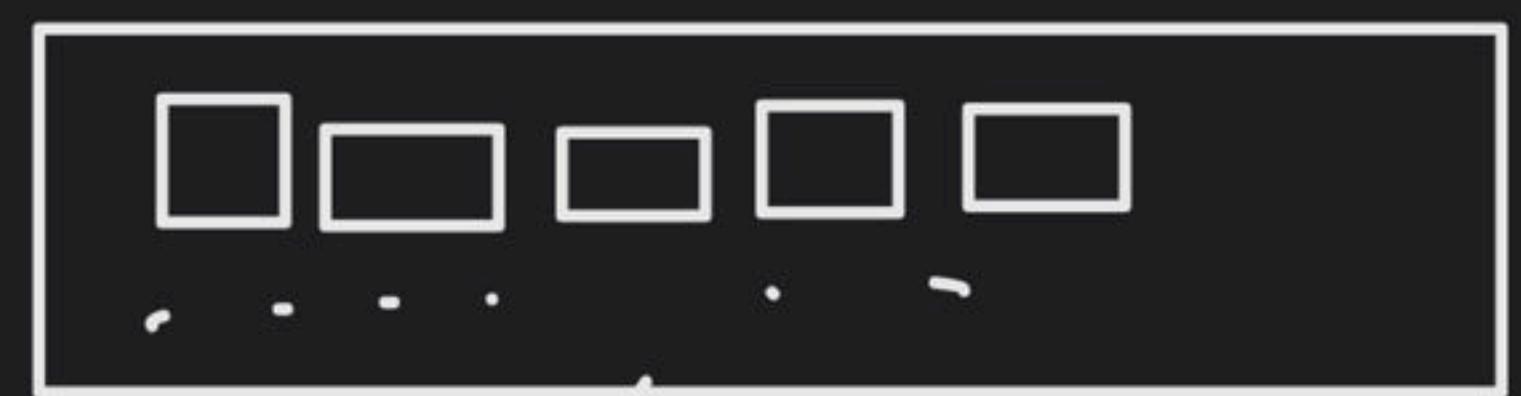
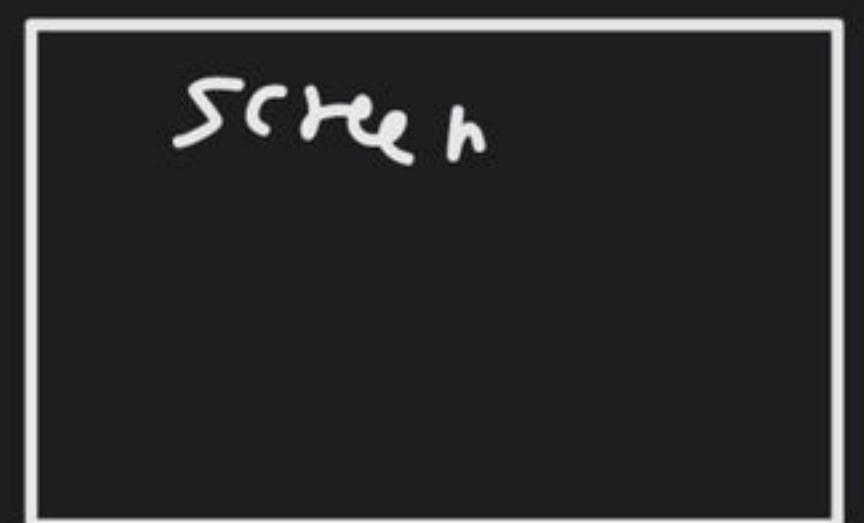
- 32764



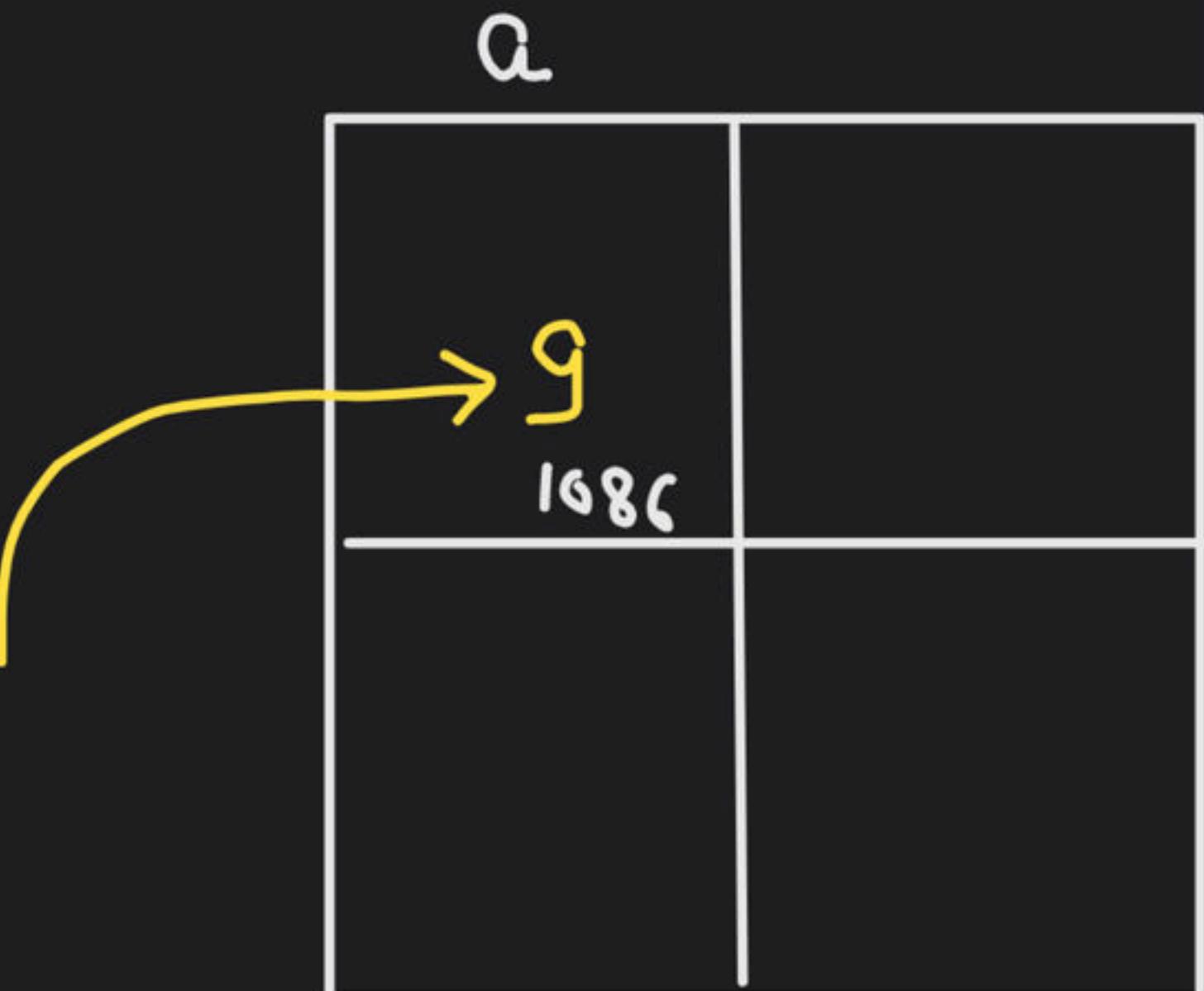
# Taking i/P from keyboard

---

scanf  
 ↳ To read i/P  
 from key  
 'int a;'  
 CPU



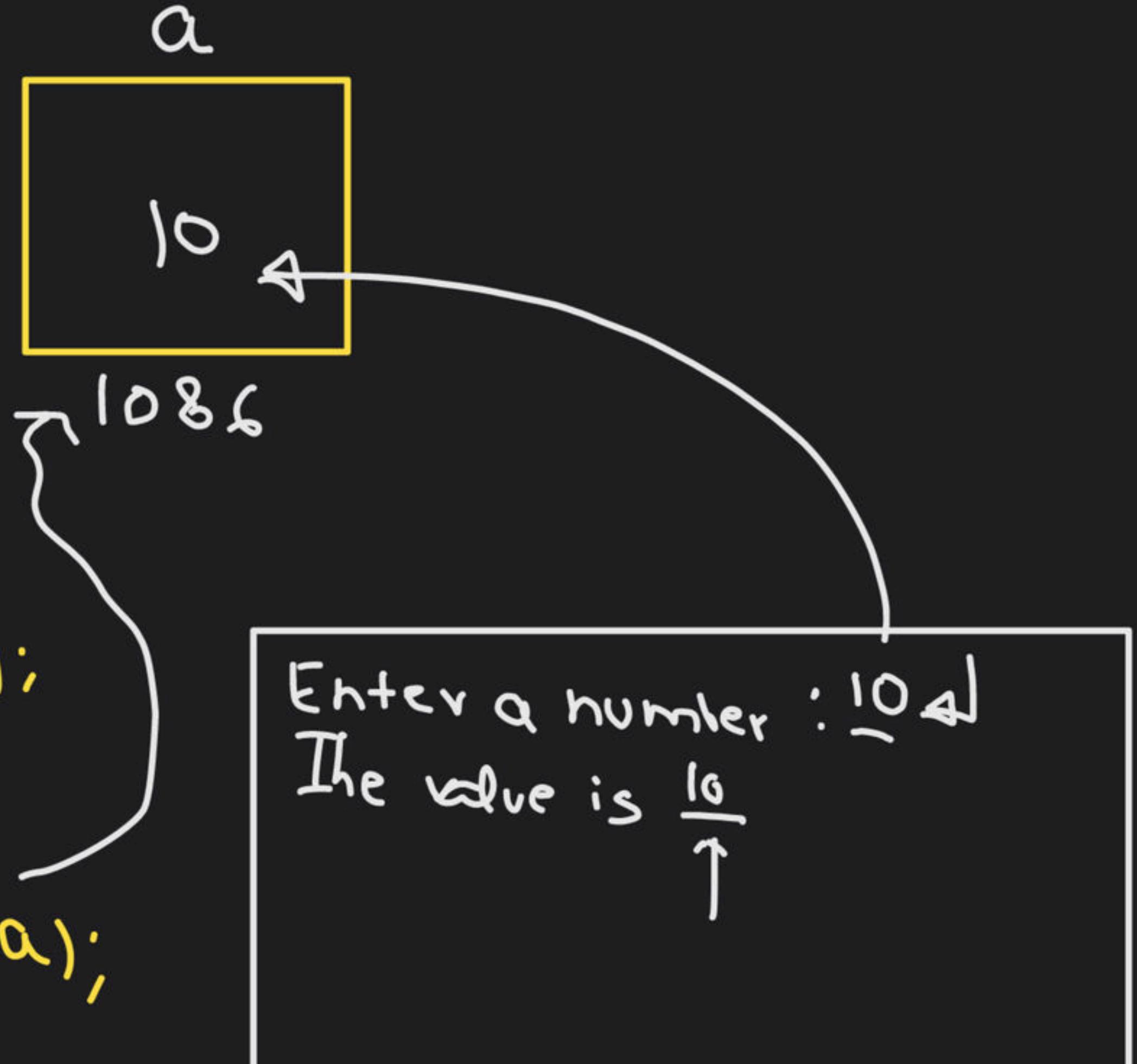
KB



scanf('%.d', <sup>Address</sup>&a );  
 ① format specifier

```
#include <stdio.h>
Void main() {
    int a;
```

- ① printf(" Enter a number : ");
  - ② scanf("%d", &a);
  - ③ printf(" The value is %d", a);
- }



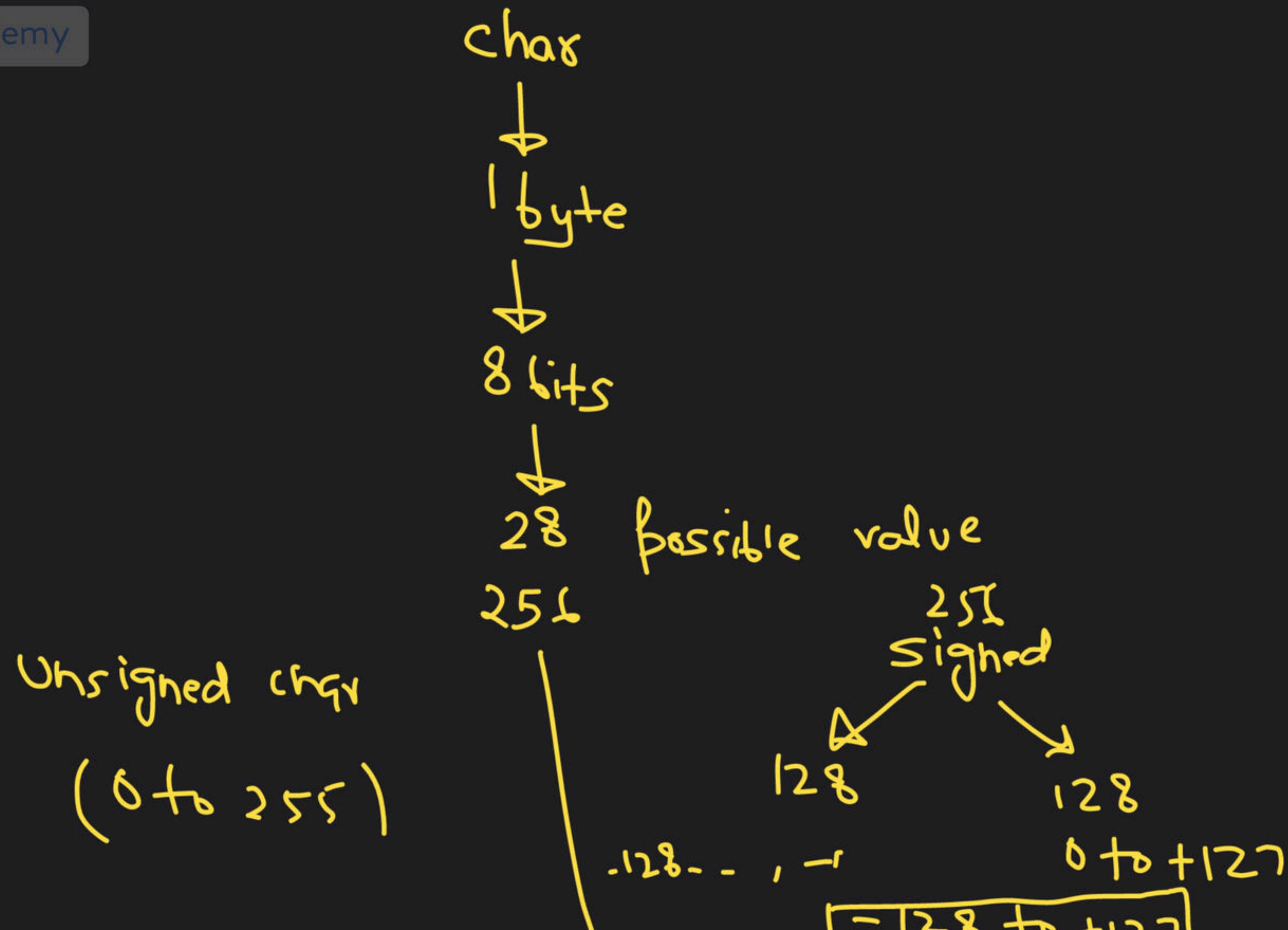
*Taking*

```
#include<stdio.h>
void main() {
    int a,b,sum;
    printf("Enter 2 numbers");
    scanf("%d %d", &a, &b);
    sum = a+b;
    printf("The sum of %d and %d is %d", a, b, sum);
}
```

2 no. from keyboard & print their sum

a → 10  
b → 20  
sum → 30

Enter 2 numbers|10 20|  
The sum of 10 and 20 is 30



ch9r

Range  $\Rightarrow$  Signed

-128 to +127

 $\Rightarrow$  Unsigned

0 to 255

Symbol store

## Character System

```
#include <stdio.h>  
void main()  
{  
    int a = 10;  
    ...  
}
```

10 → 00001010  
No.              No.  
decimal        binary

01000...  
01010...  
00001010  
000...  
...  
...

# Language

Set of  
Symbols

using a character  
system

Symbols in a  
lang. is rep.  
by a positive  
integer & constant.

## English type

@

info

## Character System

601 - - 10

info

American  
Standard  
Code for  
Information  
Interchange

## ASCII

∴ C → char. System

A - 65

a - 97

b - 48

@

B - 66

b - 98

c - 49

#

C - 67

c - 99

d - .

.

e - .

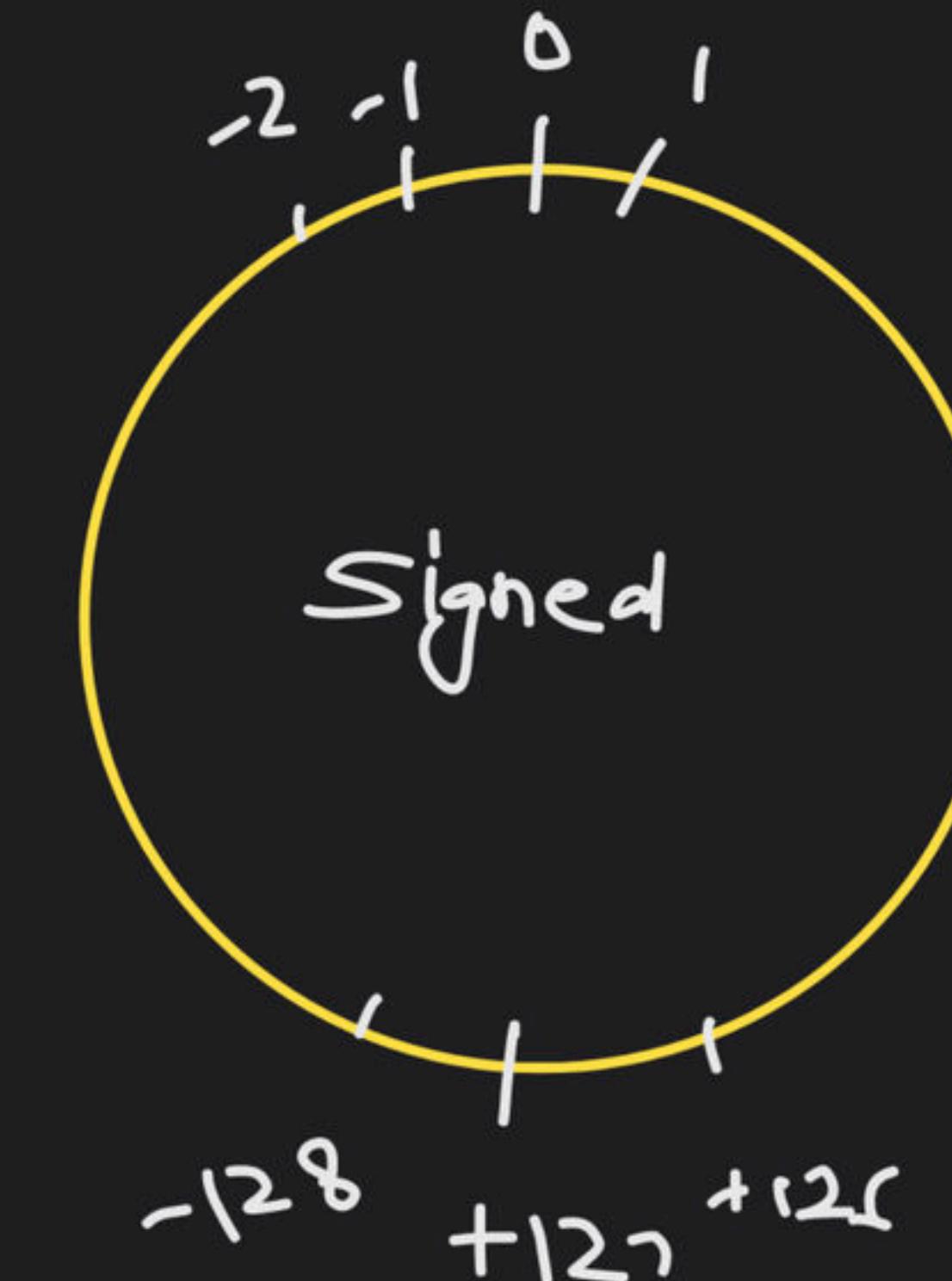
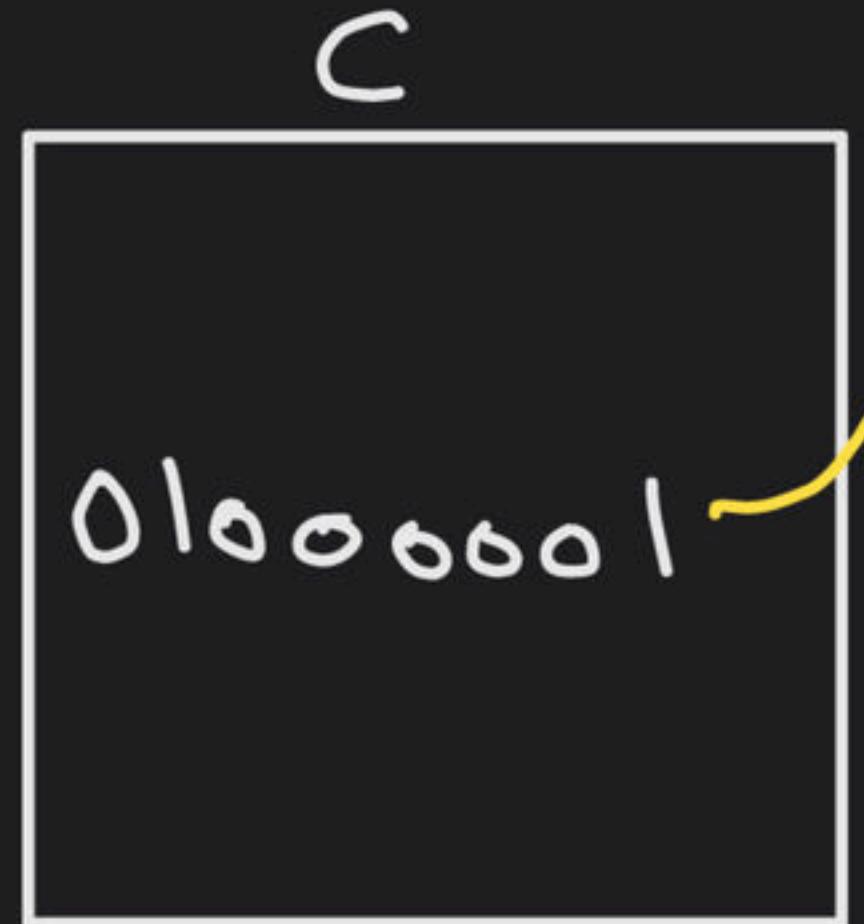
f - .

Z - 90

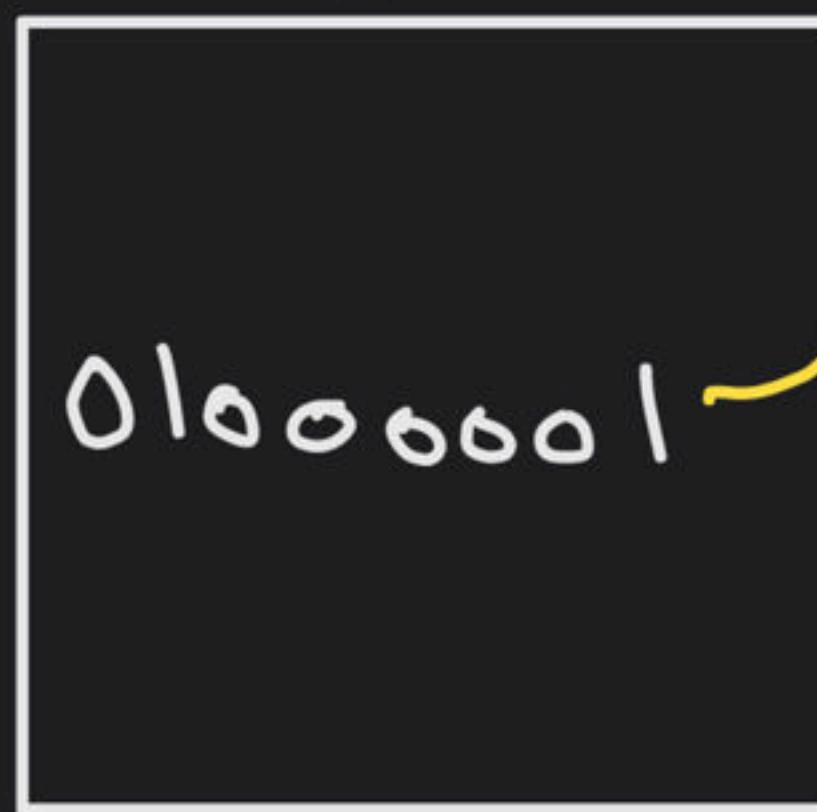
z - 122

```
char c = 65;
```

```
printf("./d", c);
```



```
char c = 65;  
printf("./c", c);
```



Character  
System

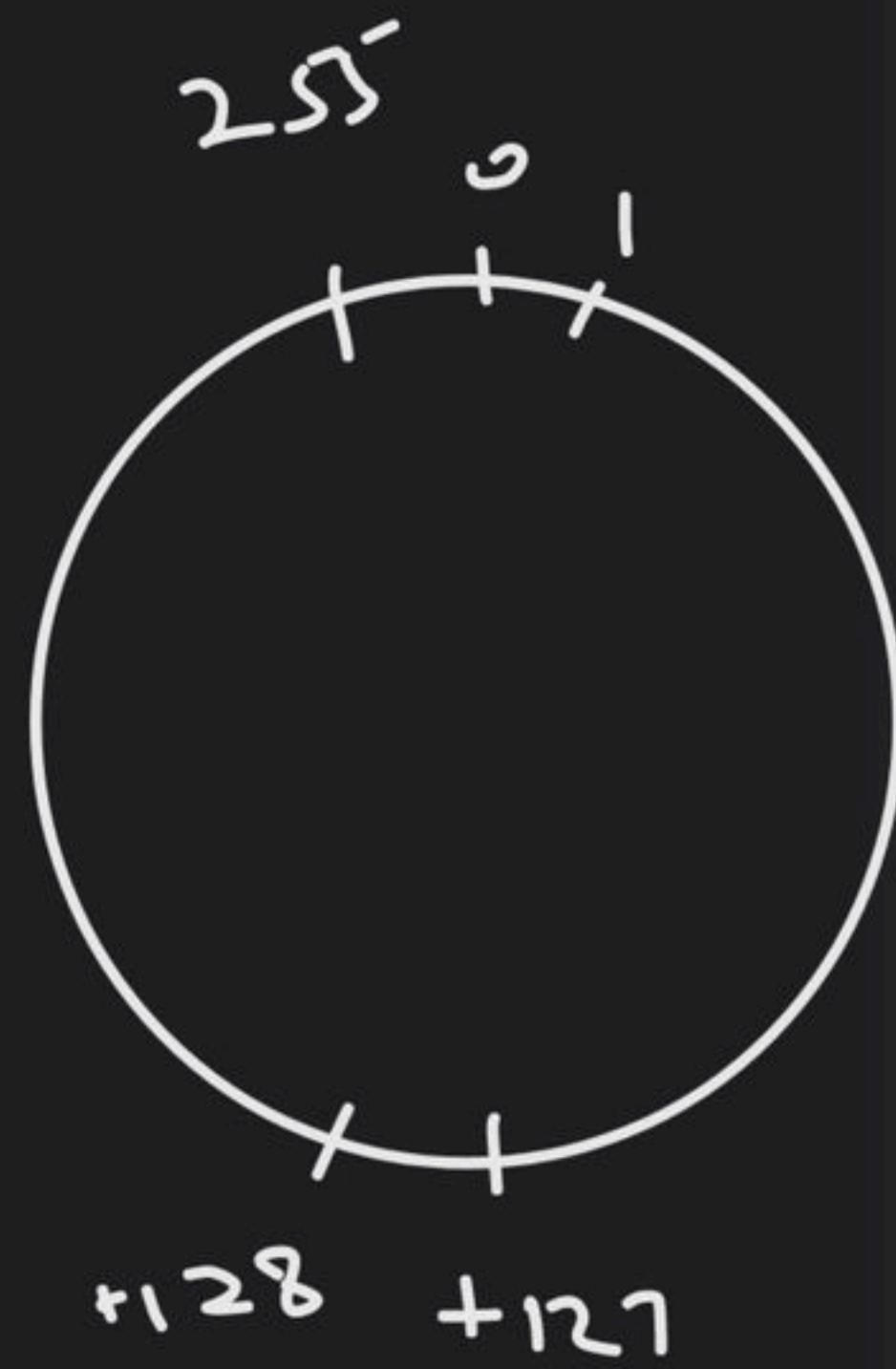
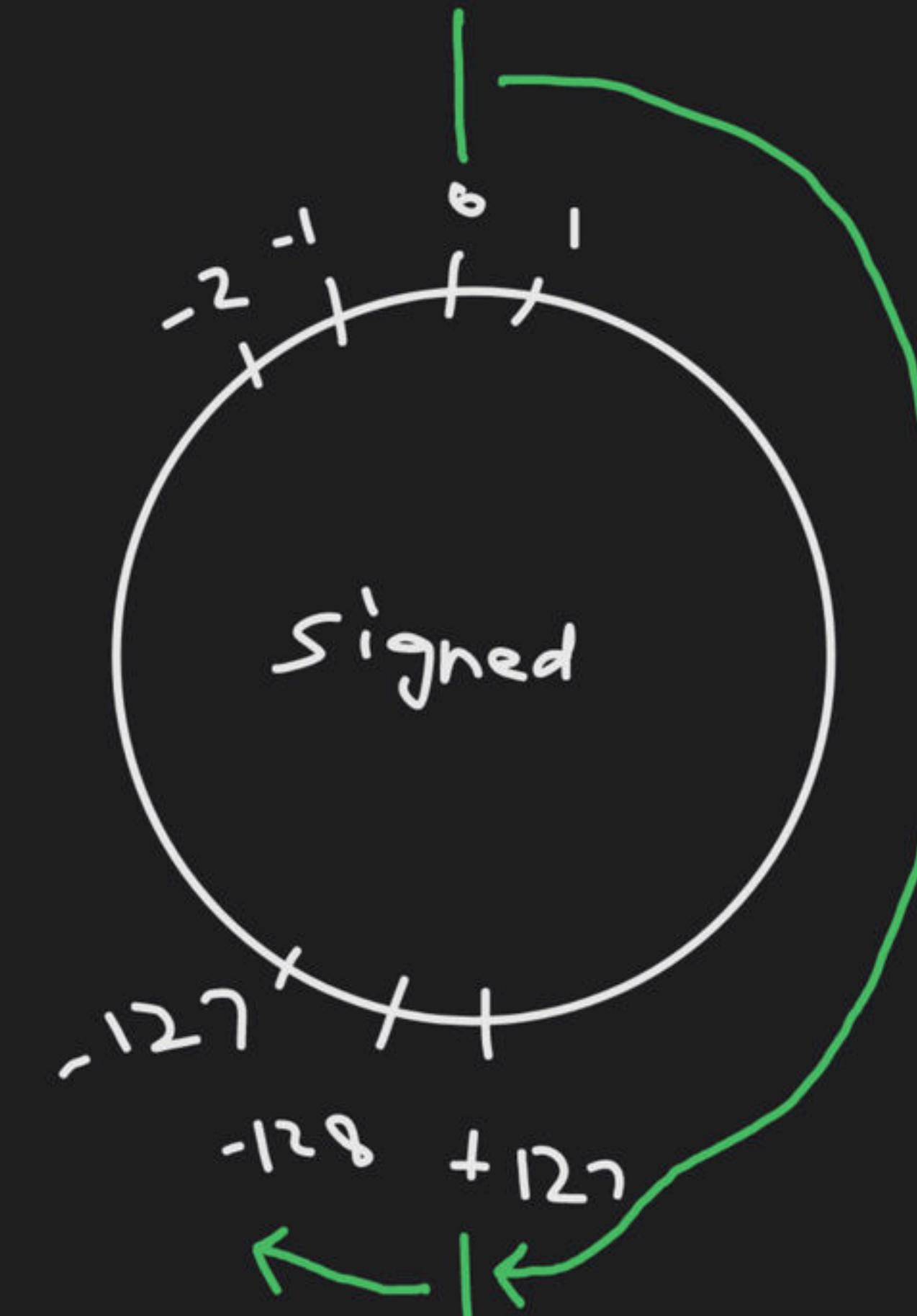


```
char c = 128;  
printf("%d",c);
```



-128

O/P: -128





```
char c = 132;
```

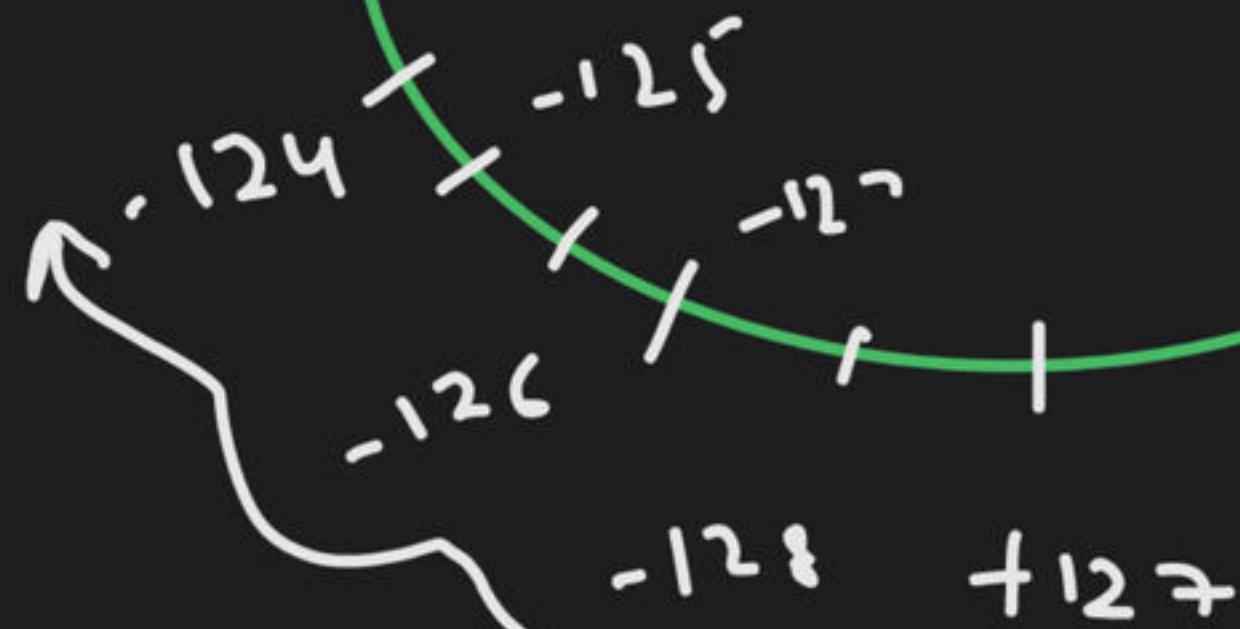
```
printf("%d", c);
```



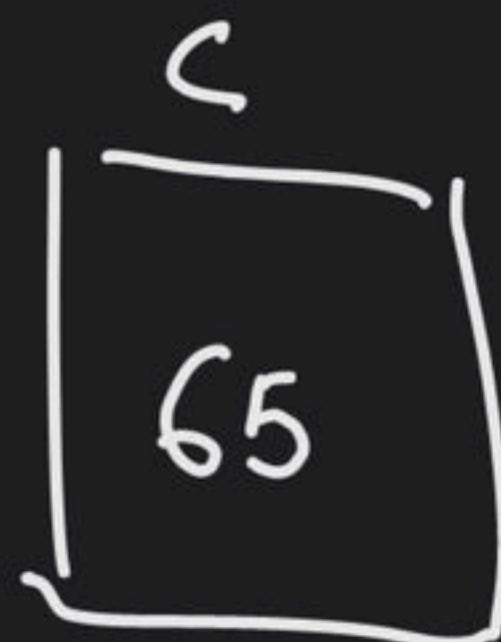
$$= -124$$

Total values  $\Rightarrow 256$

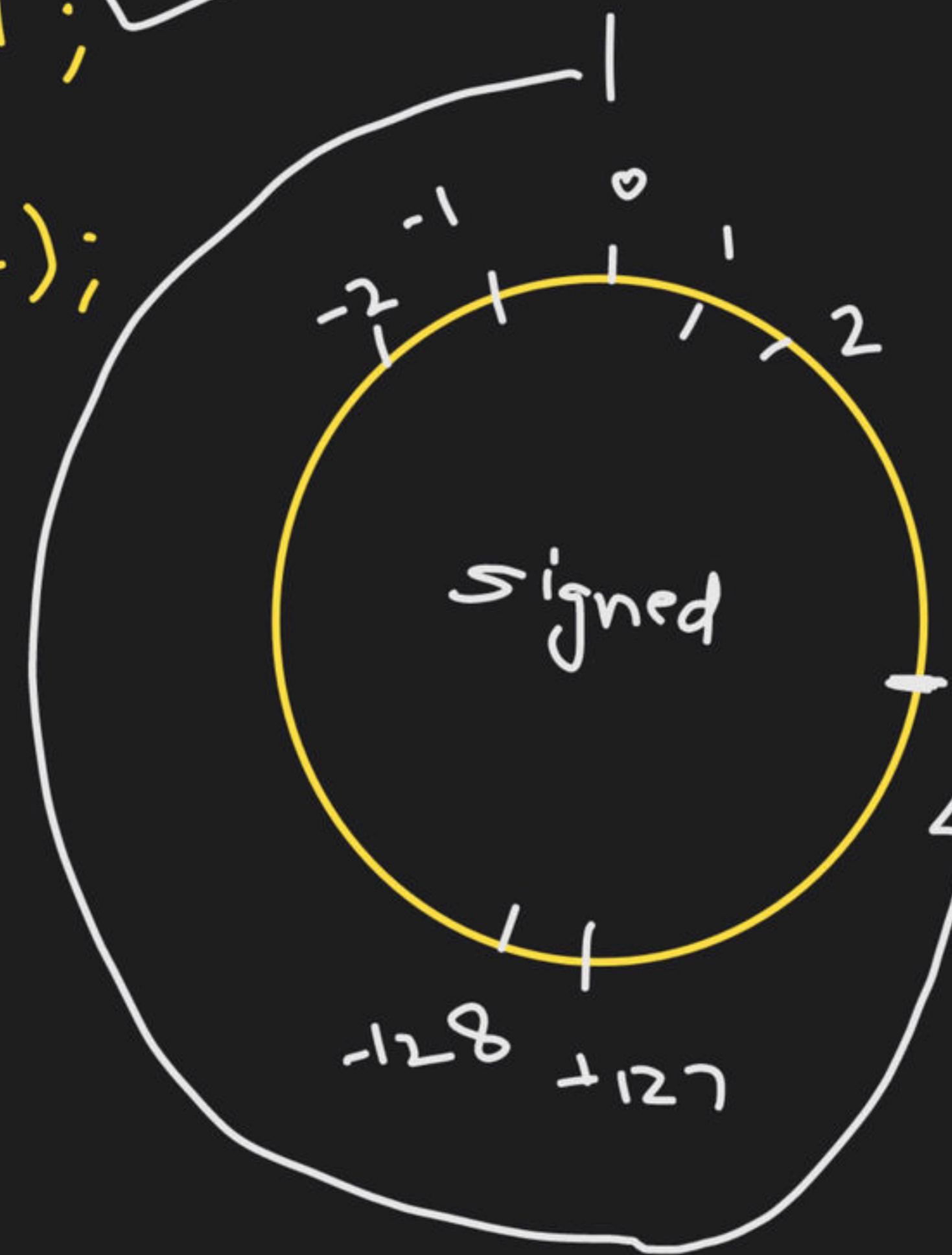
signed



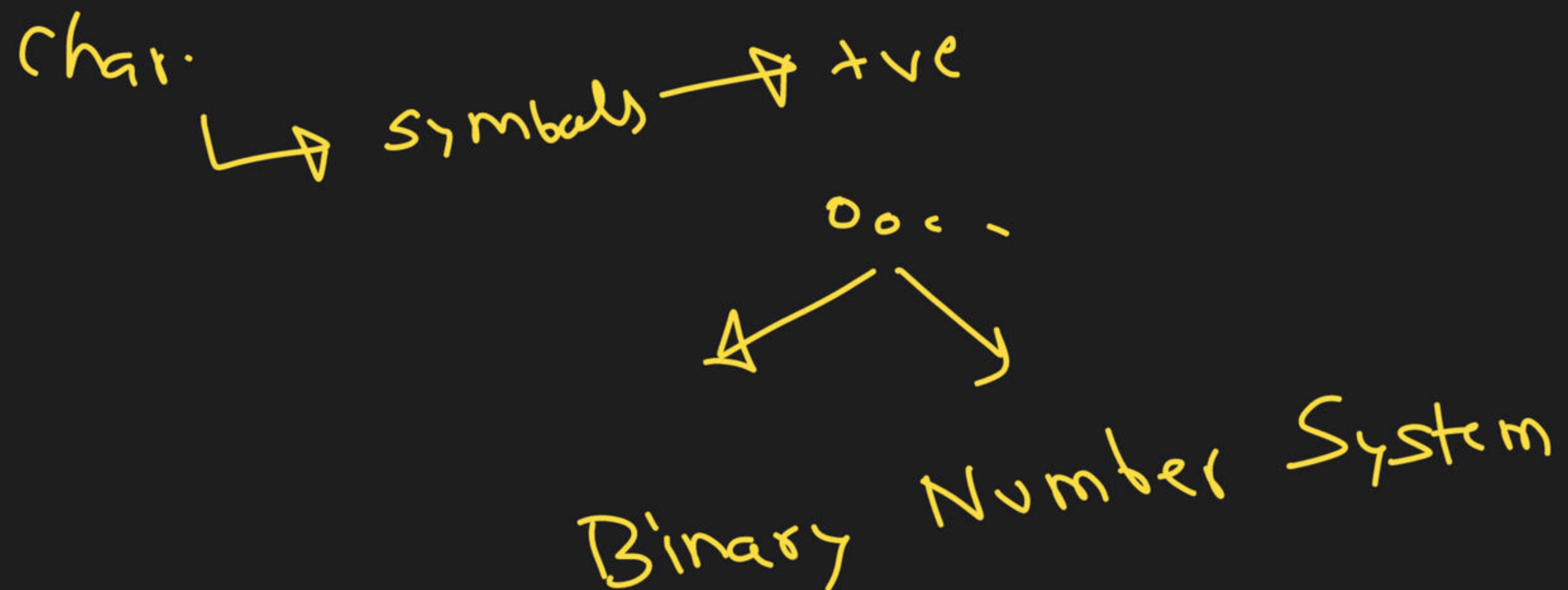
char c = -191; ✓  
printf("./c", c);



∴ C → char. System ⇒



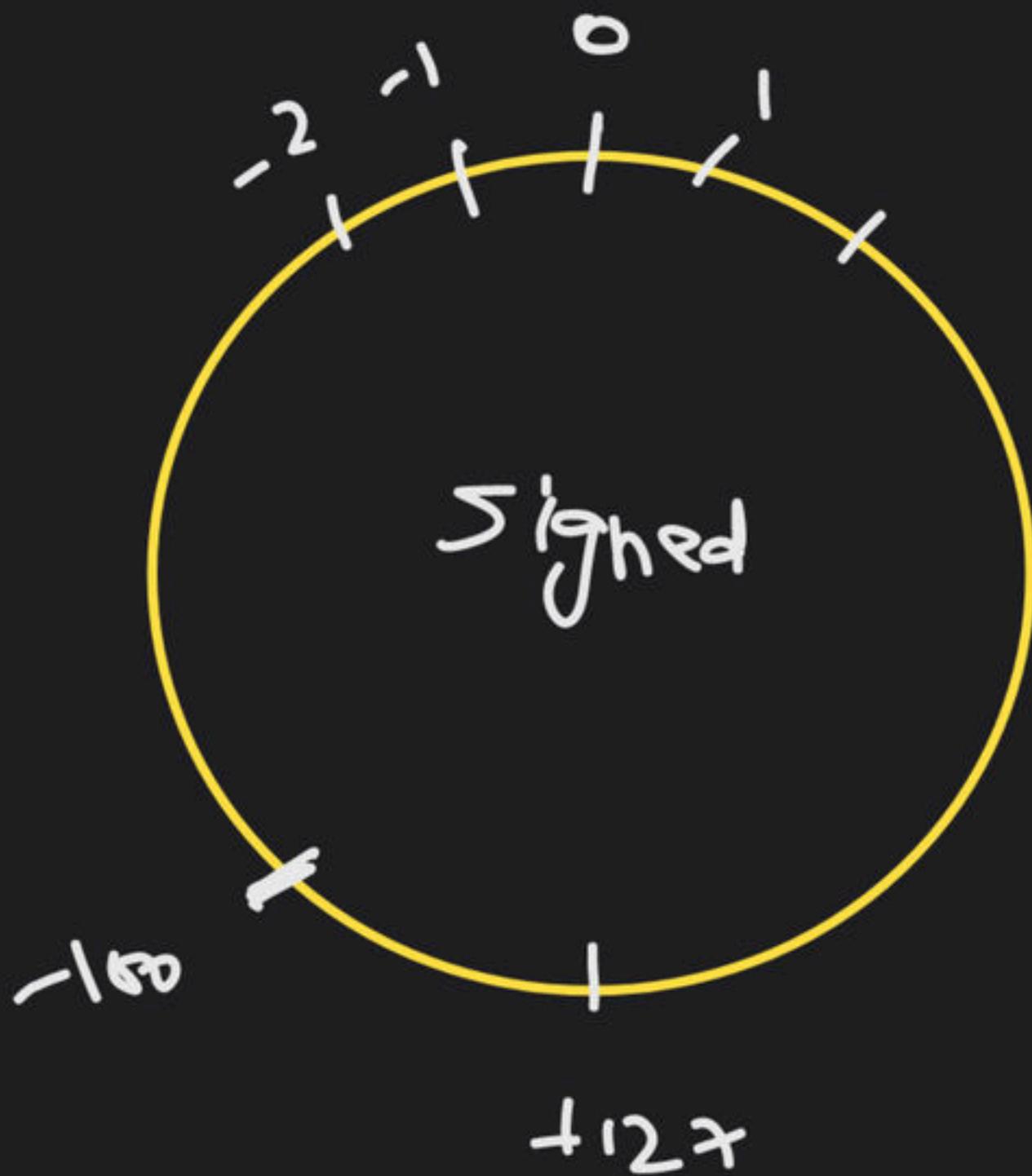
String



c  
-100

char c = -100

printf("-/.c", c);



char . System

unsigned

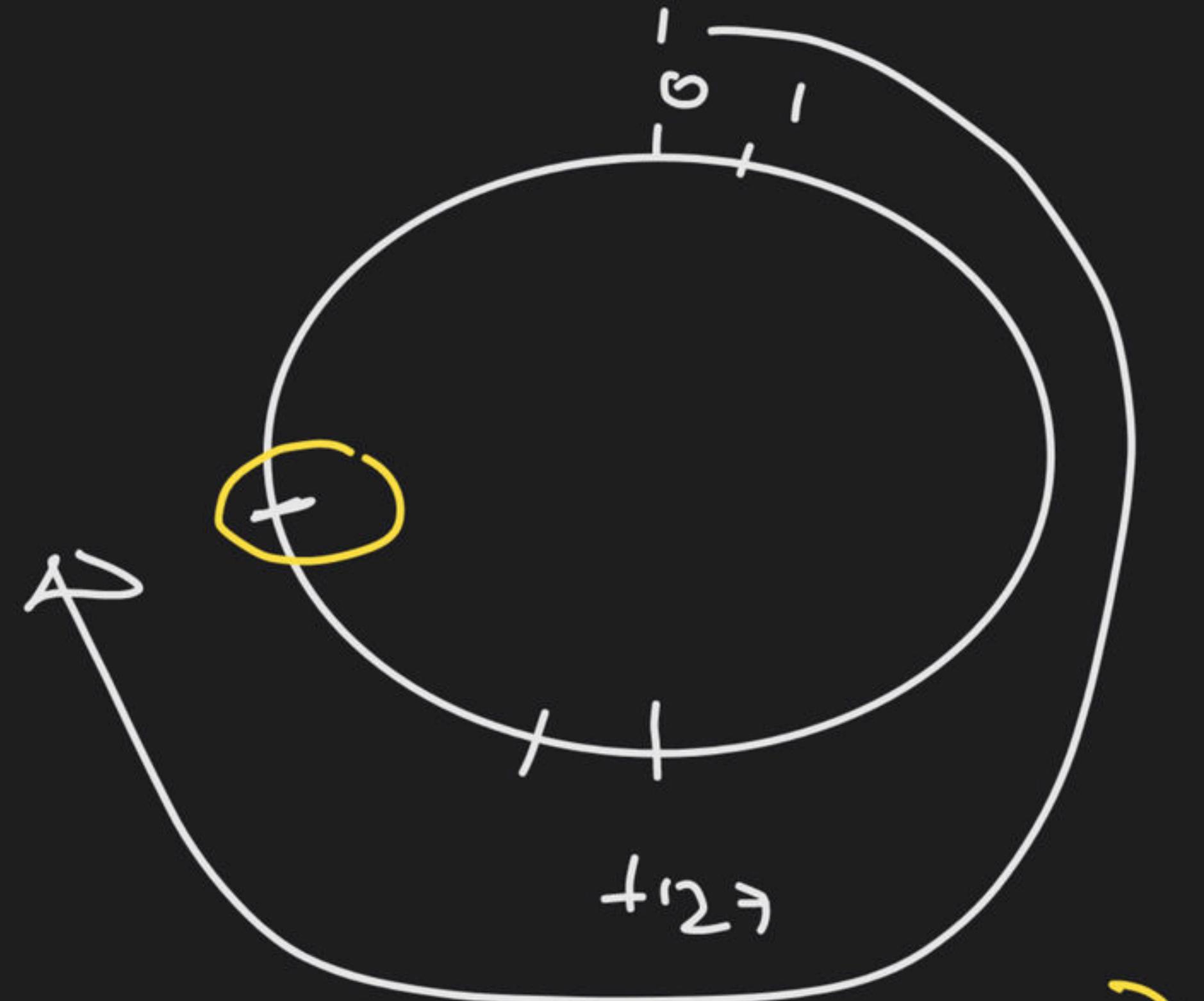
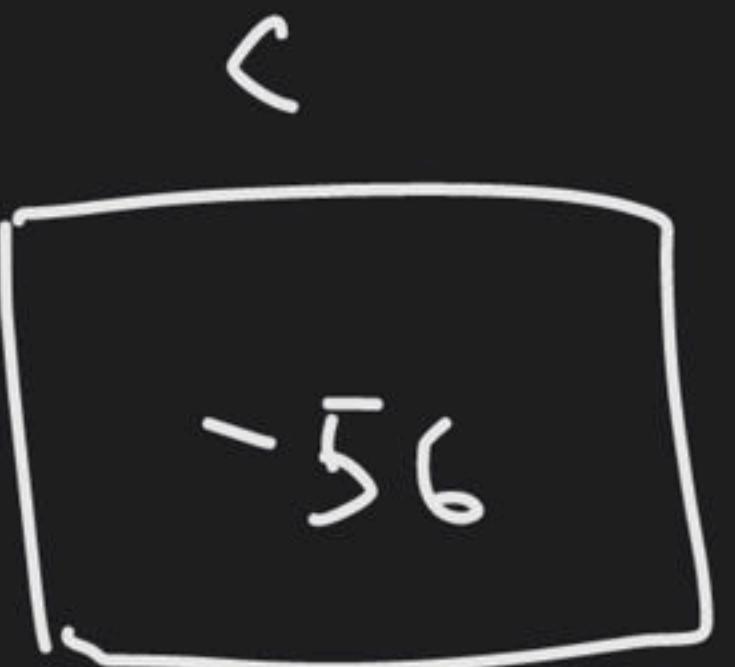
-100  $\Rightarrow$  152

This symbol  
Ka  
Ascii  
code  
152  
 $\downarrow$   
↙

./c → char. System  
( unsigned )

```
char c = 200;
```

```
printf("./c", c);
```



recording →

char  $c = 257$

-128 → Unsigned  
value  
 $+128$



## Escape Sequence

- ① \n : Move the cursor from its current position to the beginning of next line.

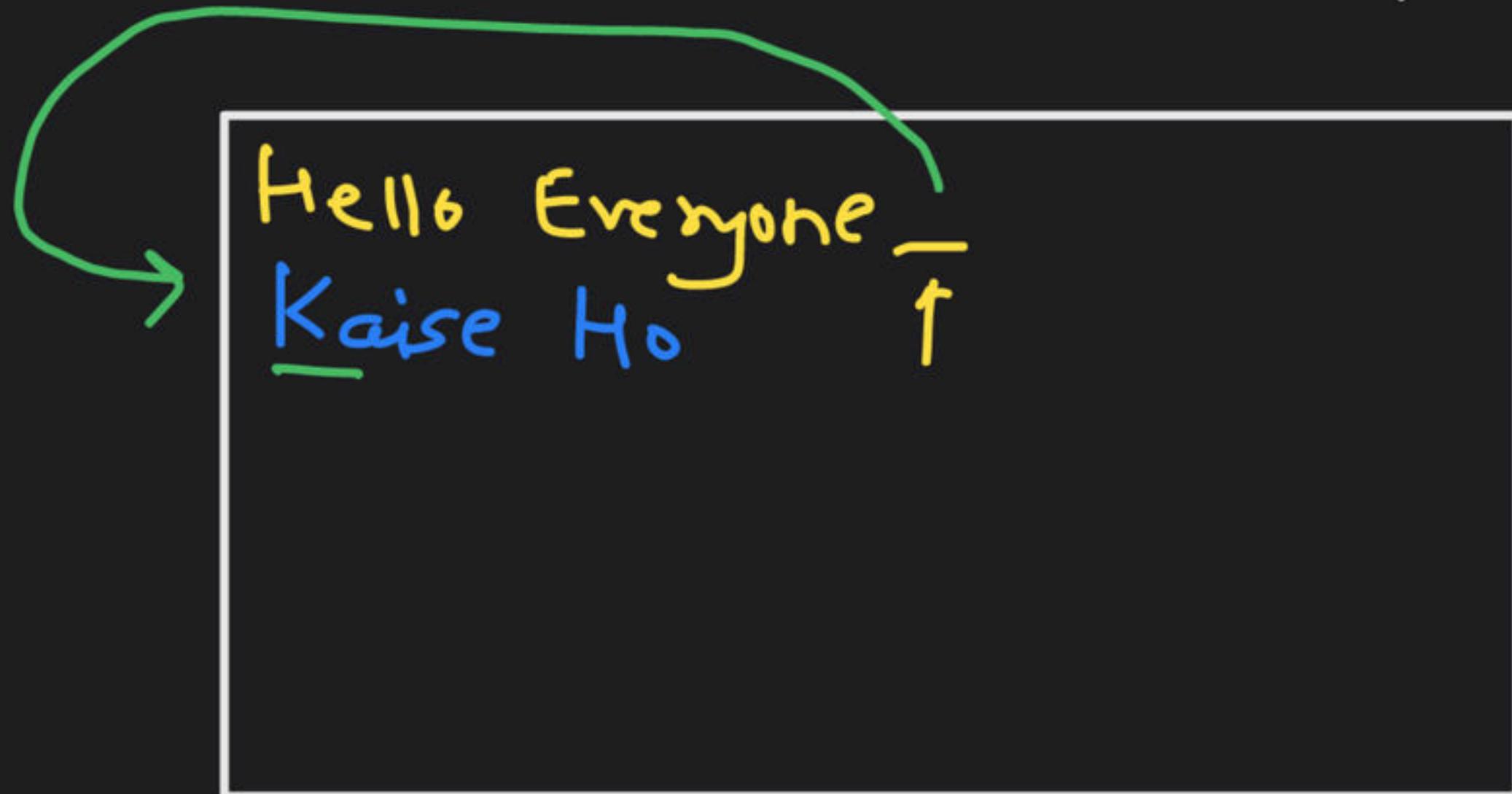
```
printf("Hello Everyone"); ✓  
printf("Kaise Ho");
```

Hello Everyone Kaise Ho  
↑

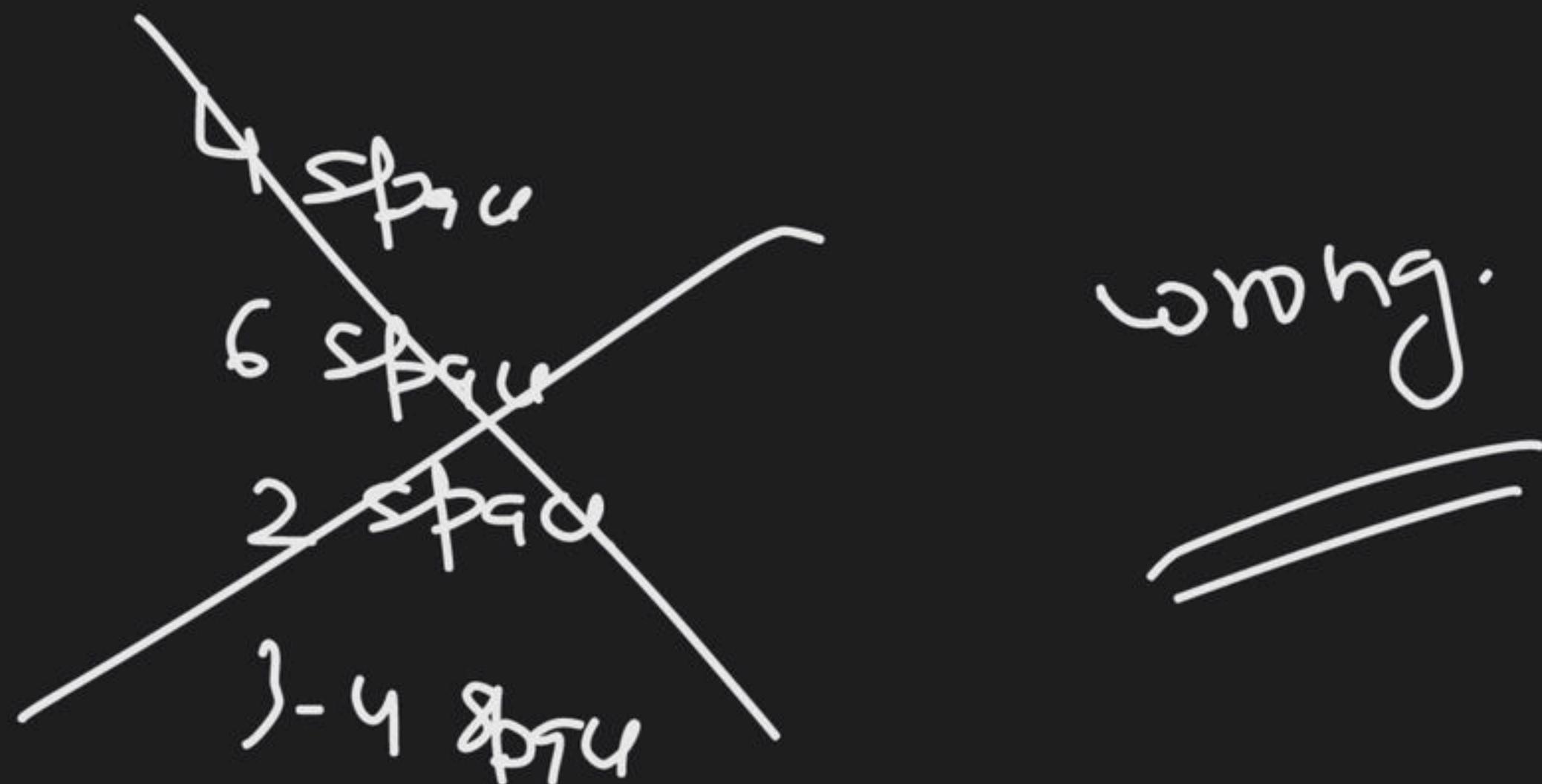
```
printf("Hello Everyone\n");  
printf("Kaise Ho"); ✓
```

Hello Everyone  
Kaise Ho  
↑

✓ `printf("Hello Everyone");`  
✓ `printf("\nKaise Ho");`



$\lambda t \rightarrow \bar{t}ab$

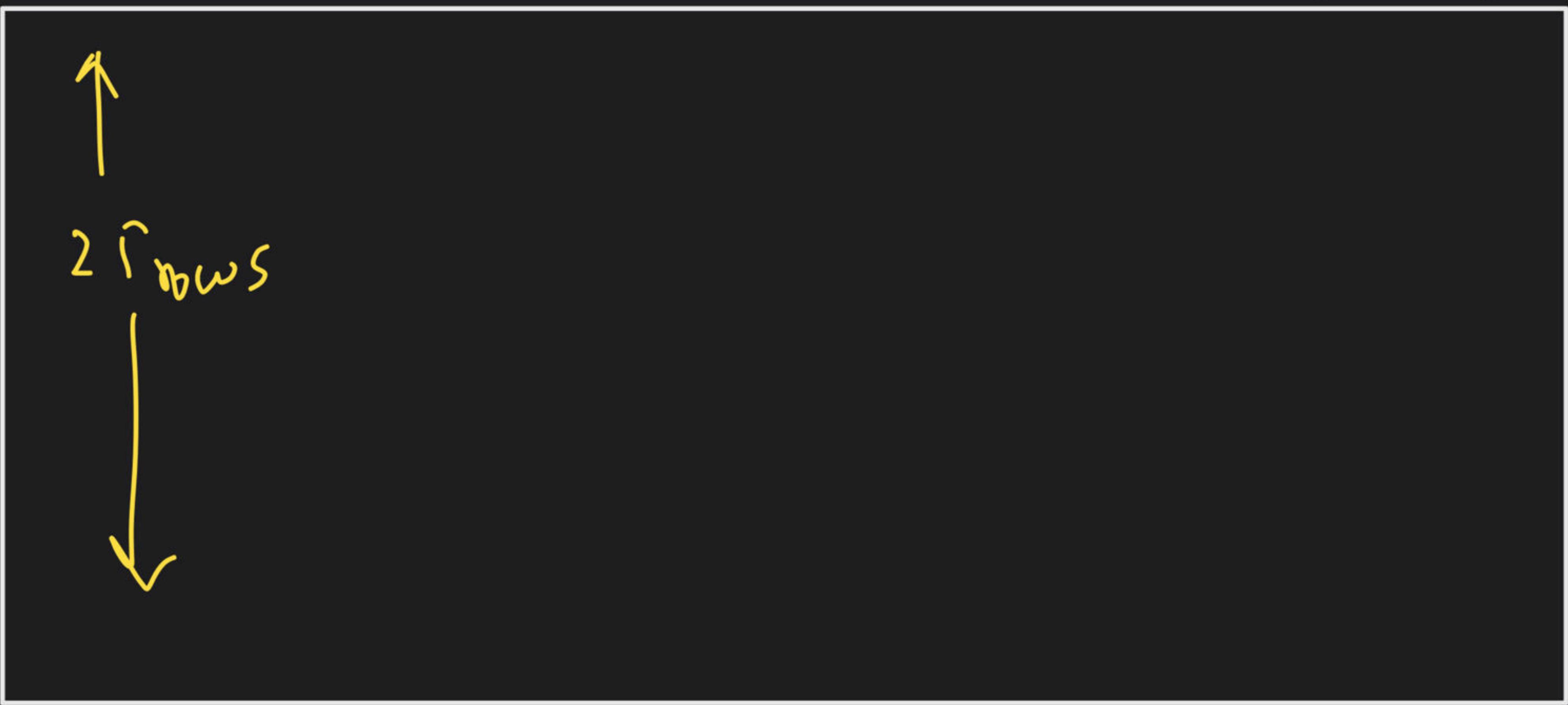


wrong.

. 0001 -/-

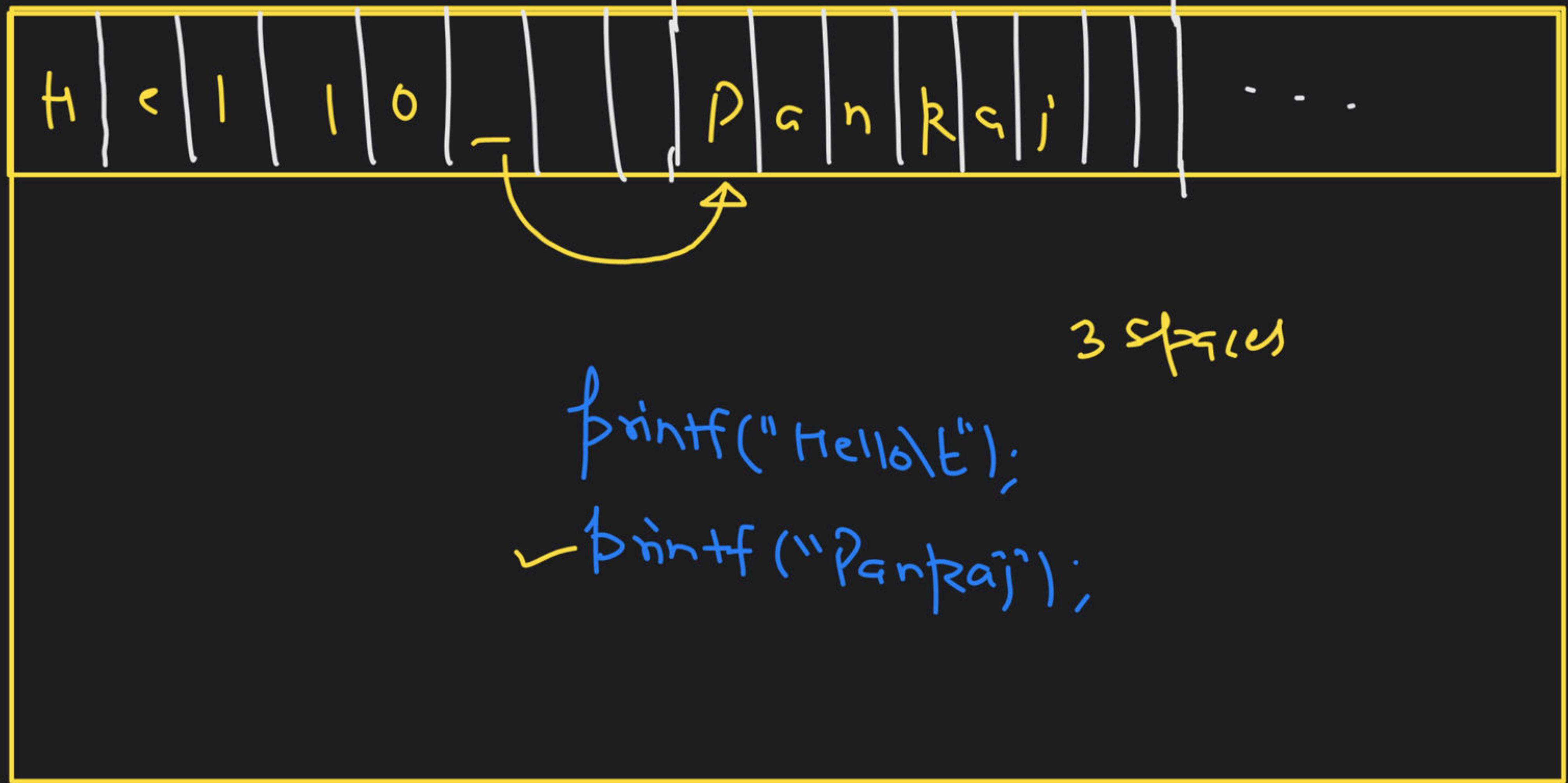


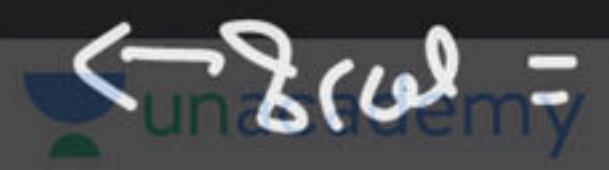
86 calcs



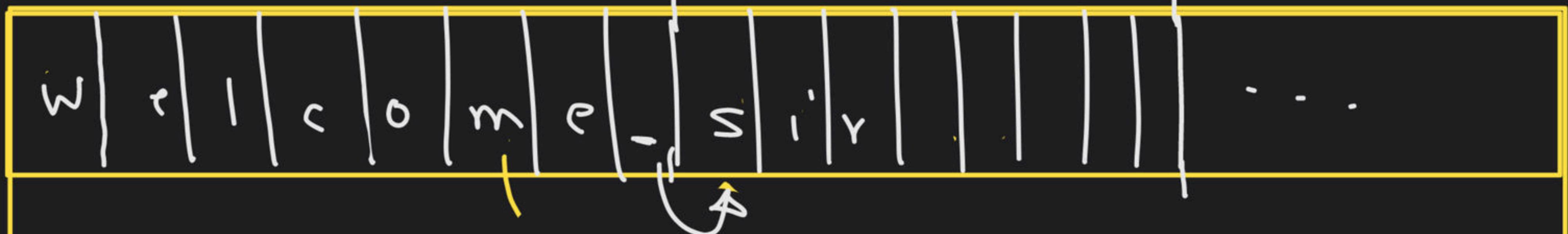


\t → Move the cursor to next available frame.

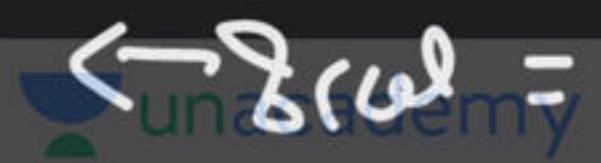




$\leftarrow$  8 col = 1 frame  $\longrightarrow$  |  $\leftarrow$  2nd frame



```
printf("Welcome\\t");           | space  
printf("sir");
```

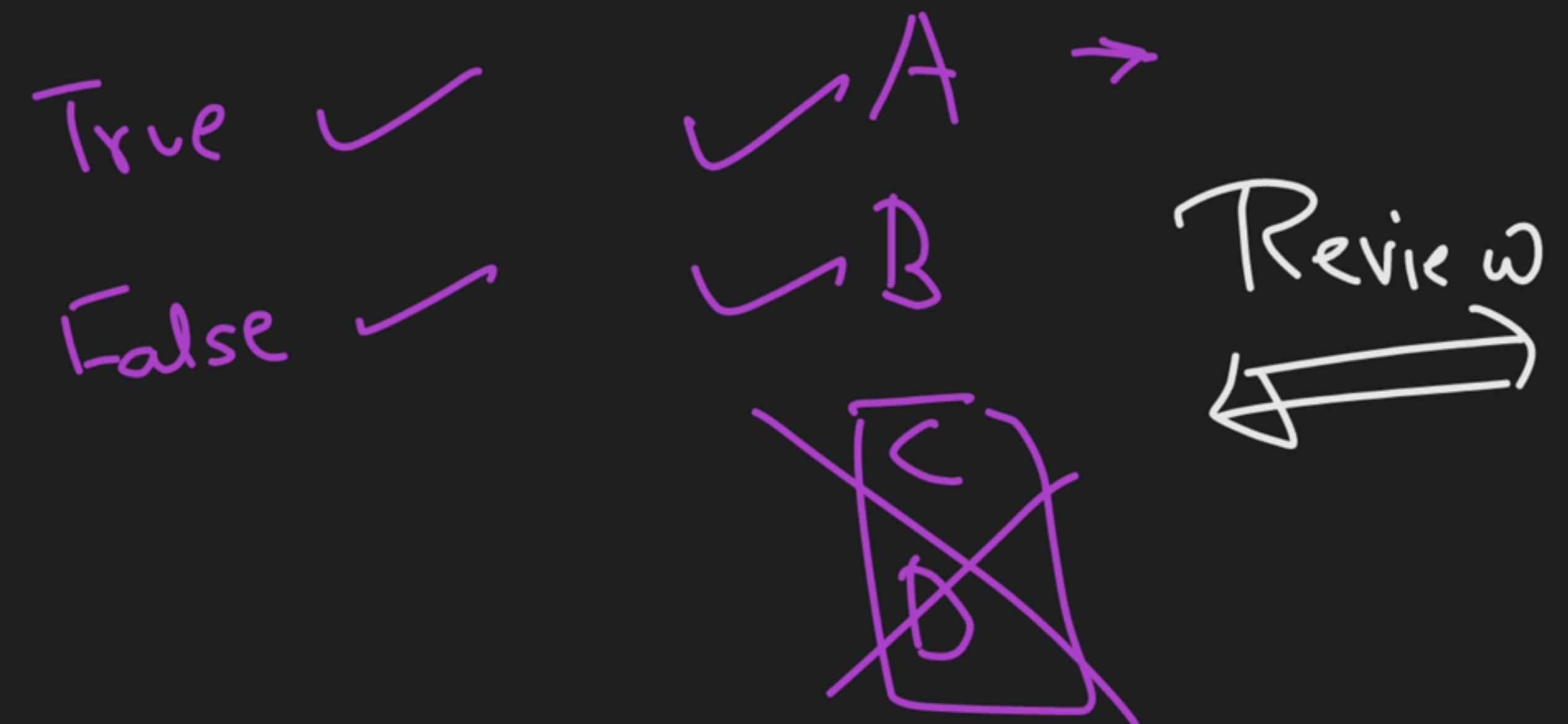


← 8 col = 1 frame → ← 2nd frame

P | a | n | k | a | j | J | i | - | R | a | w | n | h

printf("PankajJi\\t");  
printf("Rawan");

8 spaces





# THANK YOU!

Here's to a cracking journey ahead!