



Char Arrays & Strings - Class 1

Special class

→ char array

array → sequence of characters

creation → char arr[10];



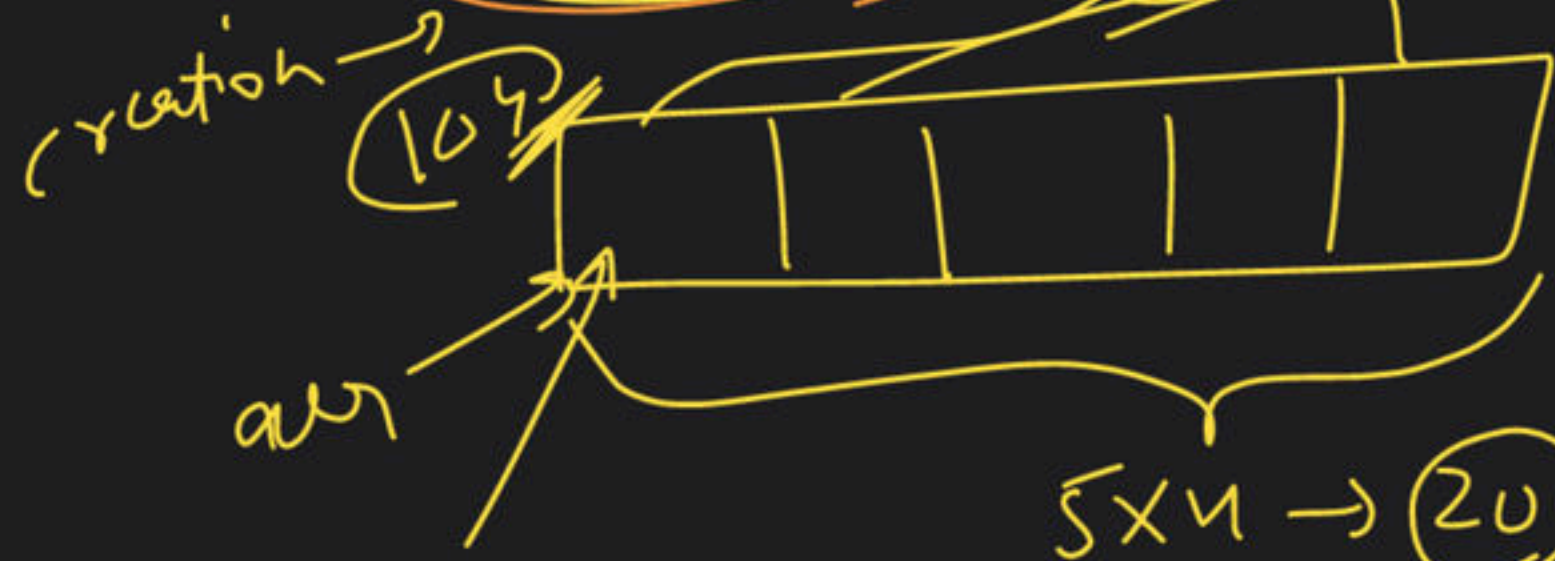
10 byt

Vector<char> arr

cout << arr → 107

int arr[5]

cin >> arr



arr[index]

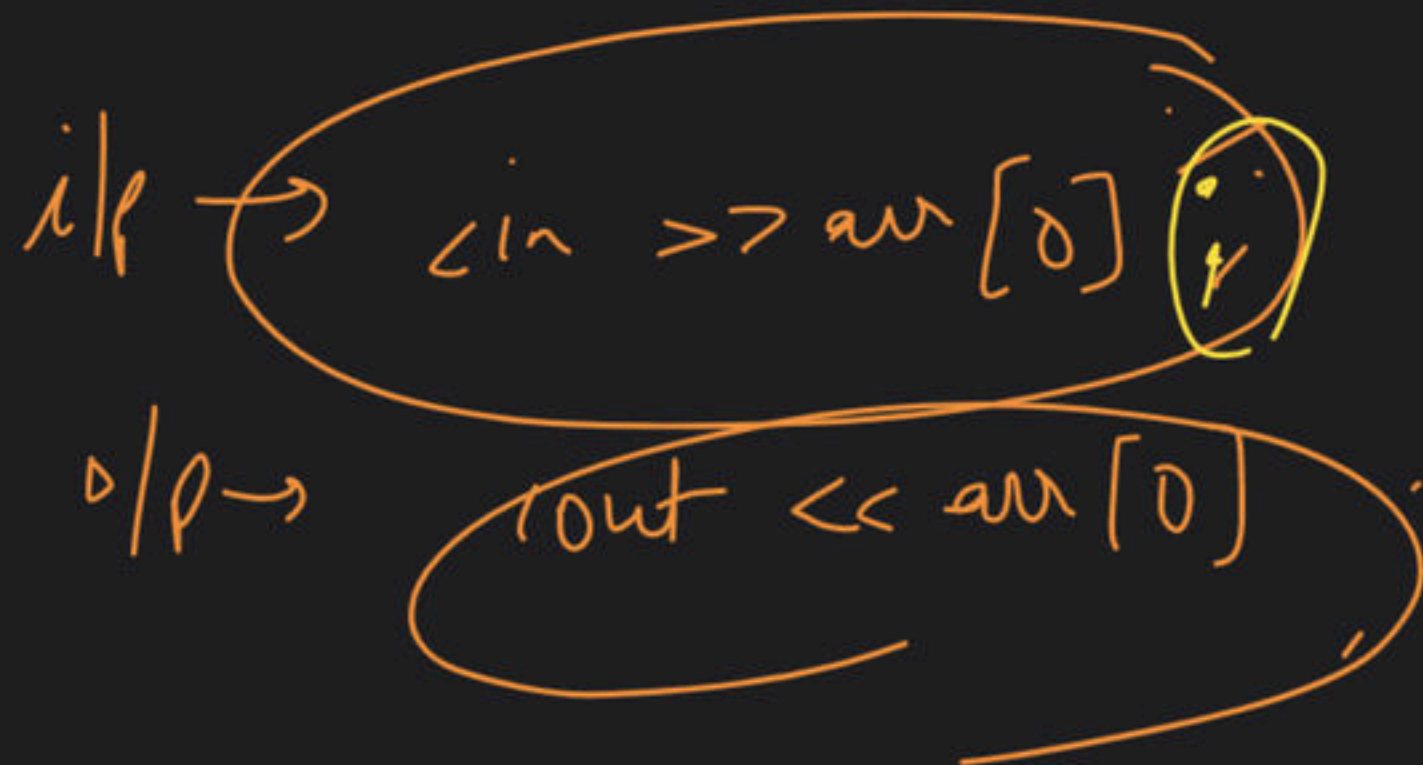
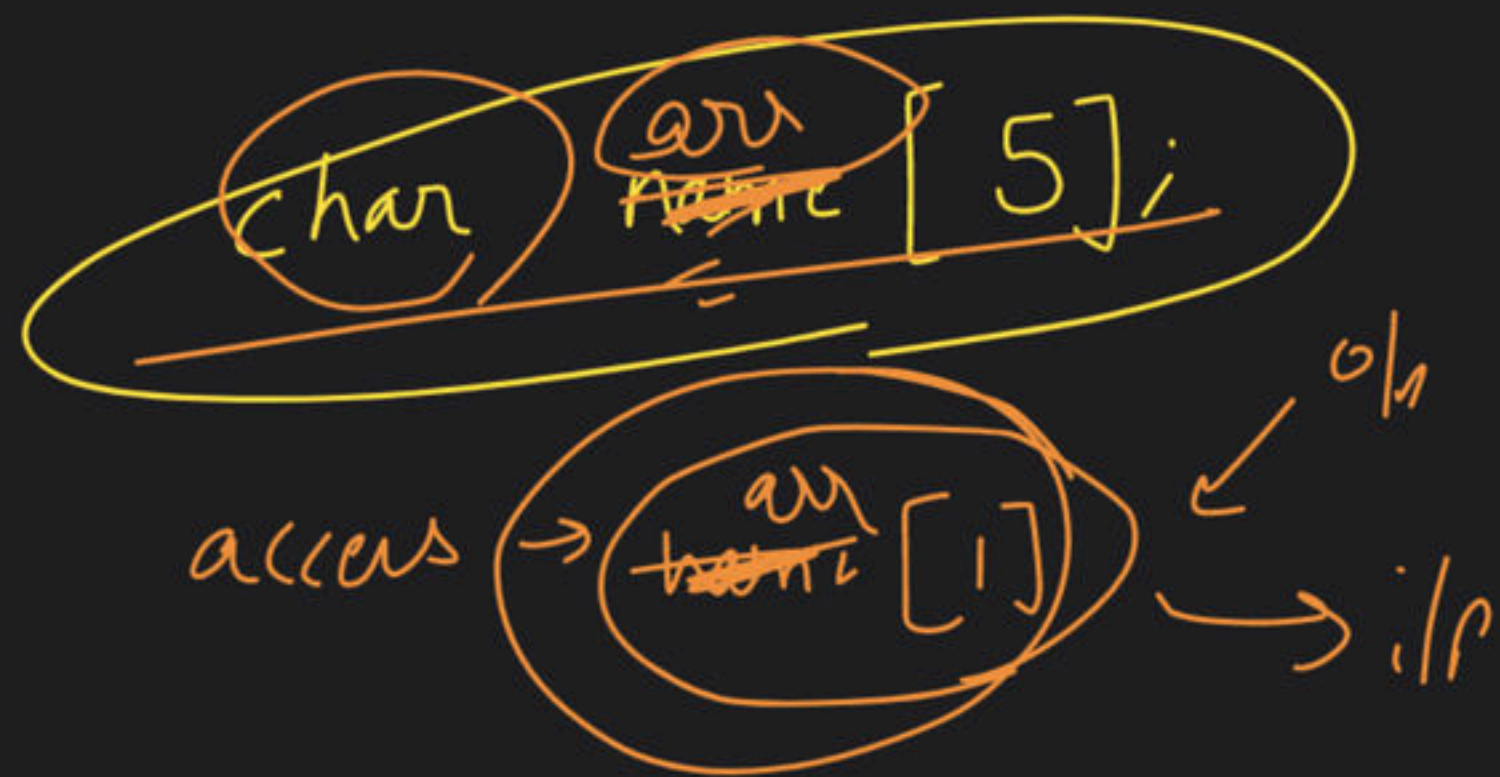
0, 1, 2, ...

i/p → cin >> arr[i] or loop

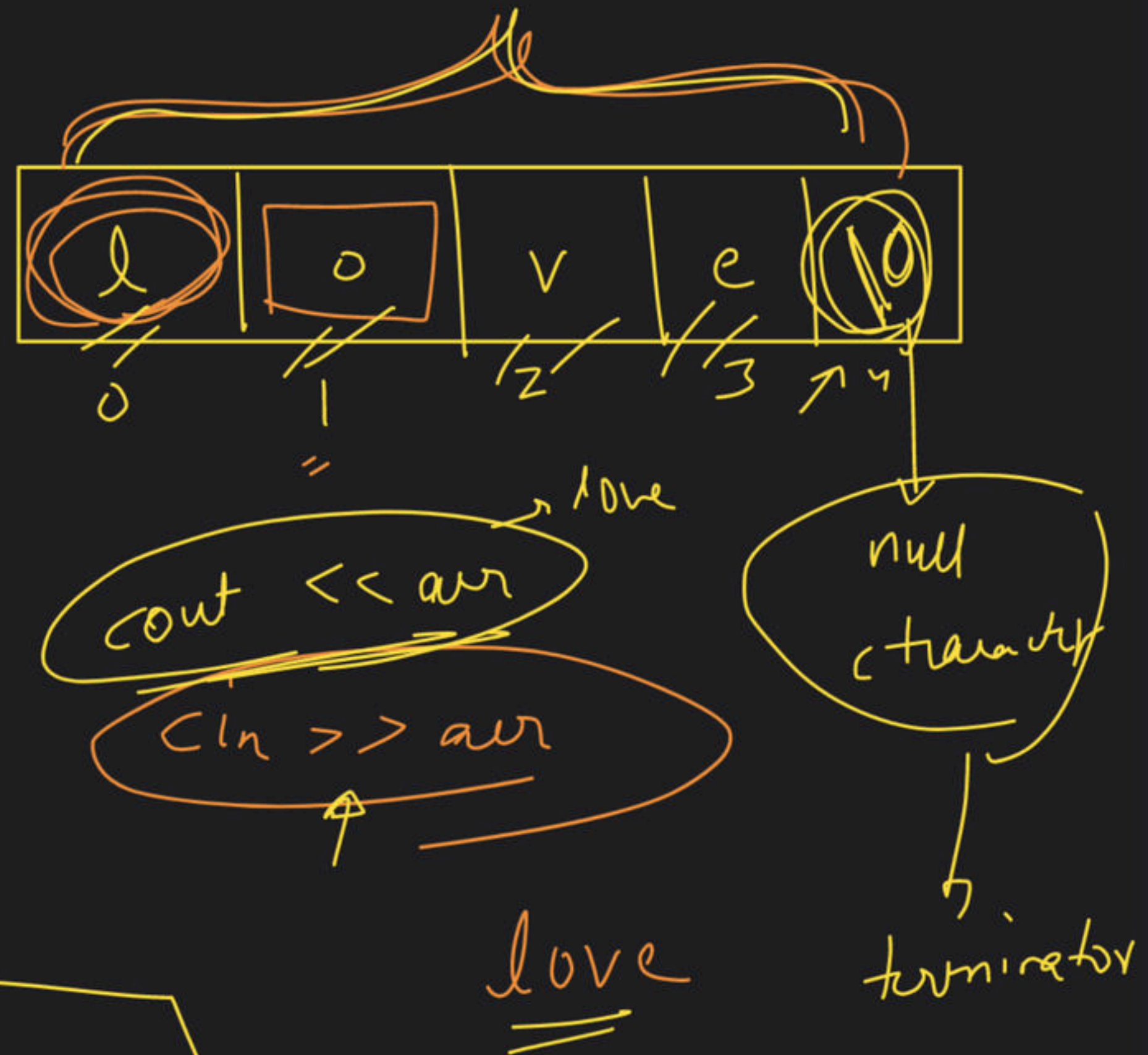
single block

all block

o/p → loop → cout << arr[i]



10 →



Char	ASCII
a	97
b	98
c	99
z	

Char	ASCII
A	65
B	66
C	67
z	

L	76
o	111
v	118
e	101

10 → 0

ASCII → 0

char Array Other way to take input

cin → delimiter

getline

enter

tab

space

My Name is Love Bhatnagar

My
Name
is
Love

My

Name

is

Love

tab

tab

→ char Array cin > getline (array size delimiter)

char arr [10];

loop ✓

cin >> arr

skips < enter & tab &

My
name
is
Love

cin > getline ()

cout << arr

delimiter → character → found → input stream

input / o/p stop

cin >> a

— → space
| n → gets
| t → tab

→ find length of char Array

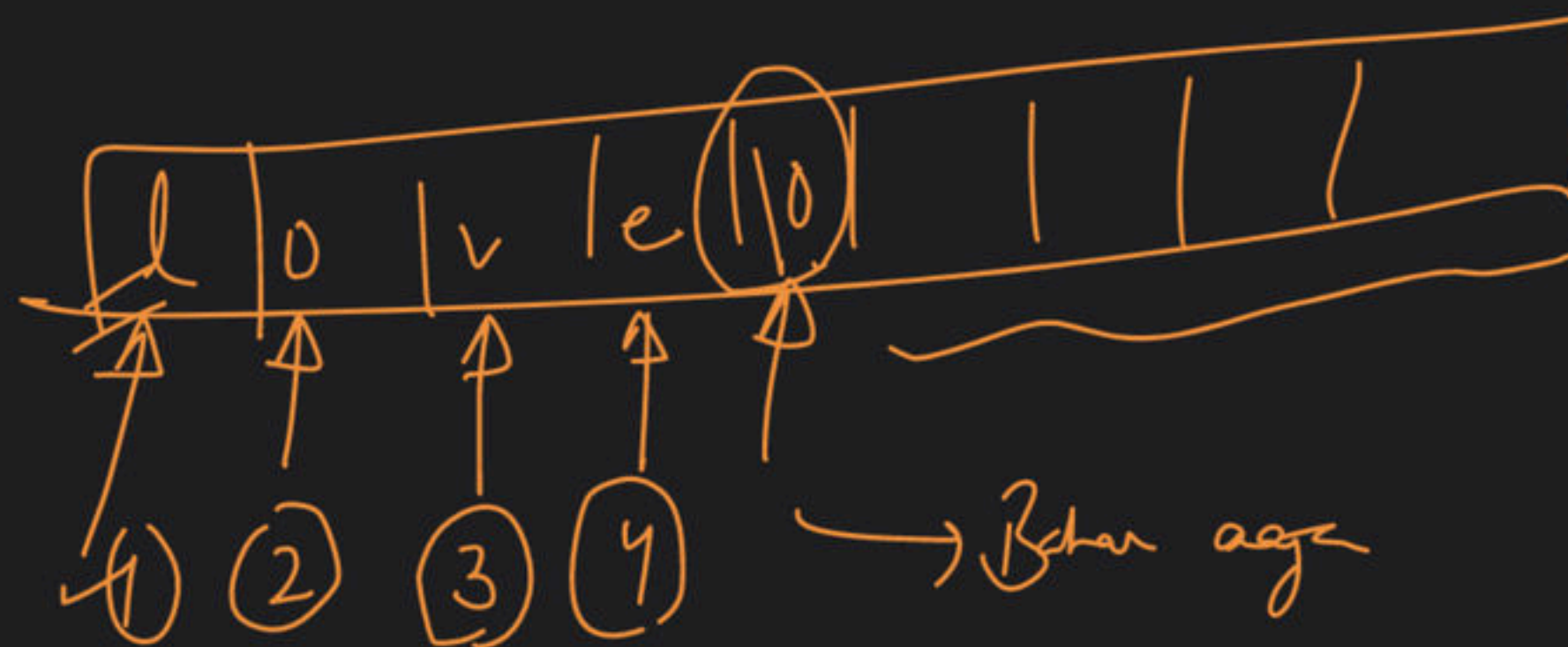
char arr[10];

cin >> arr

love

findLength()

4



count = 0, n = 8

```
for (int i = 0; i < n; i++)
```

```
{
```

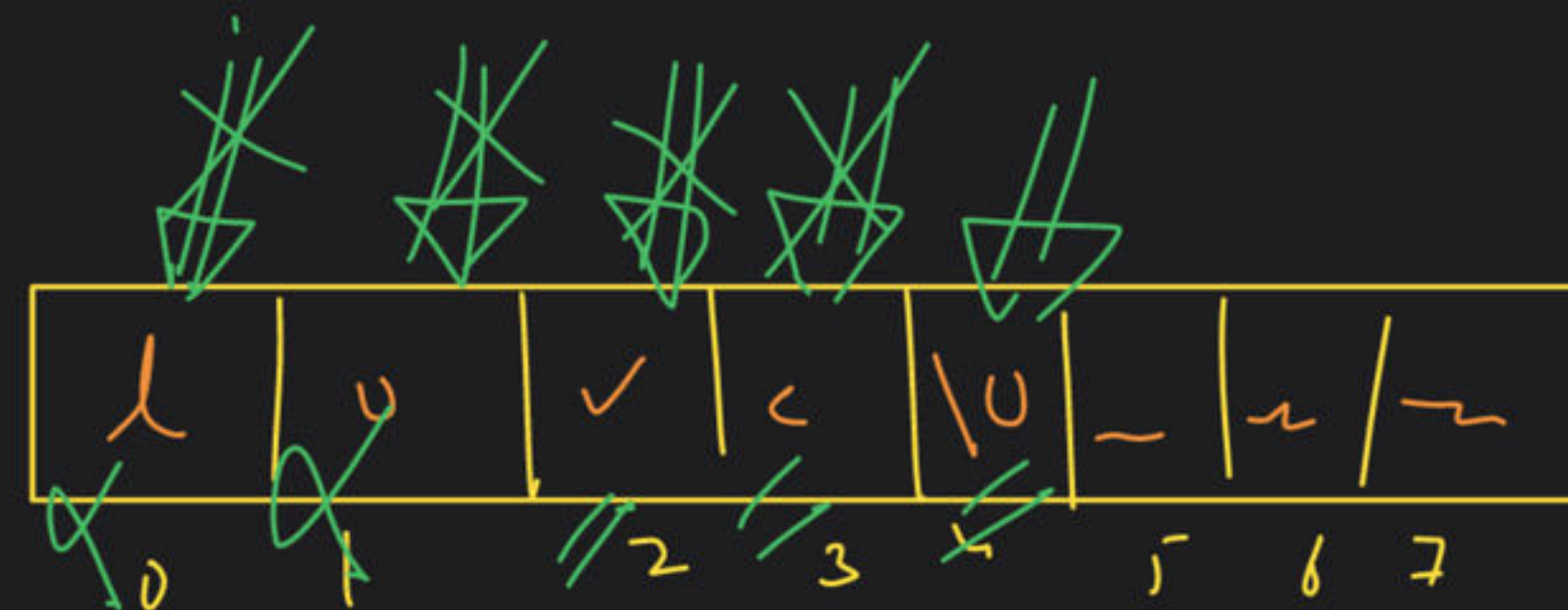
```
    if (arr[i] == '\0')
```

```
        break;
```

```
    else
```

```
        count++;
```

```
}
```



count = ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4 → return

O(n)

→ Replan character

@ → ' '

inp

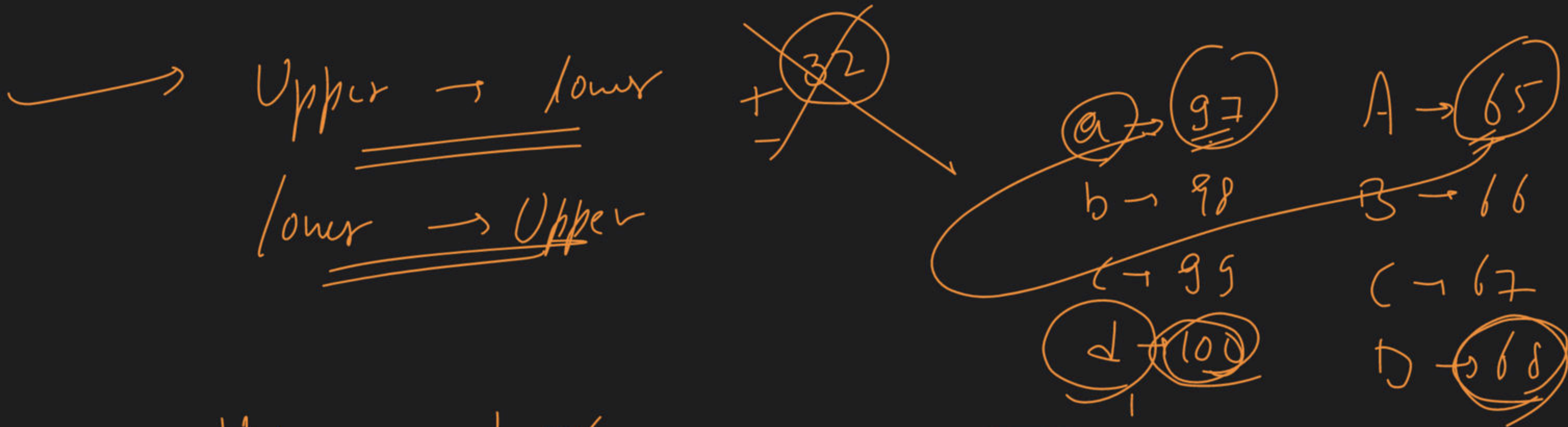
→

My @ Name @ is @ low @ Balke
↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑
2 0 0 1 0 0 0 0 0 0 0 0

o/p

→

My — Name — is — low — Balke



Upper \rightarrow lower

$$ch = ch - 'A' + 'a'$$

$$= 'A' - 'A' + 'a'$$

$$= 0 + 'a' = 'a'$$

A

$$65 - 65 + 97$$

$$68 - 65 + 97$$

~~ch = 'A'~~

D

A \rightarrow A

D

$$68 - 65 + 97$$

$$= 3 + 97 \rightarrow 100 \rightarrow$$

D

2

2

char ch;

(in > 5 && ch

cout <<

(int)ch



int a = ch



cout << a

$U \longrightarrow L$

$ch - 'A' + 'a'$

$L \longrightarrow U$

$ch - 'a' + 'A'$

2 min

Break

→ Char array → Reverse

2 min



END

i/p →

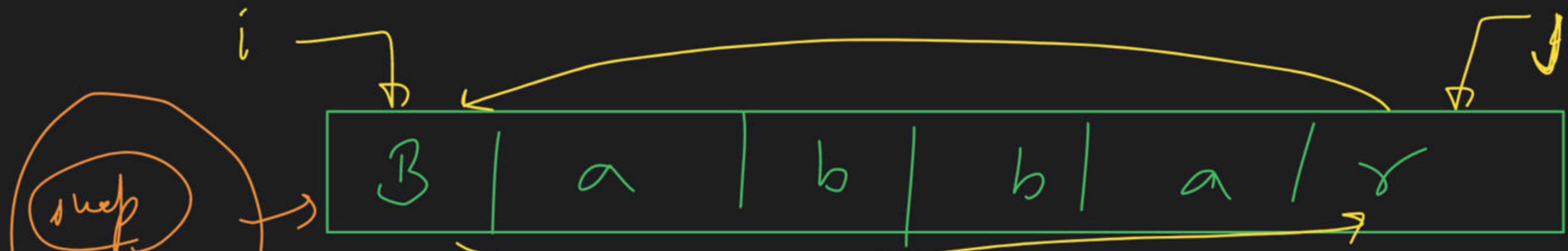
L	0	V	E
---	---	---	---

o/p →

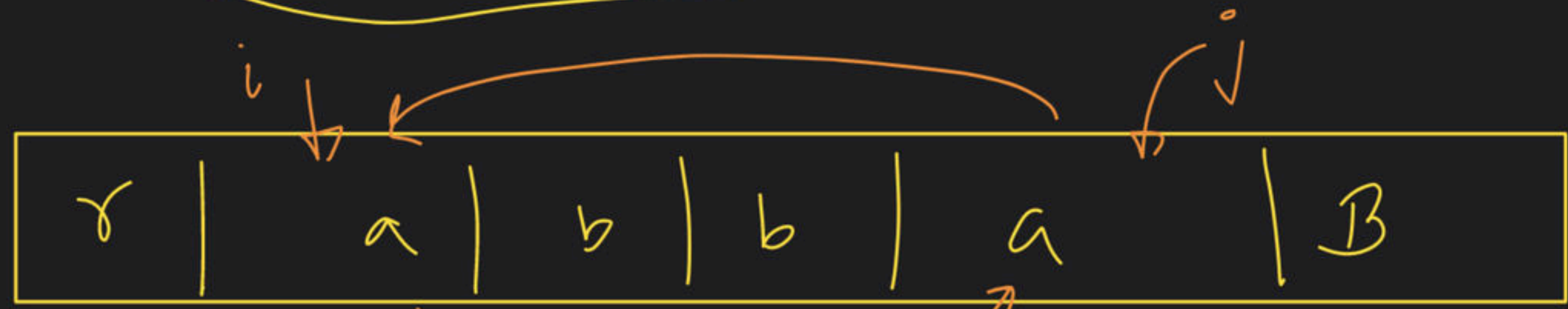
E	V	0	L
---	---	---	---

Battery
Change
time

→ EVOL ← reverse



swap
 $i++$
 $j--$



$j \leftarrow \text{arr}$



$i > j \rightarrow \text{Return}$

Palindrome

~~did~~
~~idi~~

DAD

L → R

R → L

MOM

MAM

MADAM

NITIN

R	A	C	E	C	A	R
---	---	---	---	---	---	---

WOW

NOON

L → R → RACECAR

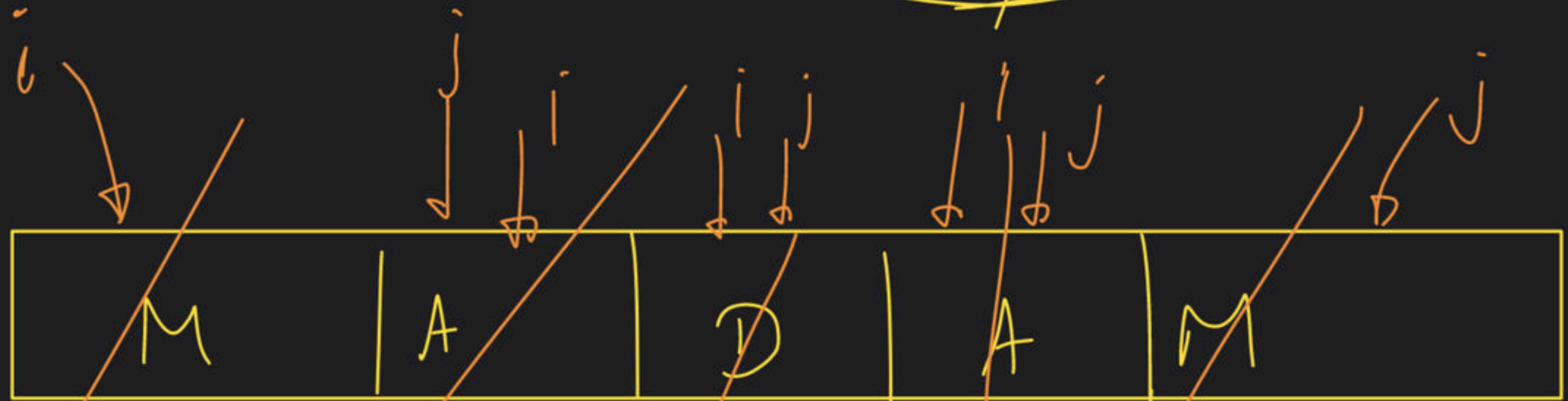
KANAK

LOL

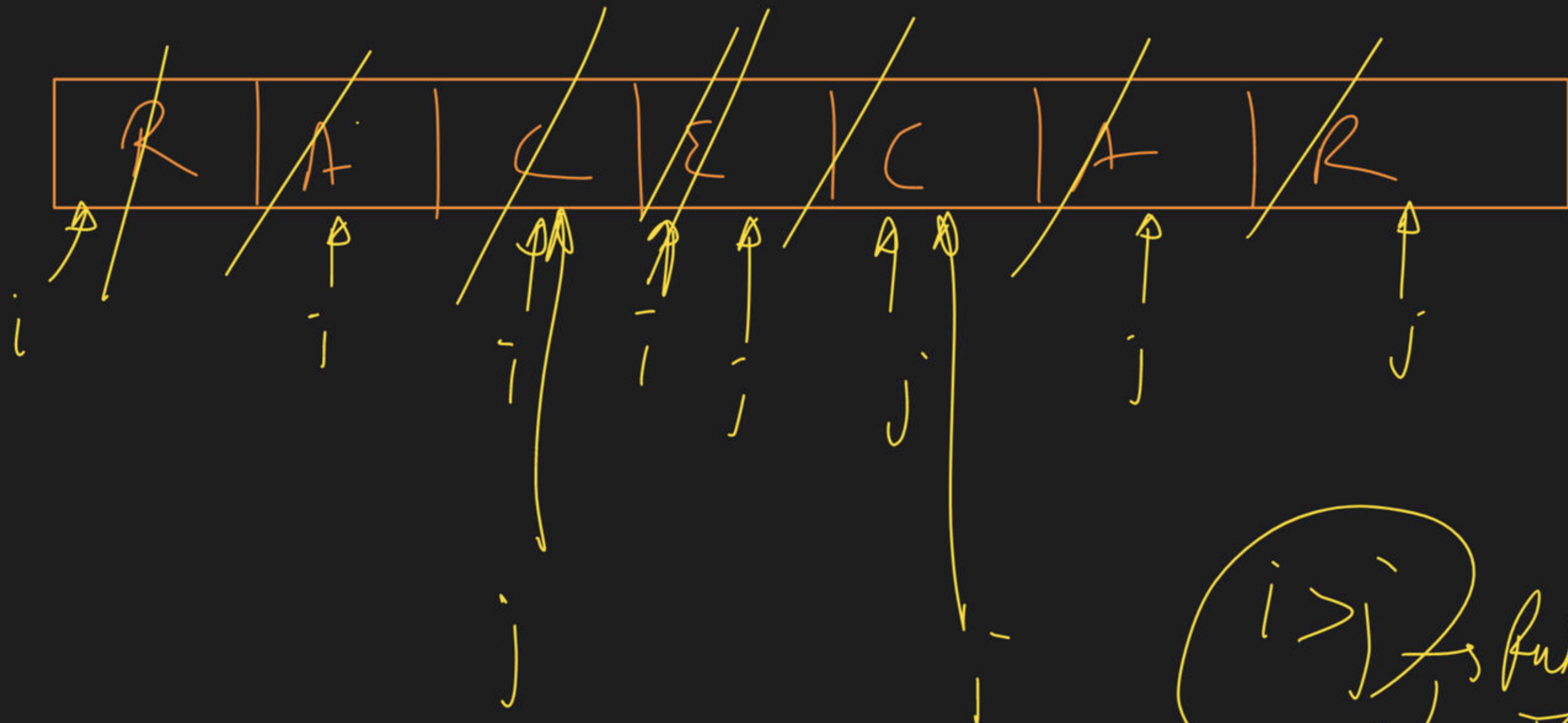
R → L

RACECAR

ip → word

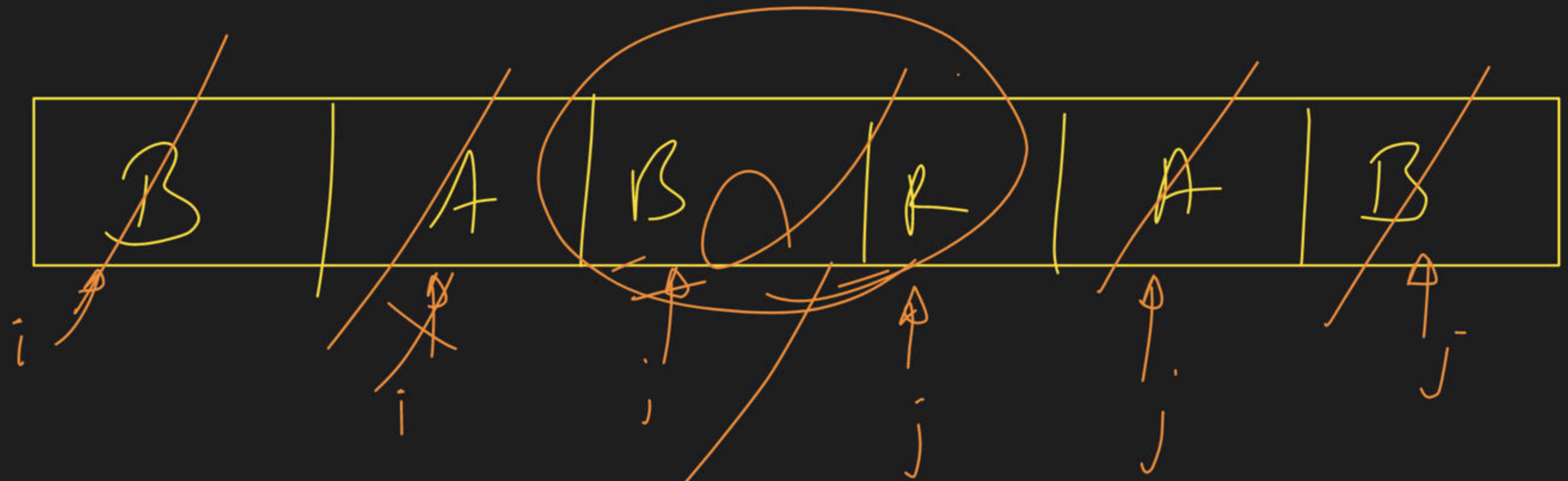


$i > j$ → True



$i > j$ → Puko
Jaw

True



action falsch



strlen

strcat

strcmp



string



char array

common

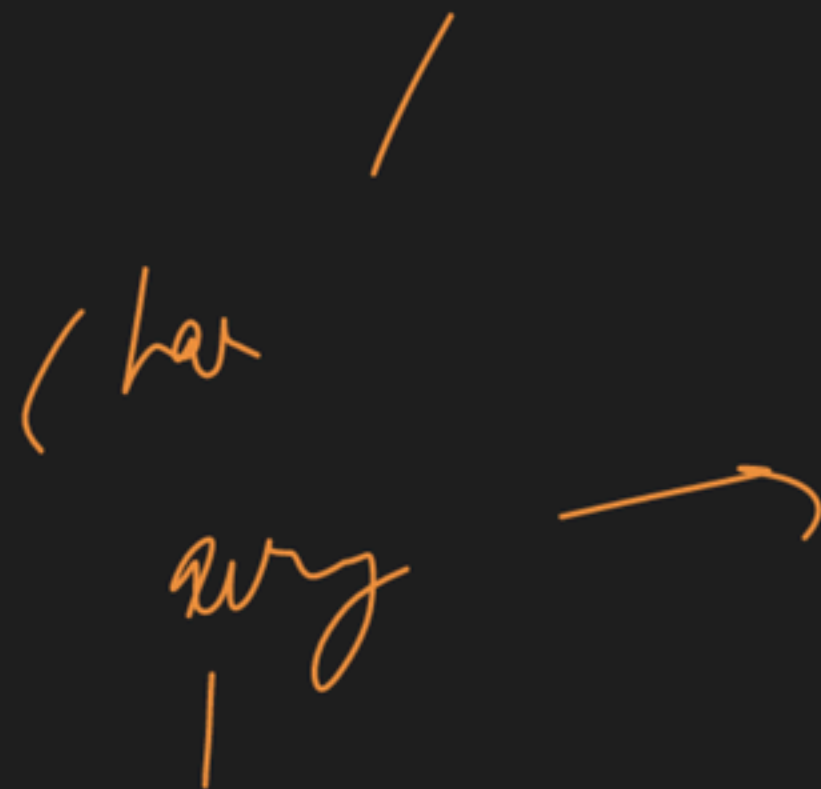
function

String



A handwritten word 'String' is enclosed in a hand-drawn rectangular box. An arrow originates from the right side of the box and points horizontally towards the right edge of the image.

has
any



The words 'has' and 'any' are written vertically, with 'has' above 'any'. A horizontal arrow points from the right side of the word 'any' towards the right edge of the image.

string

string name;

length()

index

name[3]

name.push_back('a');

name.pop_back();

cin >> name;

cout << name;

functions: -

• begin()

• end()

• front()

• back()

[]

at()

+

substr()

find()

compare()

empty()

clear()

push_back()

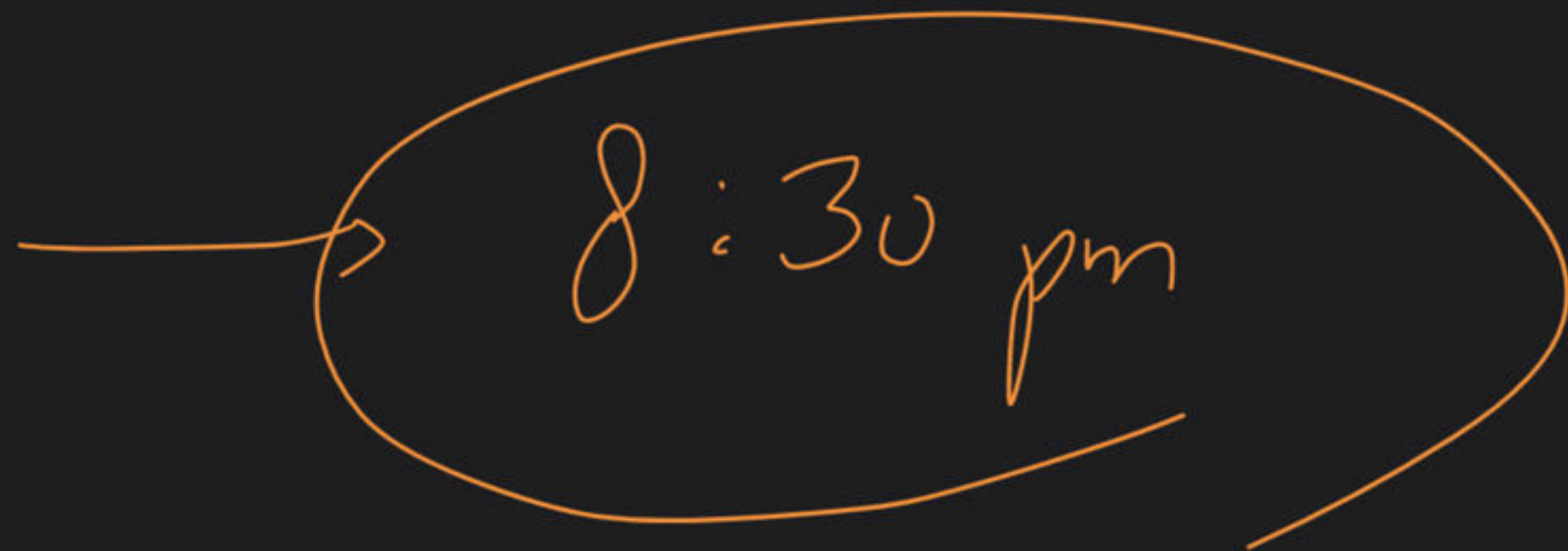
pop_back()

length()

reverse

Revision

Kal



Extra class -

