DC-Basics of Programming

Special class

Love Babbar • Apr 5, 2025

L1-Basics of Programming

Special class

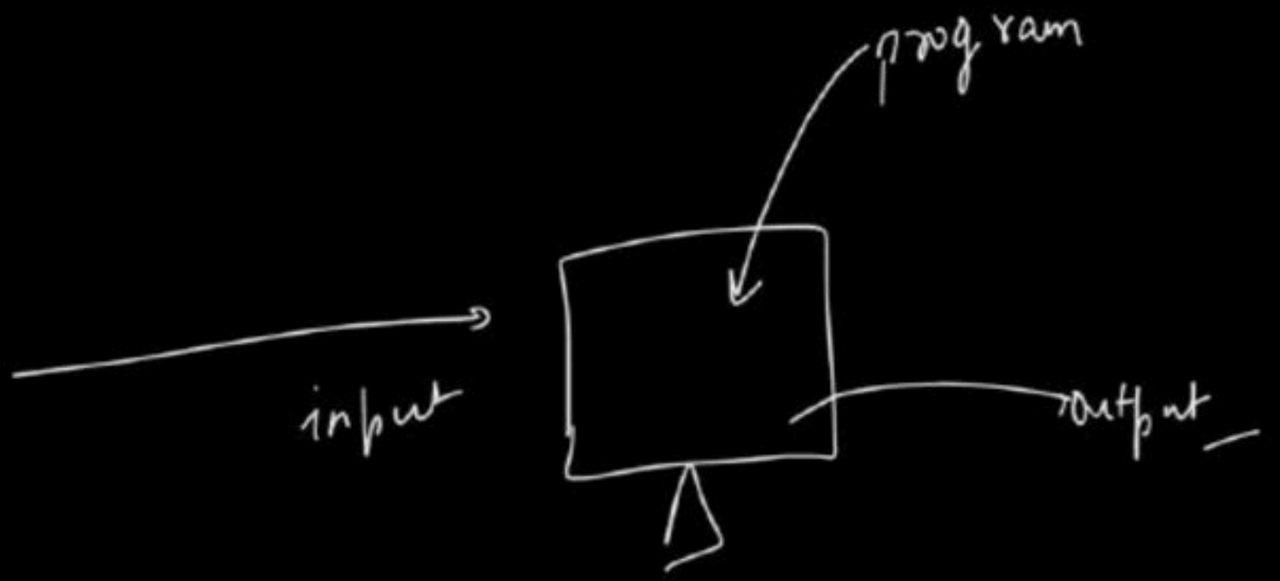
Love Babbar • Apr 2, 2025

L1 - Basics of Programming

Flowcharts and PseudoCode

—> by Codehelp

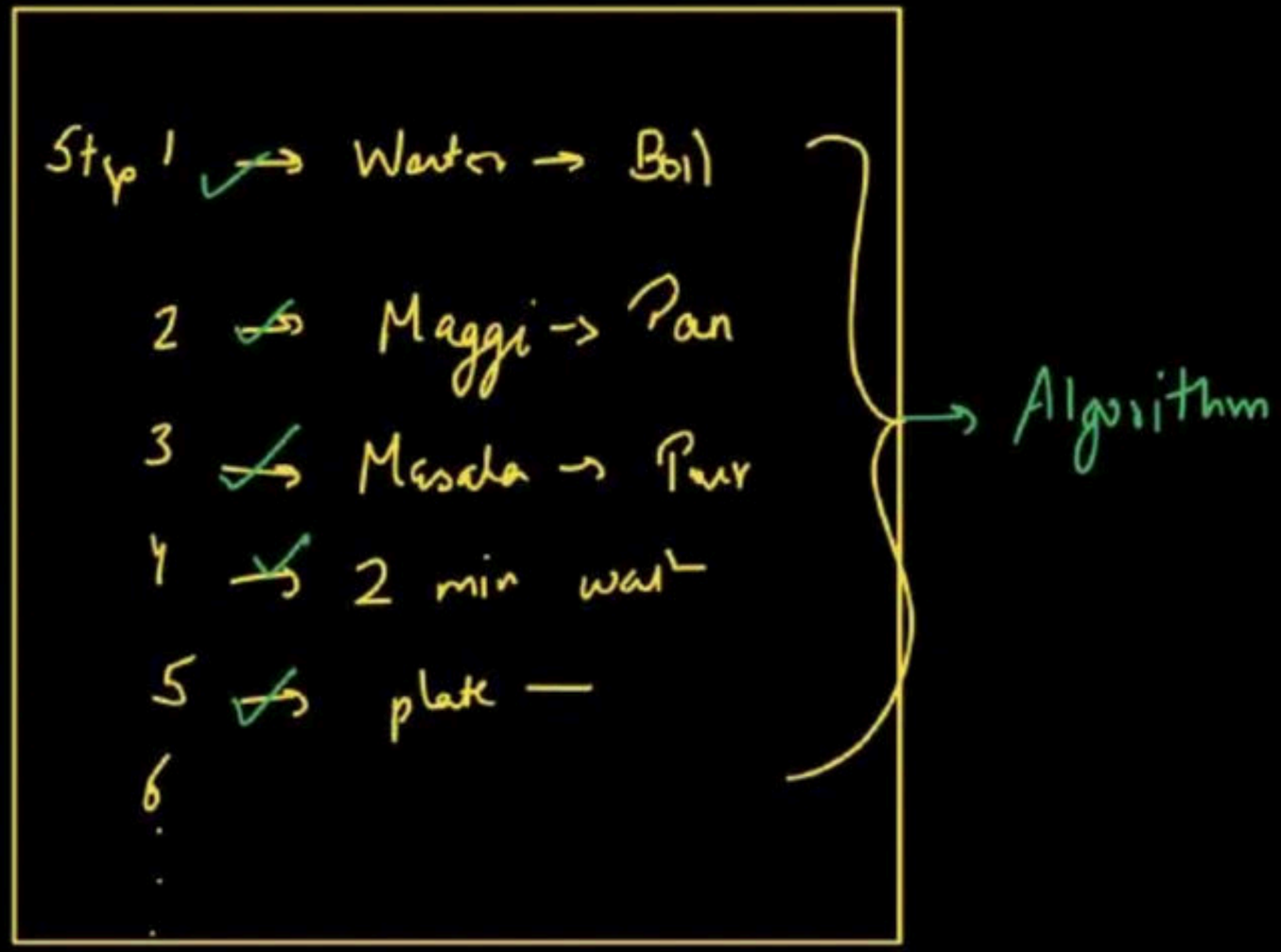
What is Programming ?

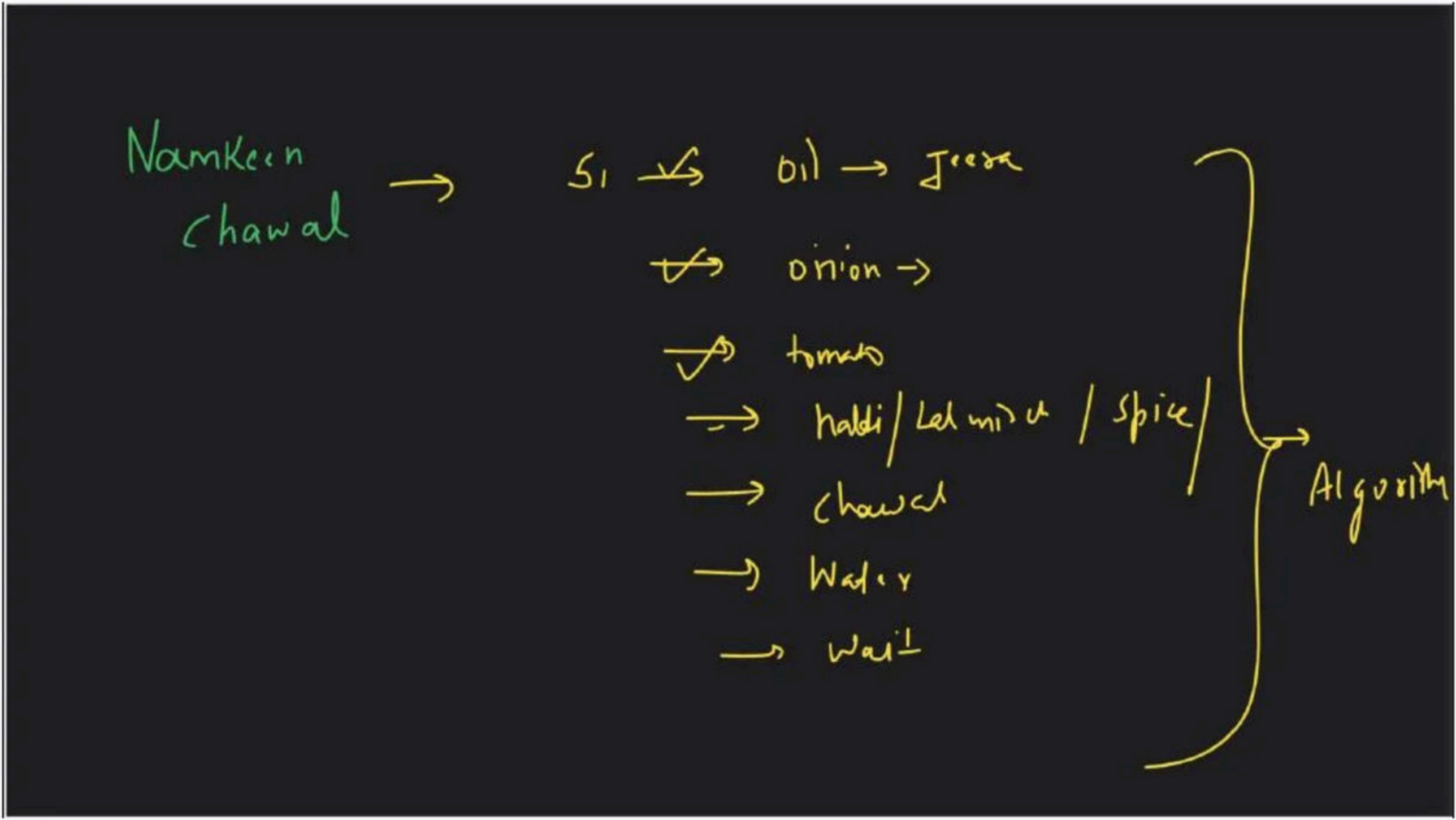


What is Algorithm ?

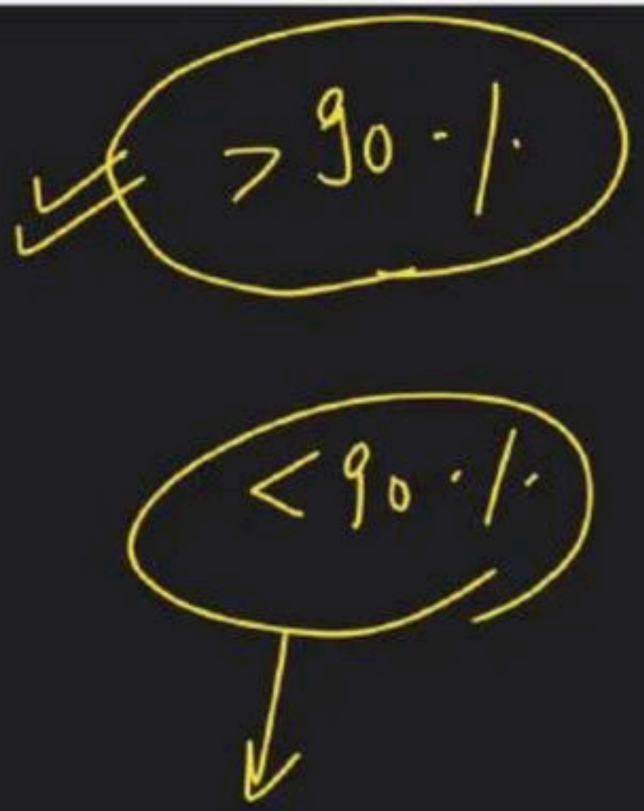
Maggi →

Shrey Vohra




Namkeen
chawal
Si
oil
onion
tomato
haldi / salt / spice /
chawal
Water
Salt
Algorithm

Pasta → ?



How to approach a Problem ?

Thought process

① Let's understand the Problem

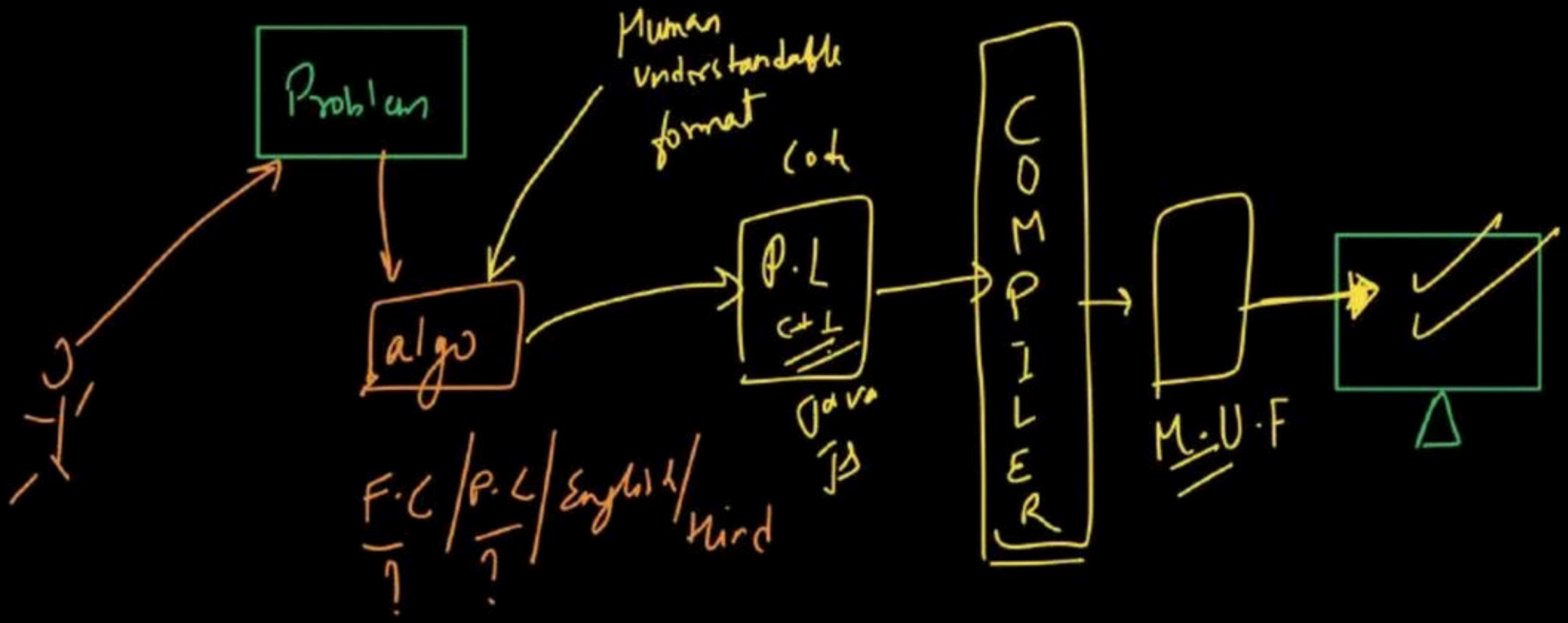
② Analyse problem → given values / relate formula / constraints
side = 4 cm

③ Create a approach → Algorithm

Optimise

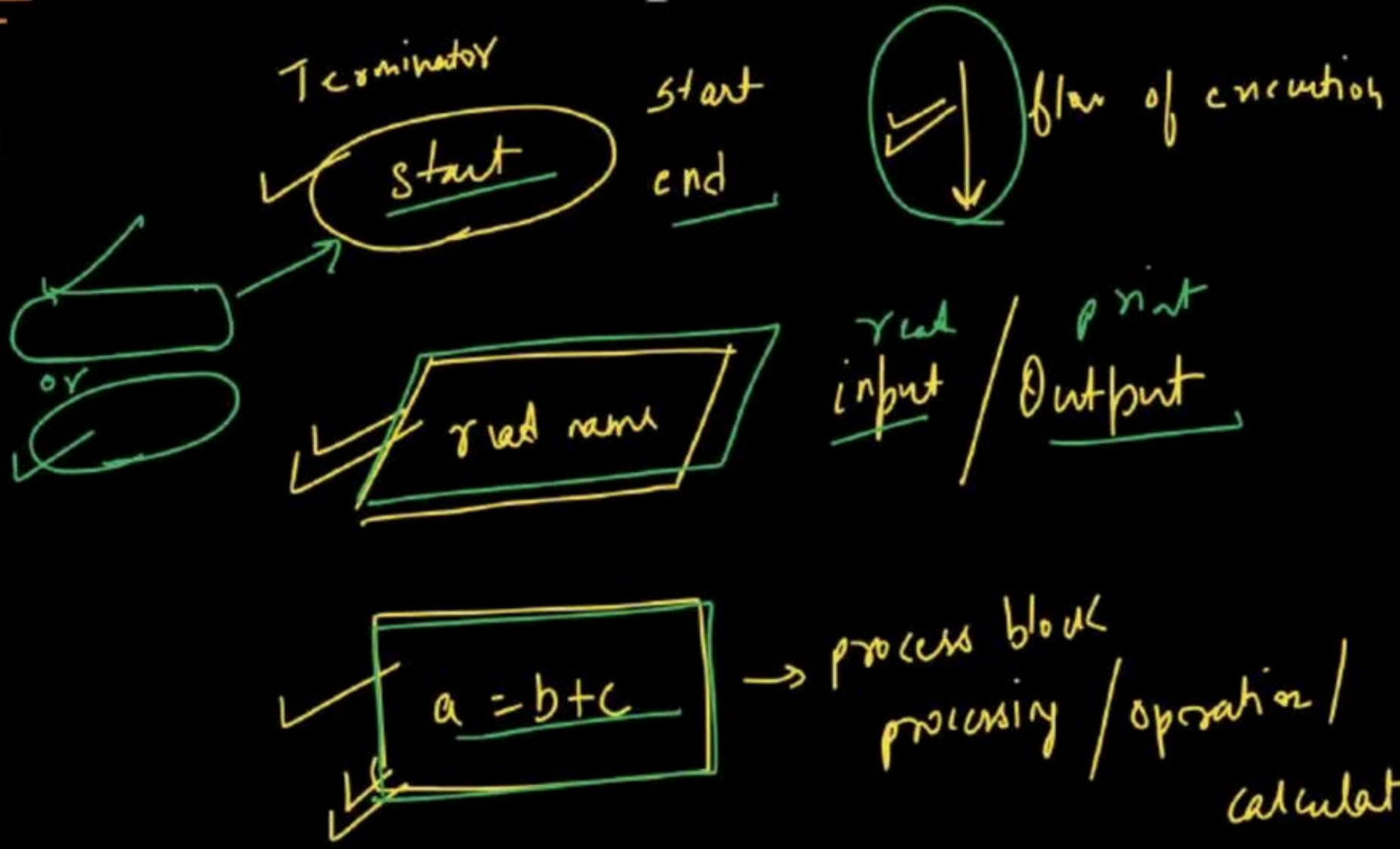
area = s × s
= 4 × 4
= 16

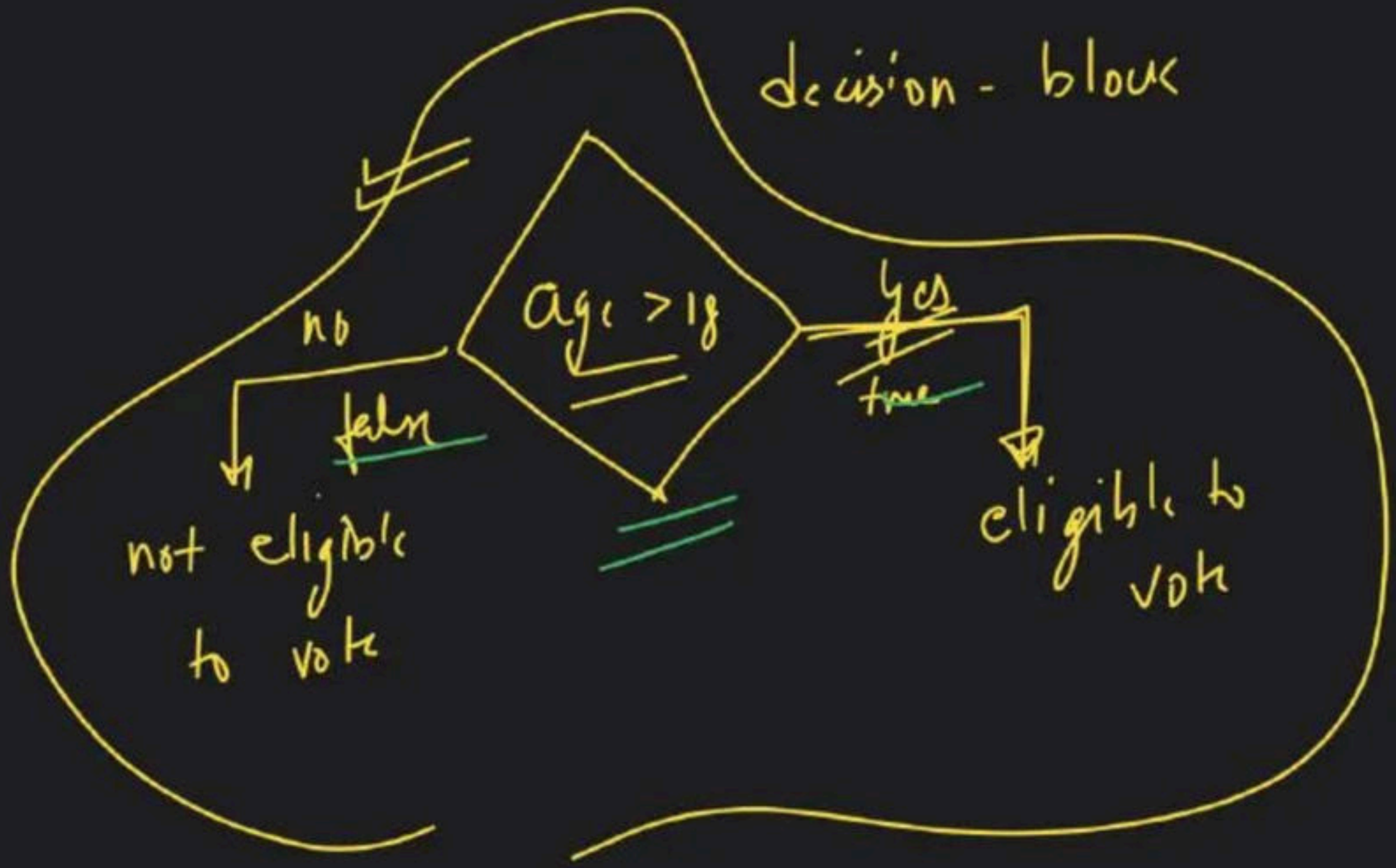
Using a computer to solve a problem



Flowchart and its components:

Diagrammatic representation of your Algorithm




decision - block
age >18
no
false
not eligible to vote
yes
true
eligible to vote

Pseudocode:

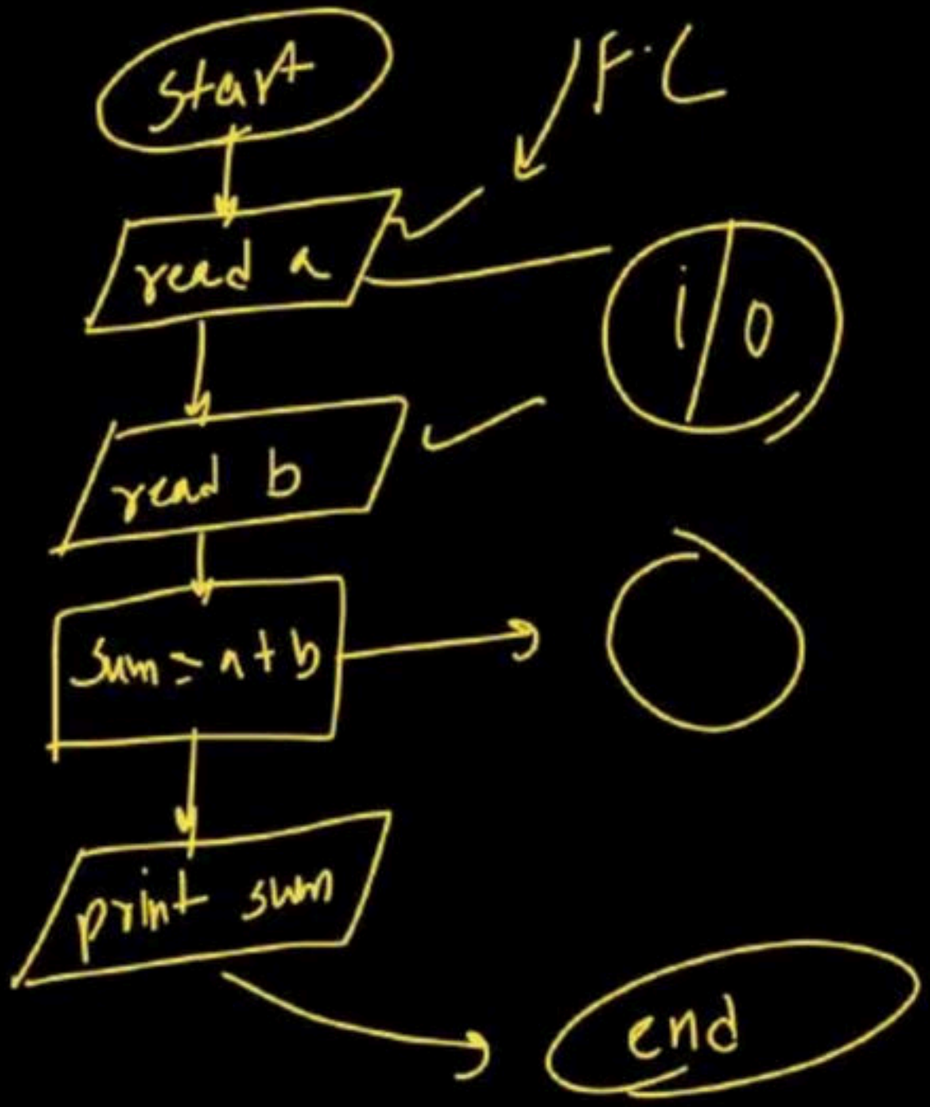
Nakli → code → denote → dusri bhasha

Sum of 2 no

- → Start
- → read the value of a & b
- → sum = a + b
- → print sum
- → end

- → Start
- → read first name
- → read middle name
- → read last name
- → f.n = f + m + l
- → print full name
- → end

Design Flowchart - Print Sum of a and b



P.L

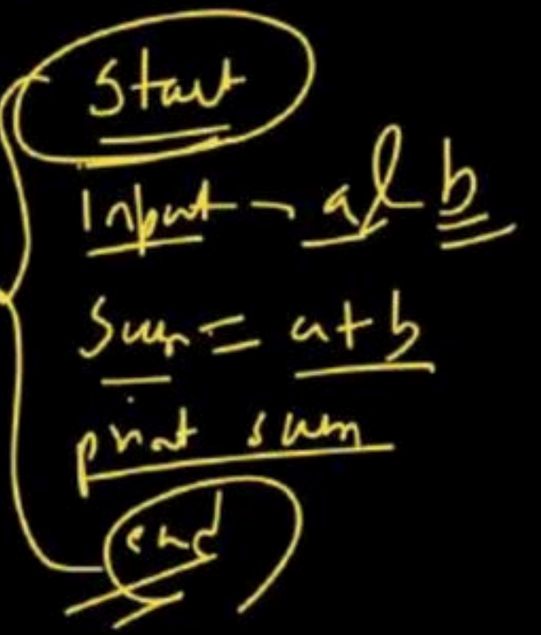
a = 10

b = 5

Sum = a + b

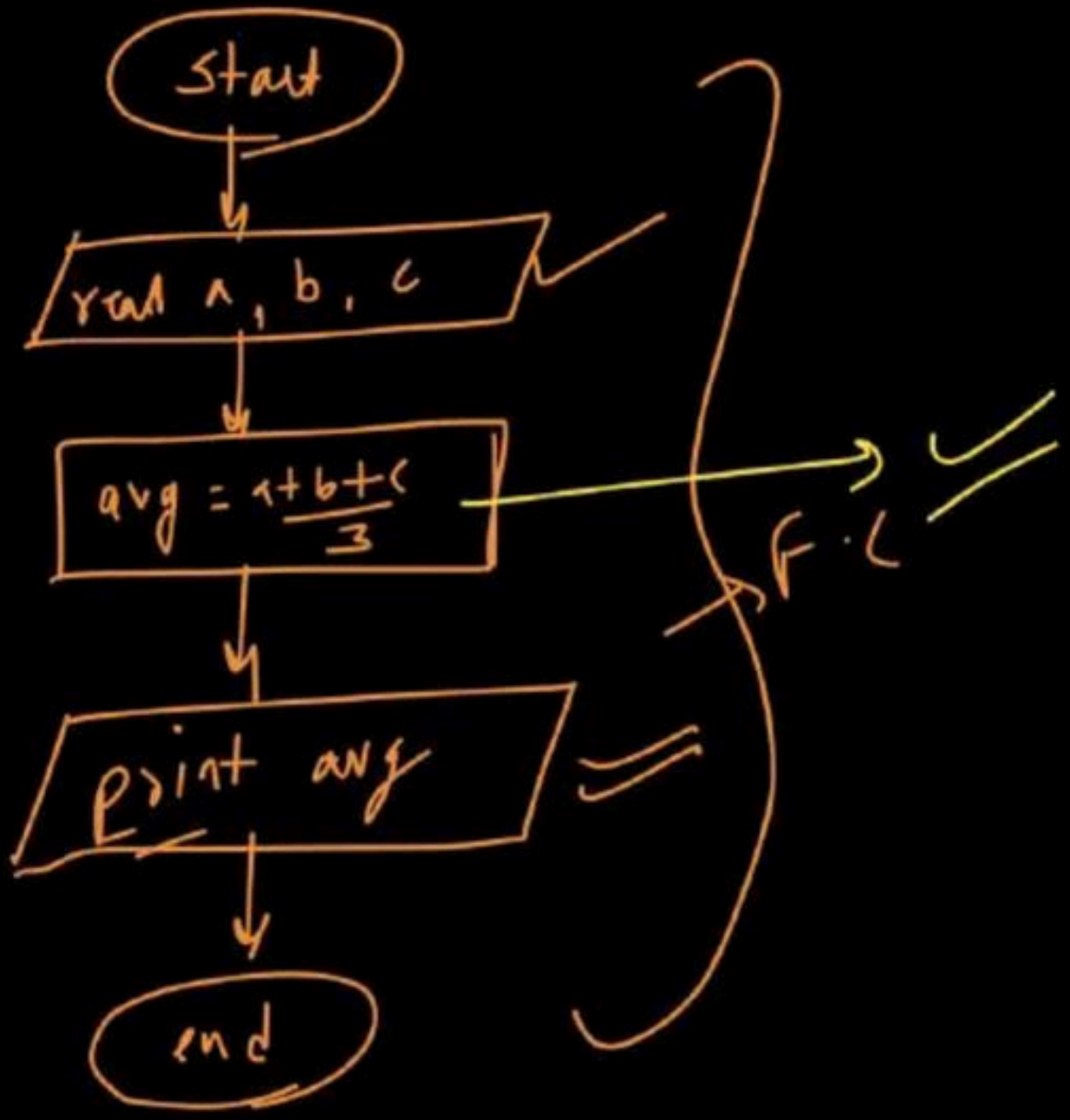
= 10 + 5

Sum = 15



print 15

Design Flowchart - Print Average of a, b and c



Start

read a = 10, b = 12, c = 14

avg = (a+b+c) / 3

= (10+12+14) / 3 = 36 / 3 = 12

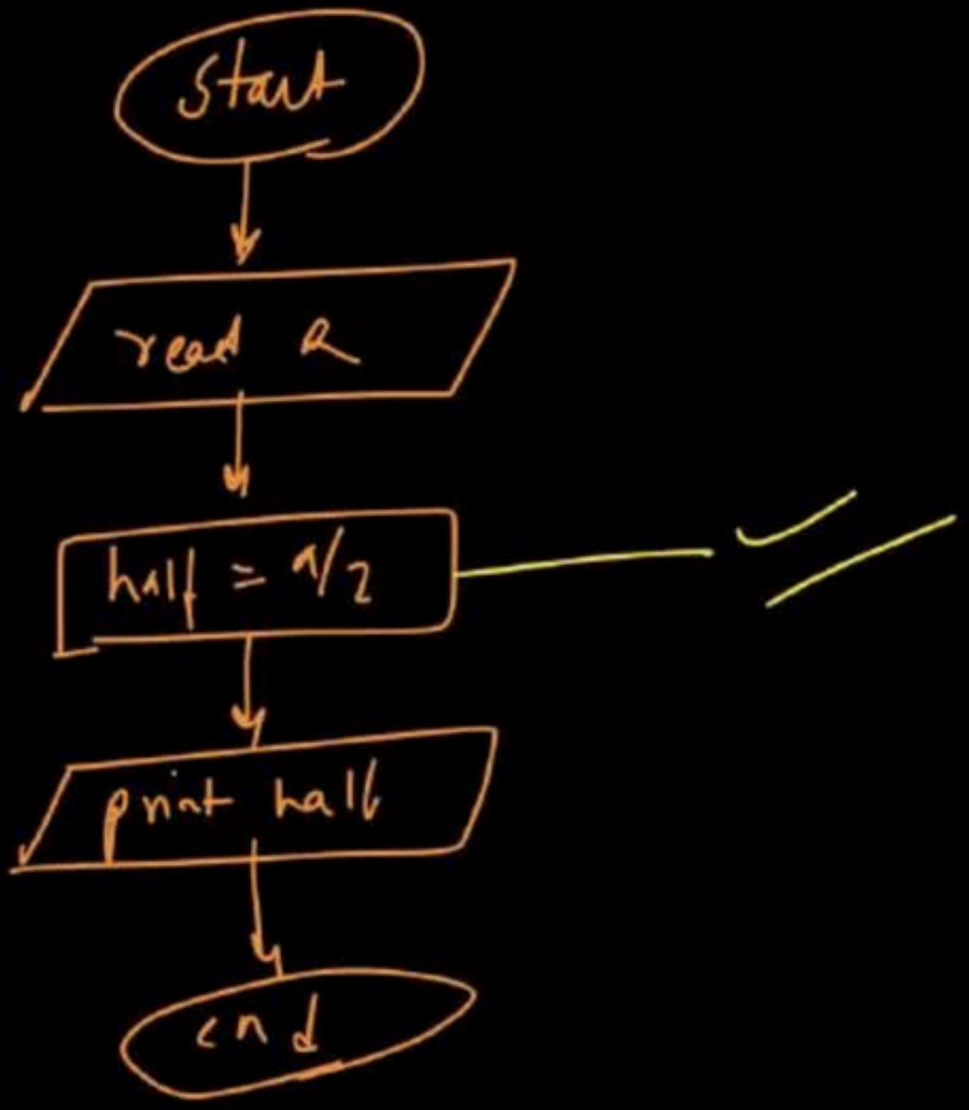
avg = 12

print avg

end

Avpit Pathak

Design Flowchart - Print half of a



start

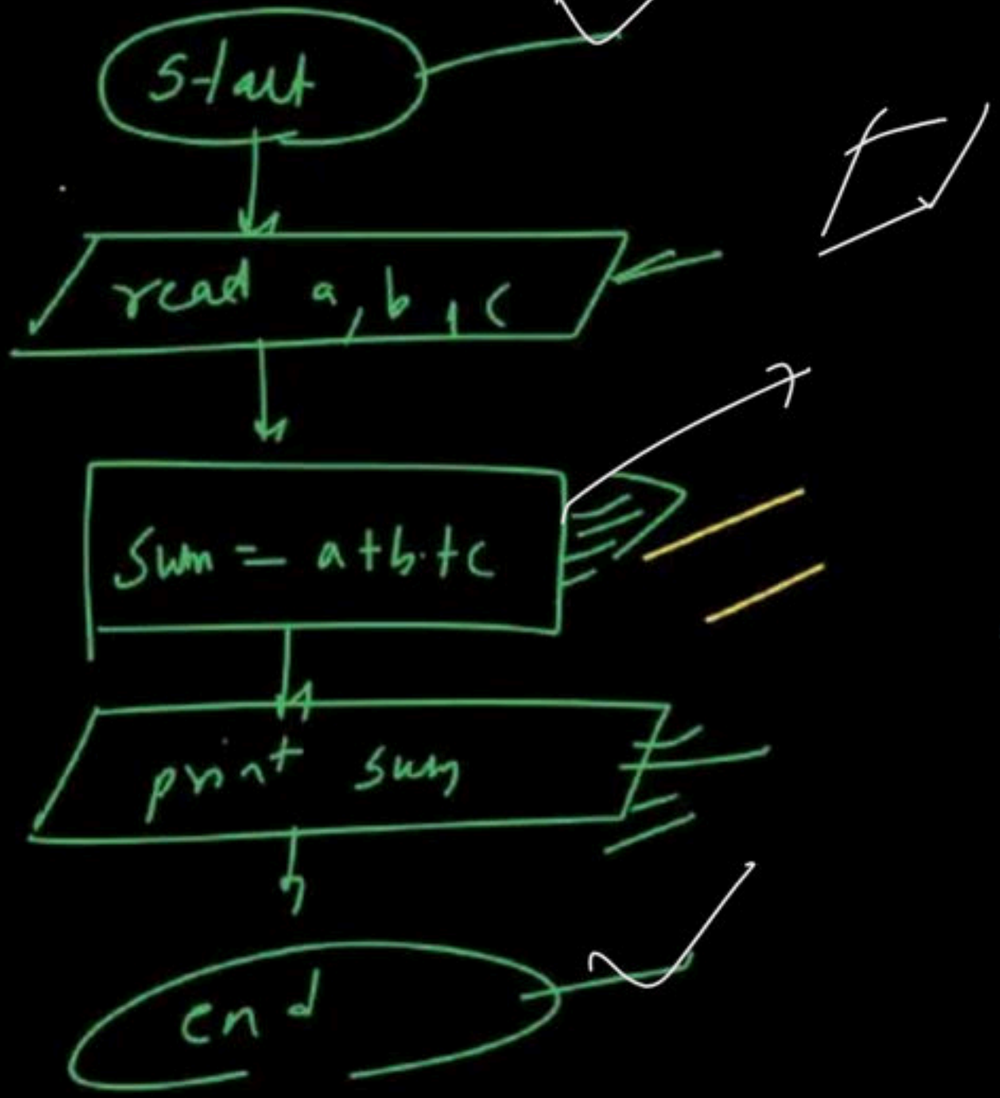
read a

half = a/2

print half

end

Design Flowchart - Take Input and add 3 numbers



start

read a, b, c

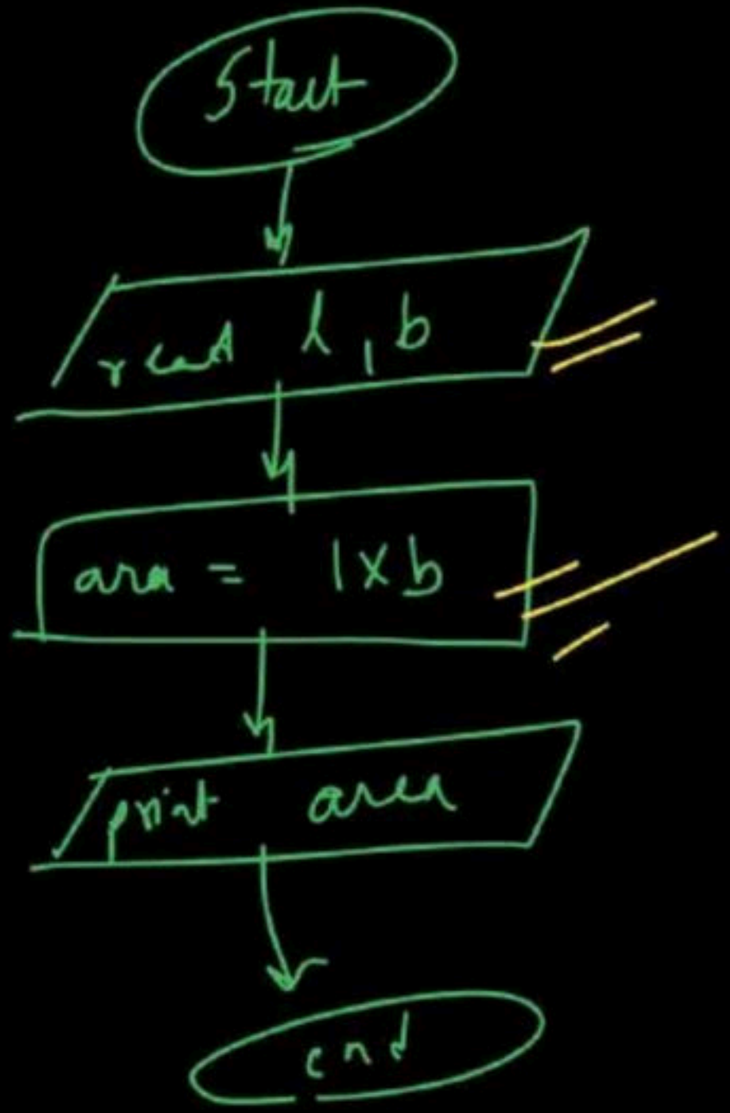
sum = a+b+c

print sum

end

Design Flowchart - Area of Rectangle

area = len × br



Start

read len, br

area = l x b

print area

end

2 min

Design Flowchart - Calculate Percentage

→ Start

→ read E, H, M, His, Sci

→ sum = E + H + M + His + Sci

→ total = 500

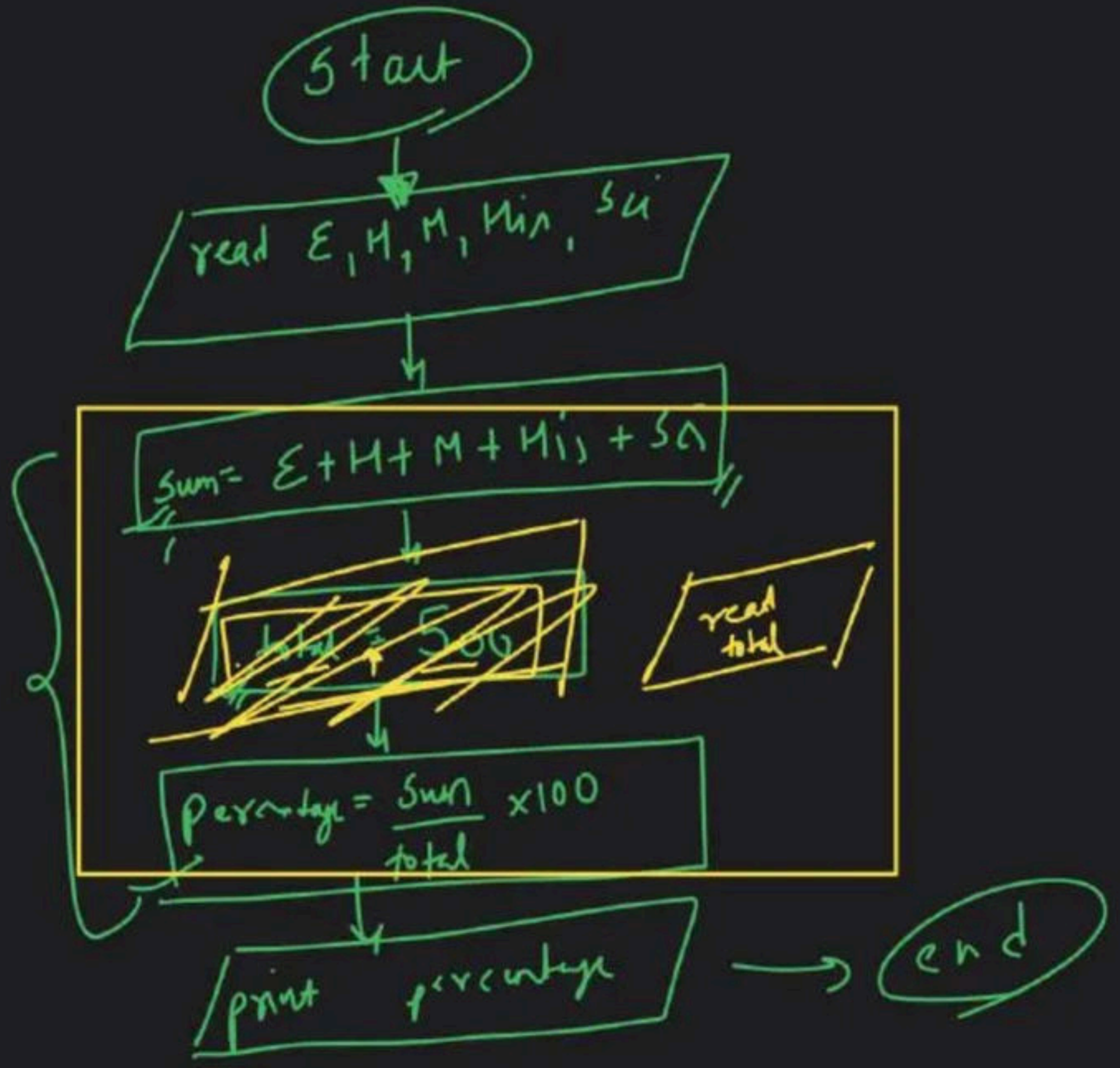
→ percentage = sum / total × 100

→ print percentage

→ end

total % :- (10 + 20 + 30 + 10 + 50) / 500 × 100

Sum of marks / total marks × 100

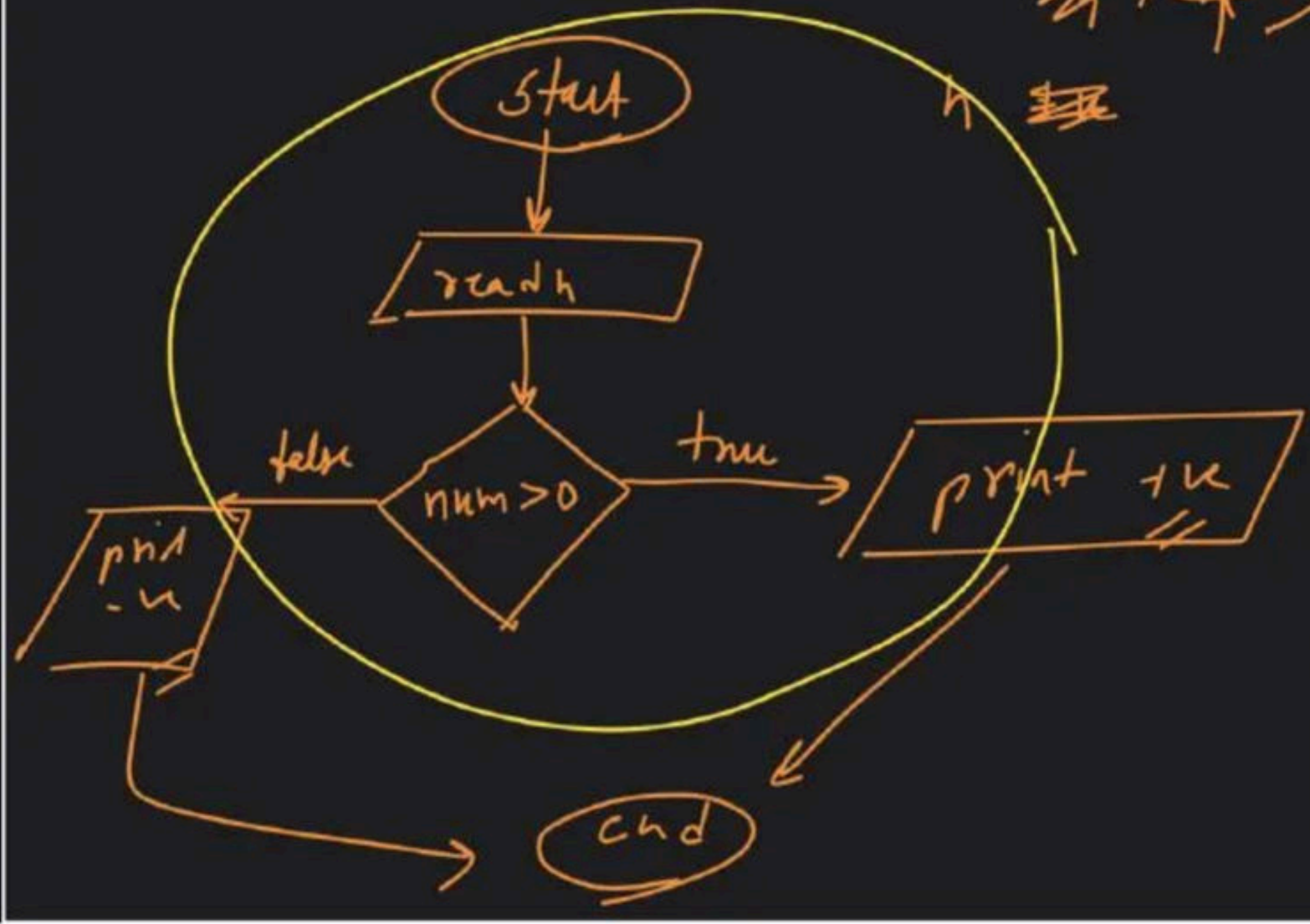

Start
read E, H, M, Mis, Sc
Sum = E + H + M + Mis + Sc
total = 500
read total
Percentage = Sum / total x100
print percentage
end

no → +ve / -ve

0 -ve

+ve → >0

-ve → <0



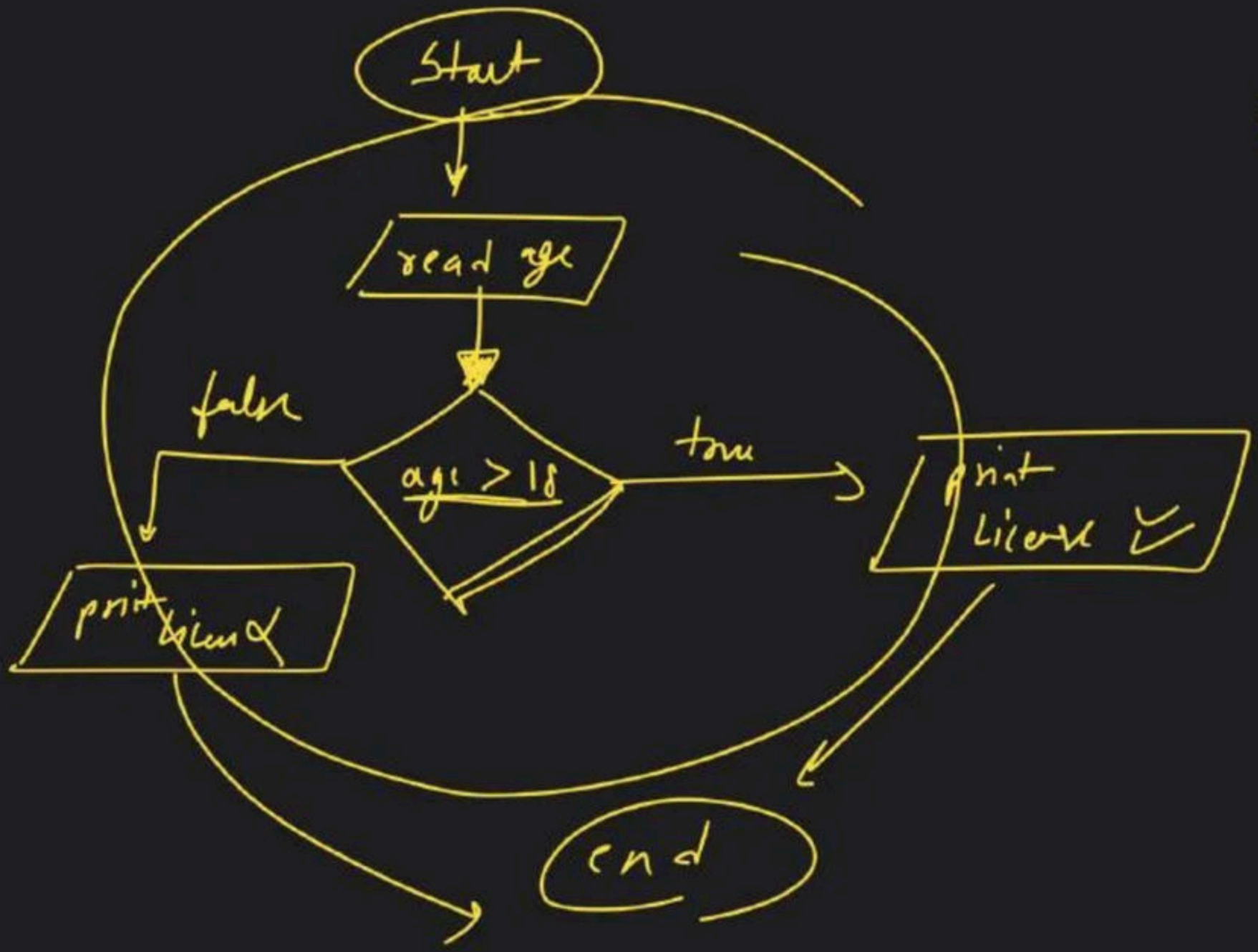
start

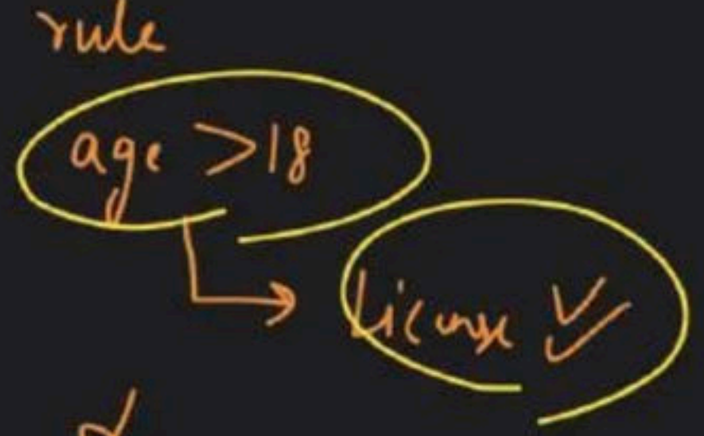
read num

if num >0, print +ve

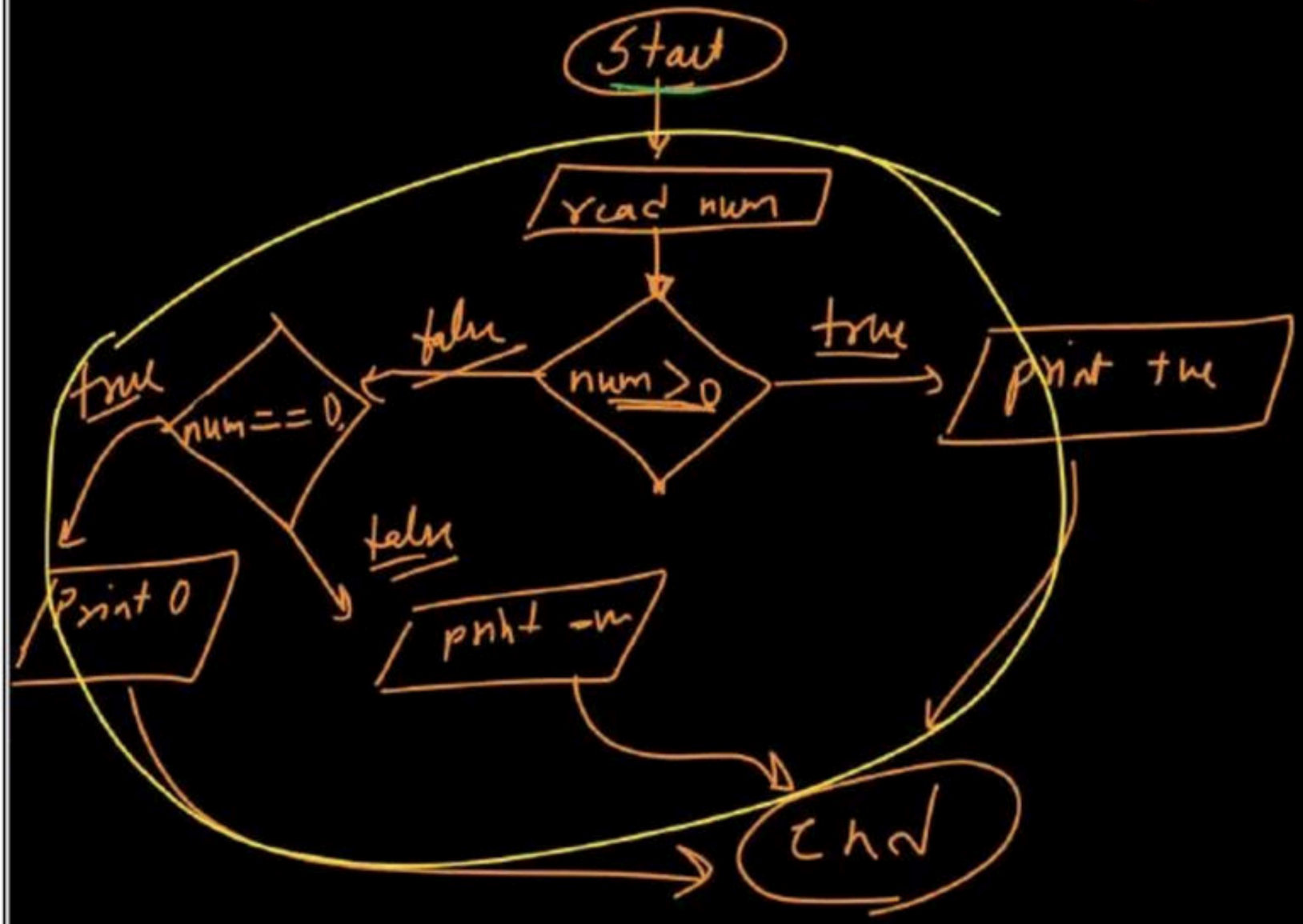
if num <0, print -ve

end


Start
read age
age > 18
false
true
print
License ✗
print
License ✓
end


rule
age > 18
License ✓
✗
License ✗

Design Flowchart - Check positive, negative or zero



Start

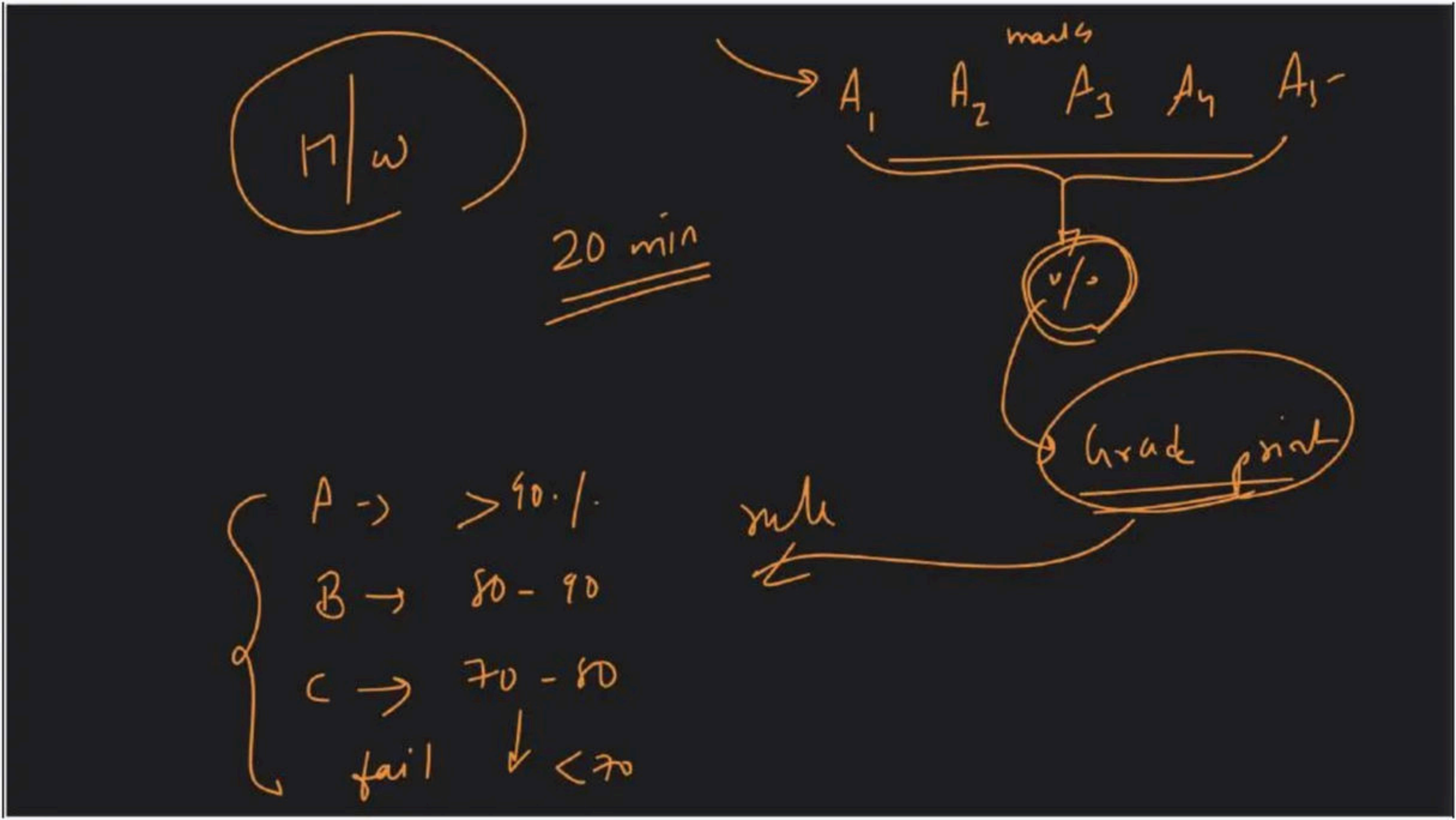
read num

if num > 0, print +ve

else if num = 0, print 0

else print -ve

end


H/w
marks
A1 A2 A3 A4 A5
20 min
V/v
Grade print
result
A -> >90.%
B -> 80 - 90
C -> 70 - 80
fail
<70

count = 0 1 2 3 4 5

limit = 5

Name → 5 print

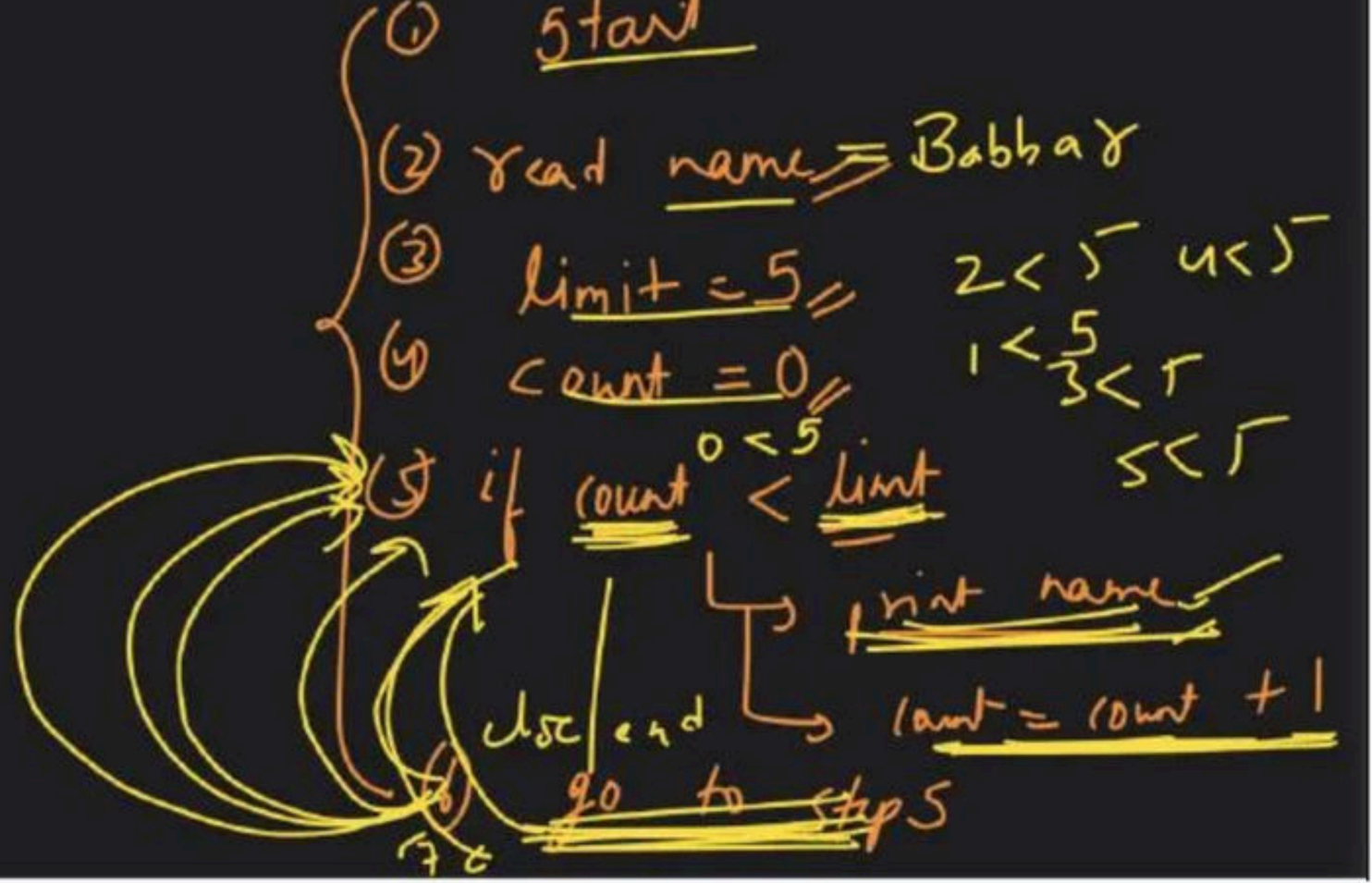
Babbar

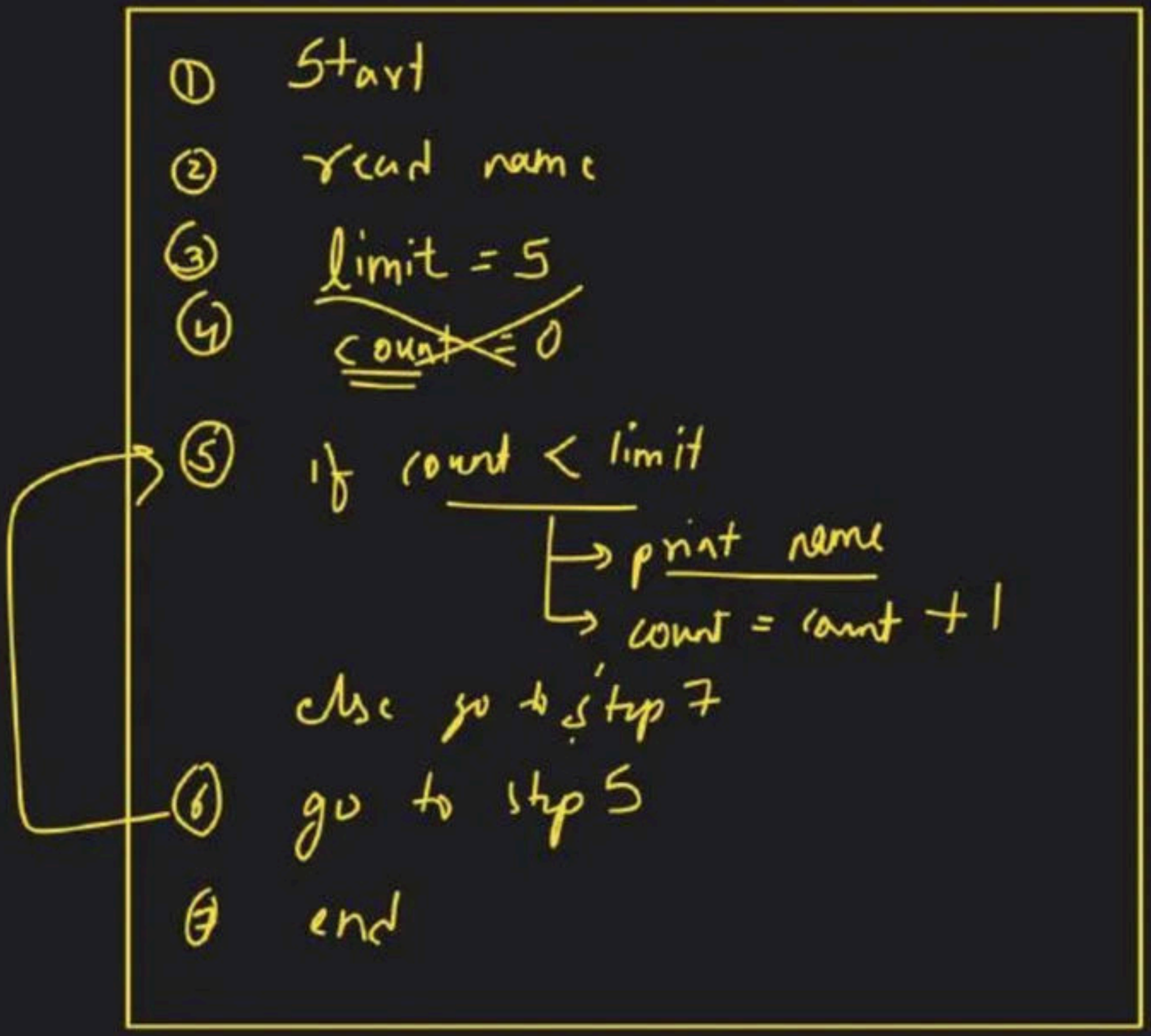
Babbar

Babbar

Babbar

Babbar

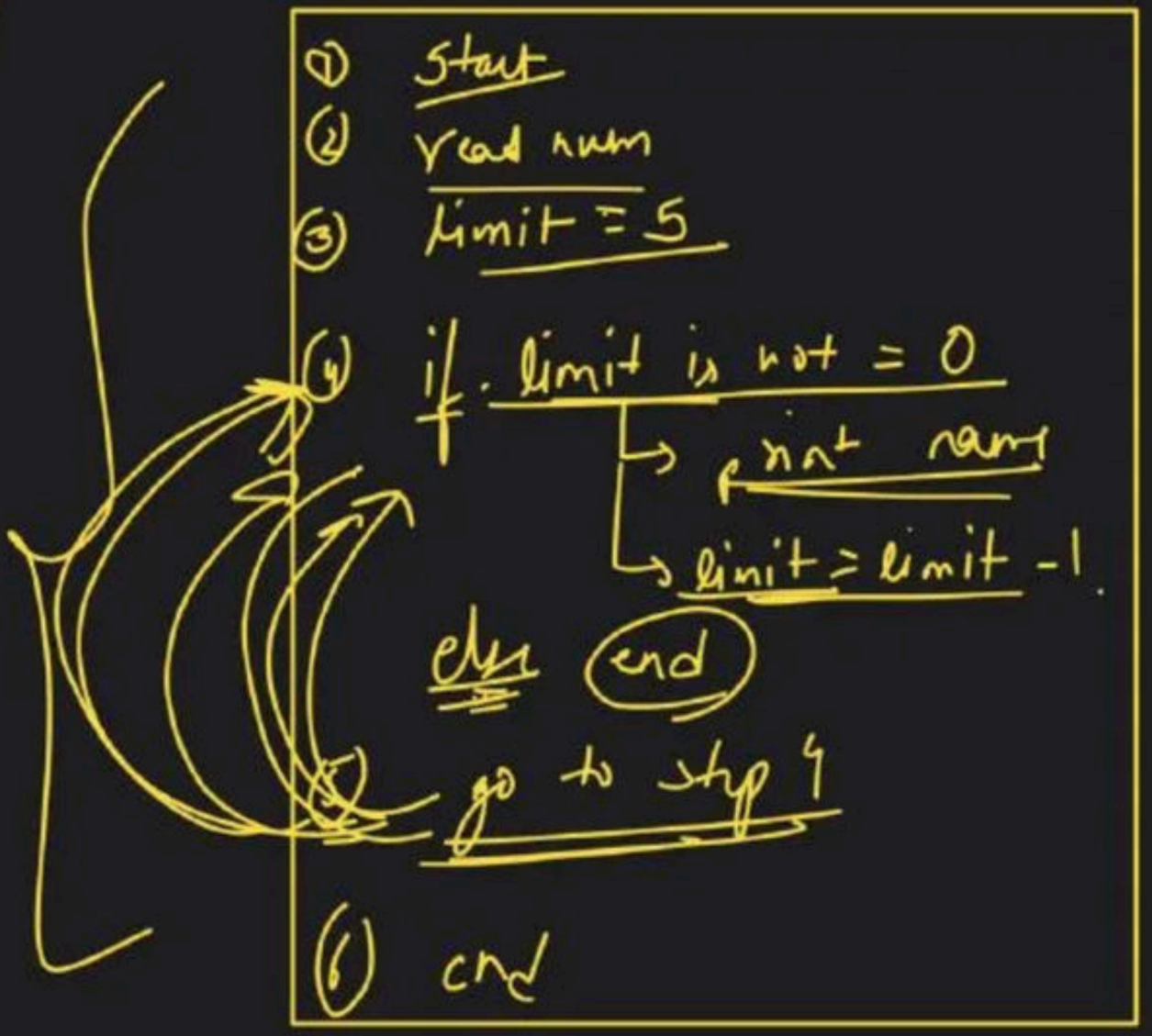


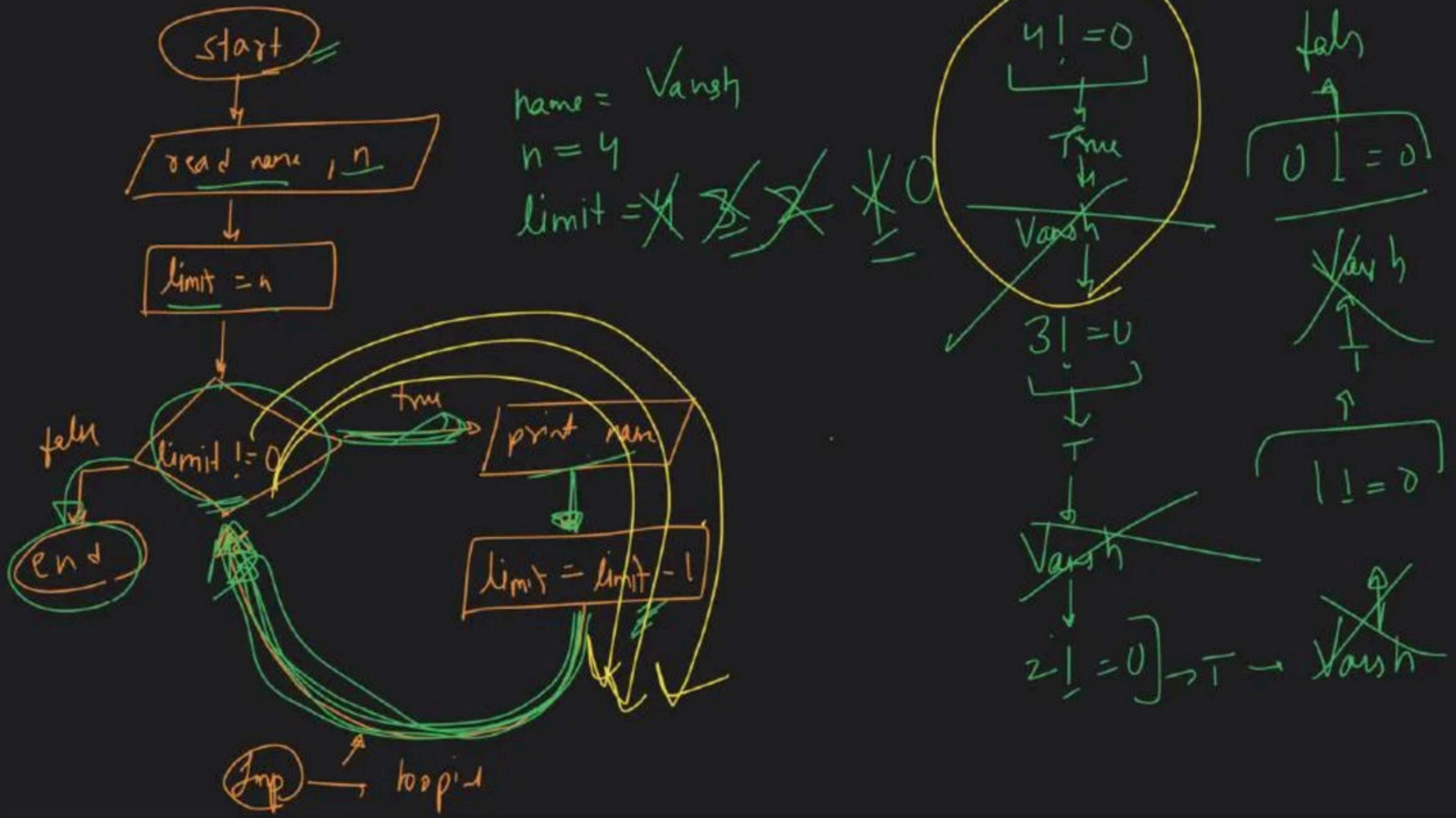

1 Start
2 read name
3 limit = 5
4 count = 0
5 if count < limit
print name
count = count + 1
else go to step 7
6 go to step 5
7 end

name = Babbar

limit :- 5 4 3 2 1 0

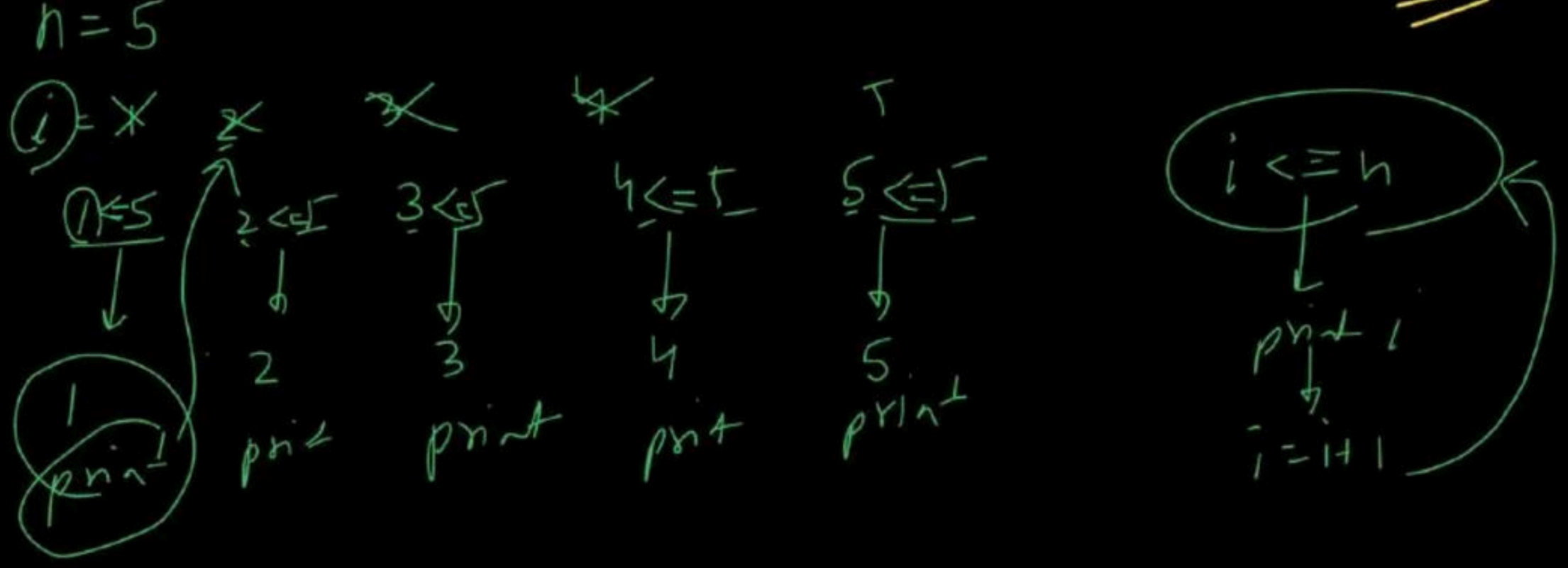
Babbar
Babbar
Babbar
Dabbar
Babbar

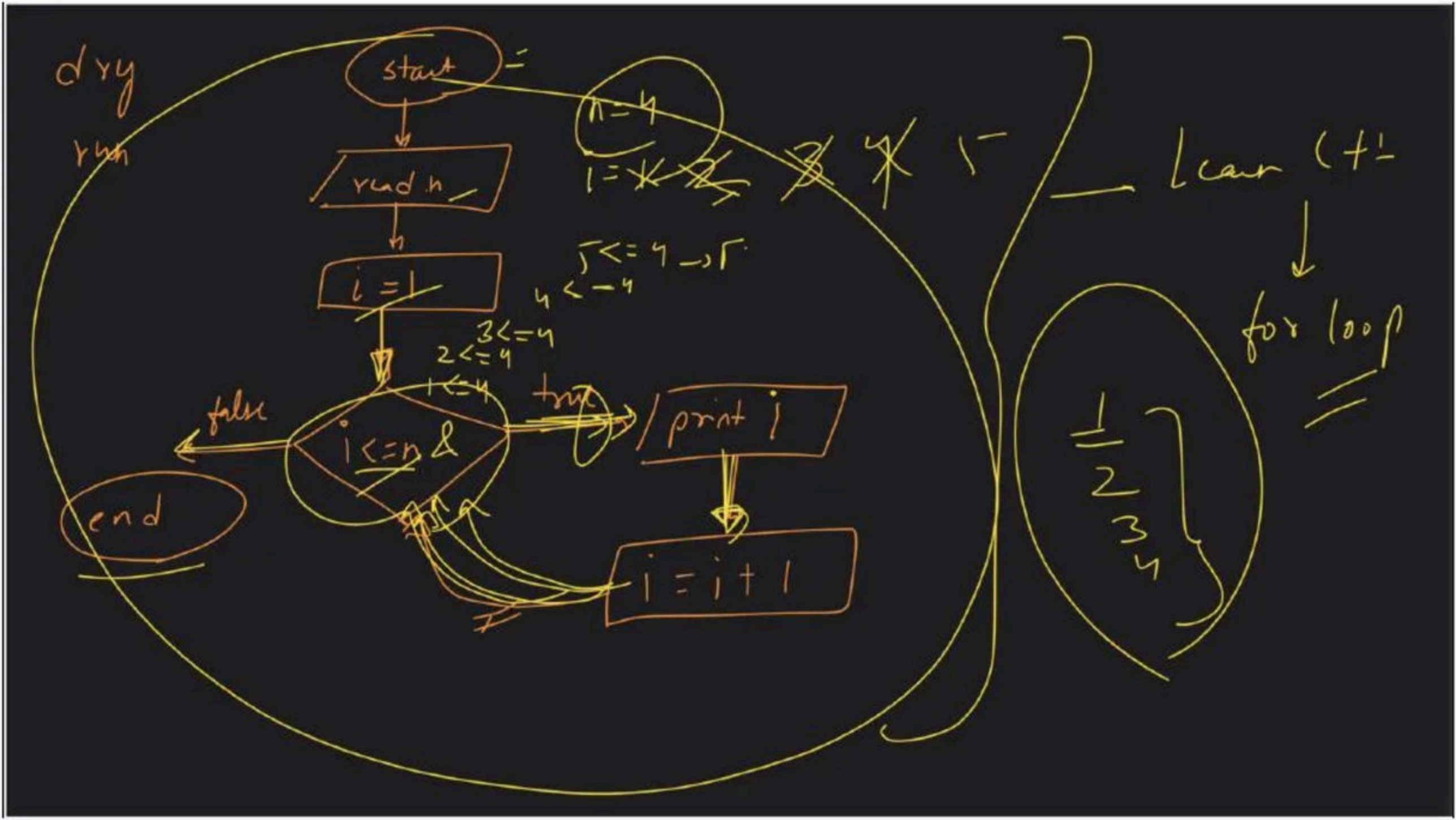


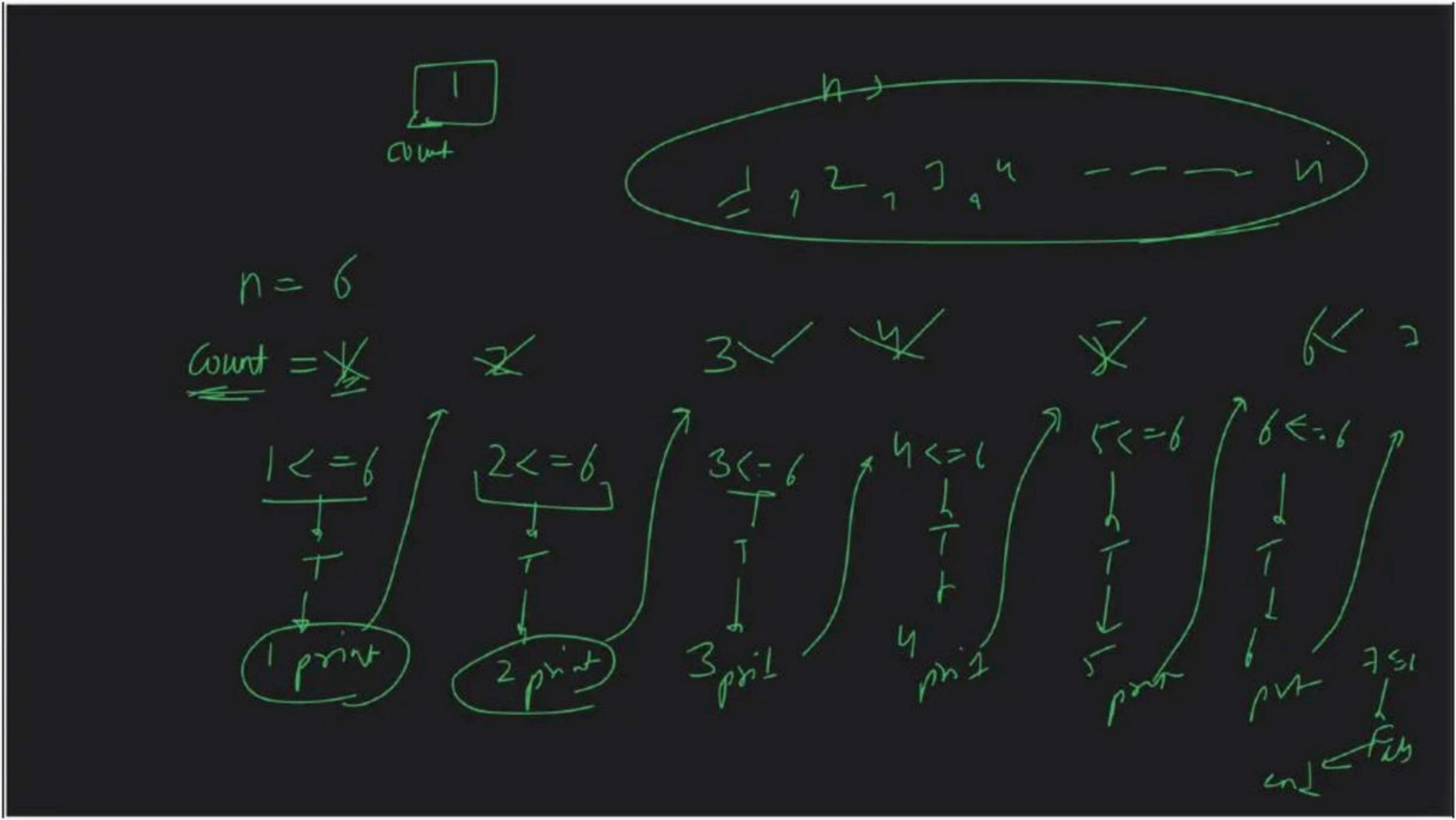

start
read name , n
limit = n
limit != 0
false
true
print name
limit = limit - 1
end
Jmp
loop1
name = Vansh
n = 4
limit = 4 3 2 1 0
4! = 0
True
Vansh
3! = 0
T
Vansh
2! = 0
T
Vansh
false
0! = 0
Vansh
1! = 0
Vansh

Design Flowchart - Print counting from 1 to N

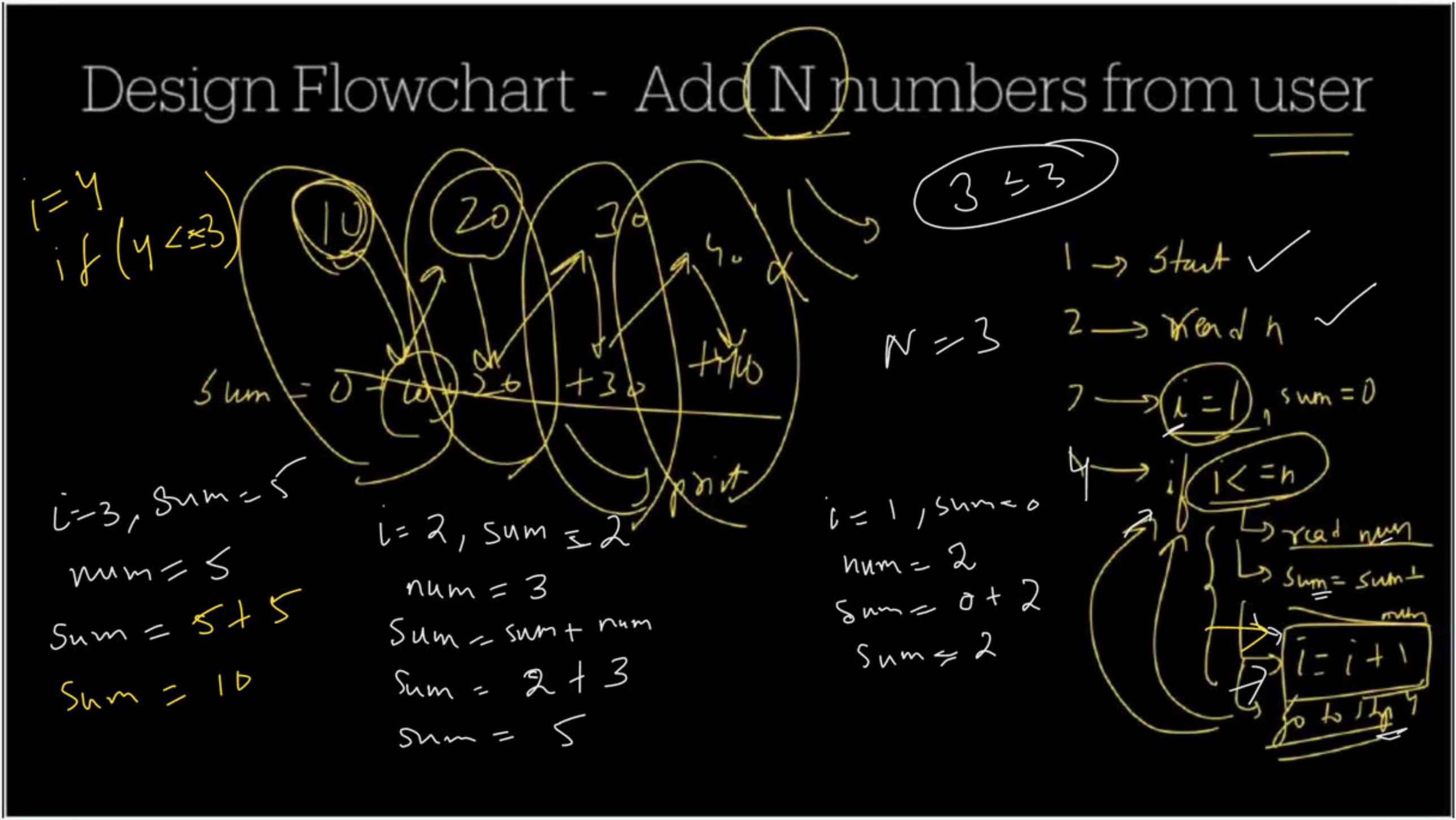
Loops

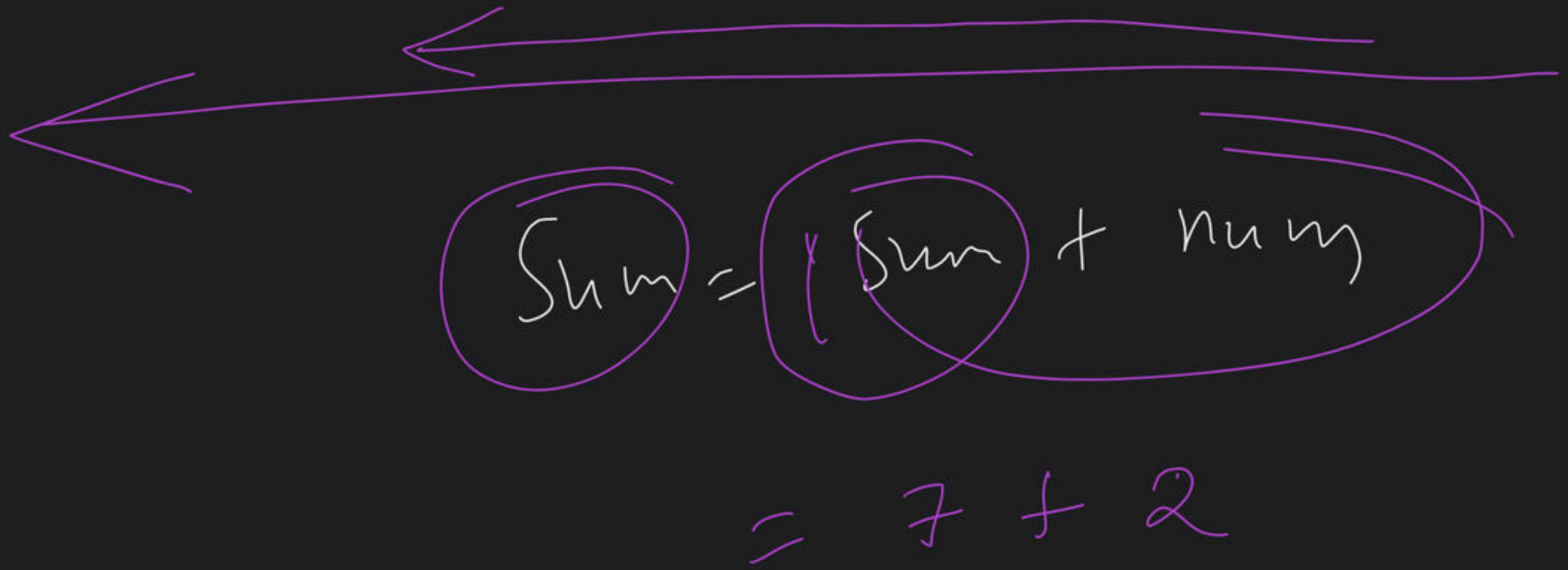



start
read n
i = 1
i <= n &
true
false
print i
i = i + 1
end
for loop


1
count
n
1, 2, 3, 4 ---- n
n = 6
count = 1 2 3 4 5 6
1<=6
T
1 print
2<=6
T
2 print
3<=6
T
T
3 print
4<=6
T
T
T
4 print
5<=6
T
T
T
T
print
6<=6
T
T
T
T
T
print
7<=6
F
end

Design Flowchart - Add N numbers from user



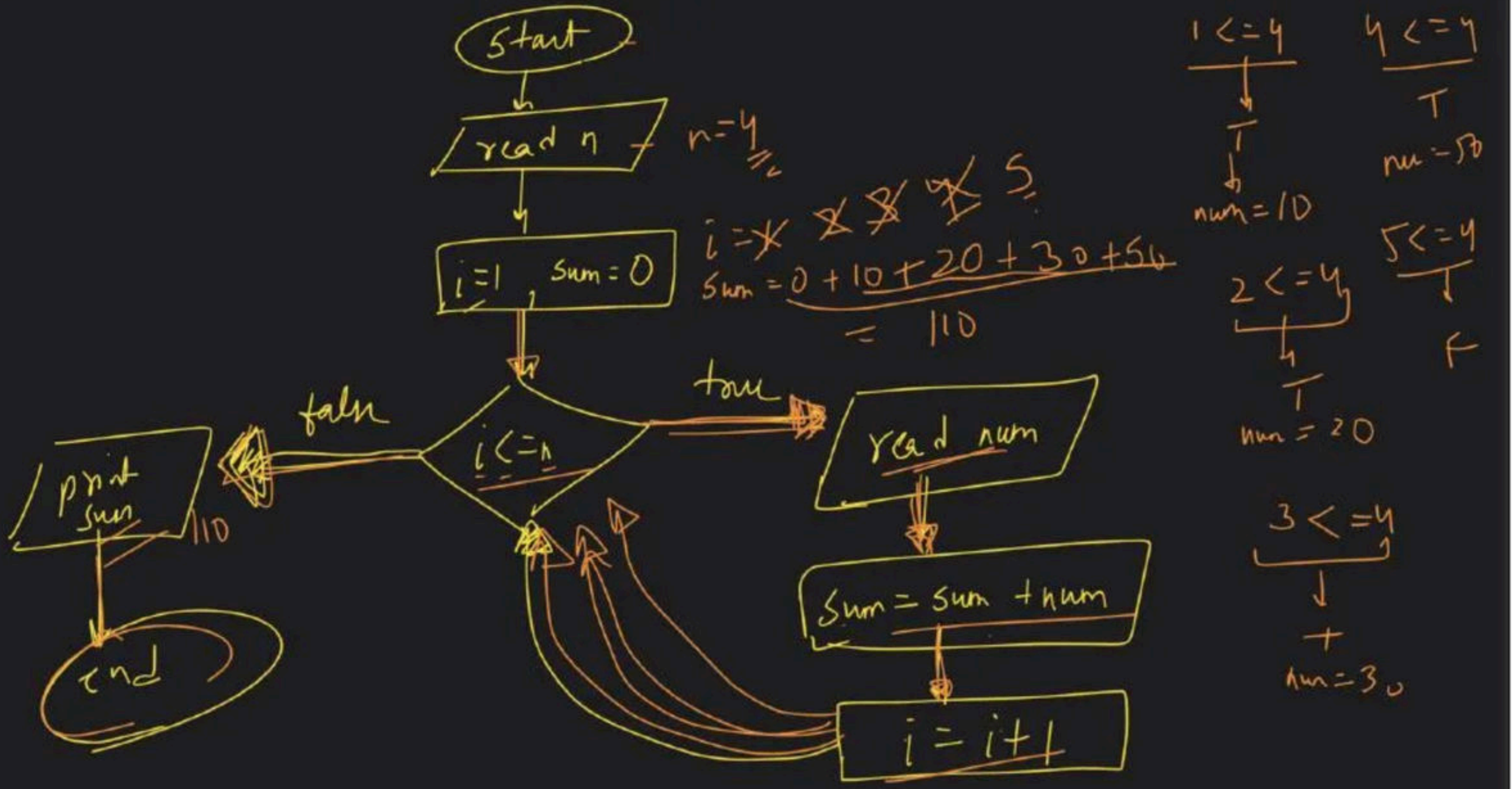

sum = sum + num
= 7 + 2

Sum = 5

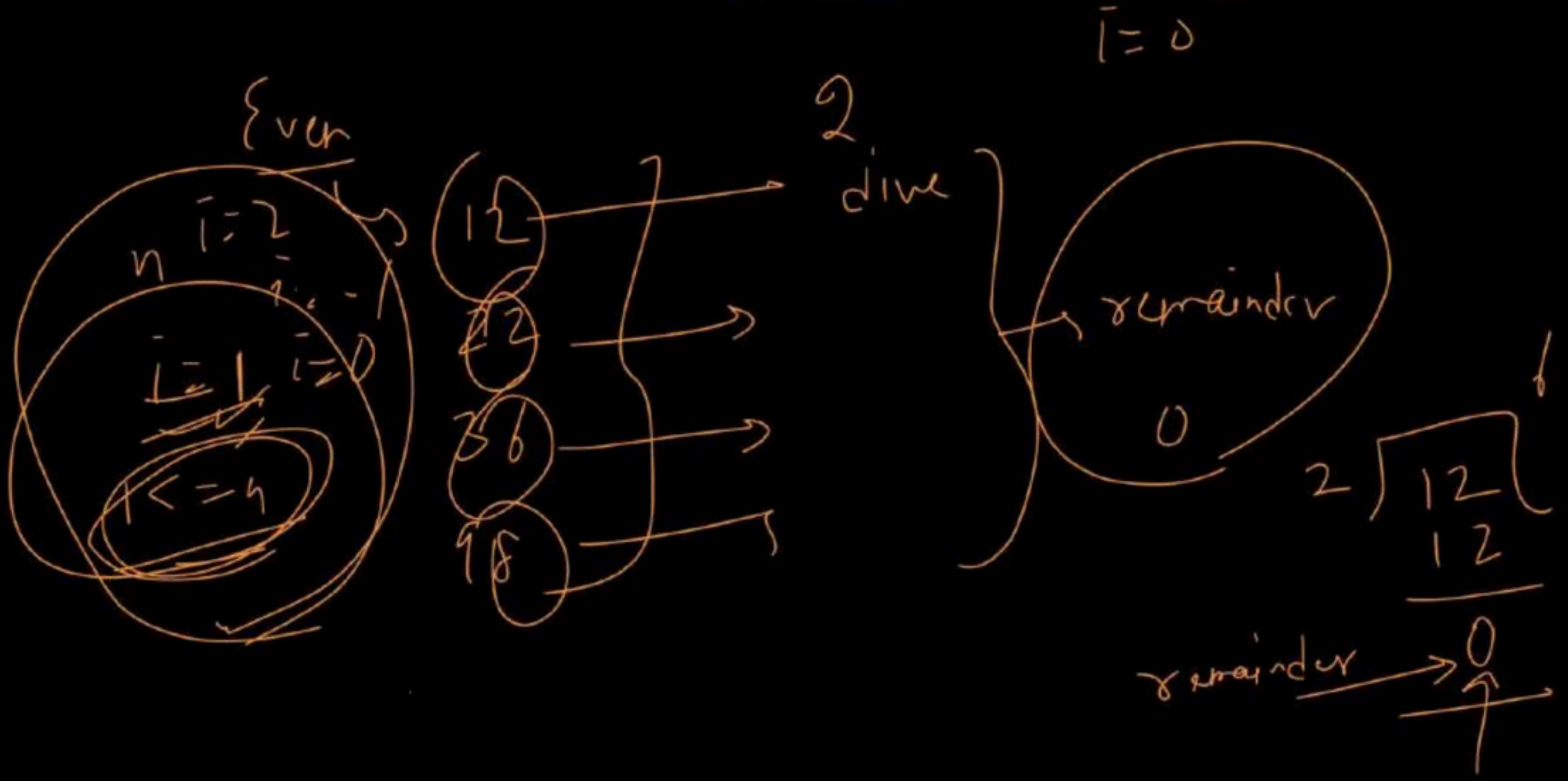
Sum = Sum + 10

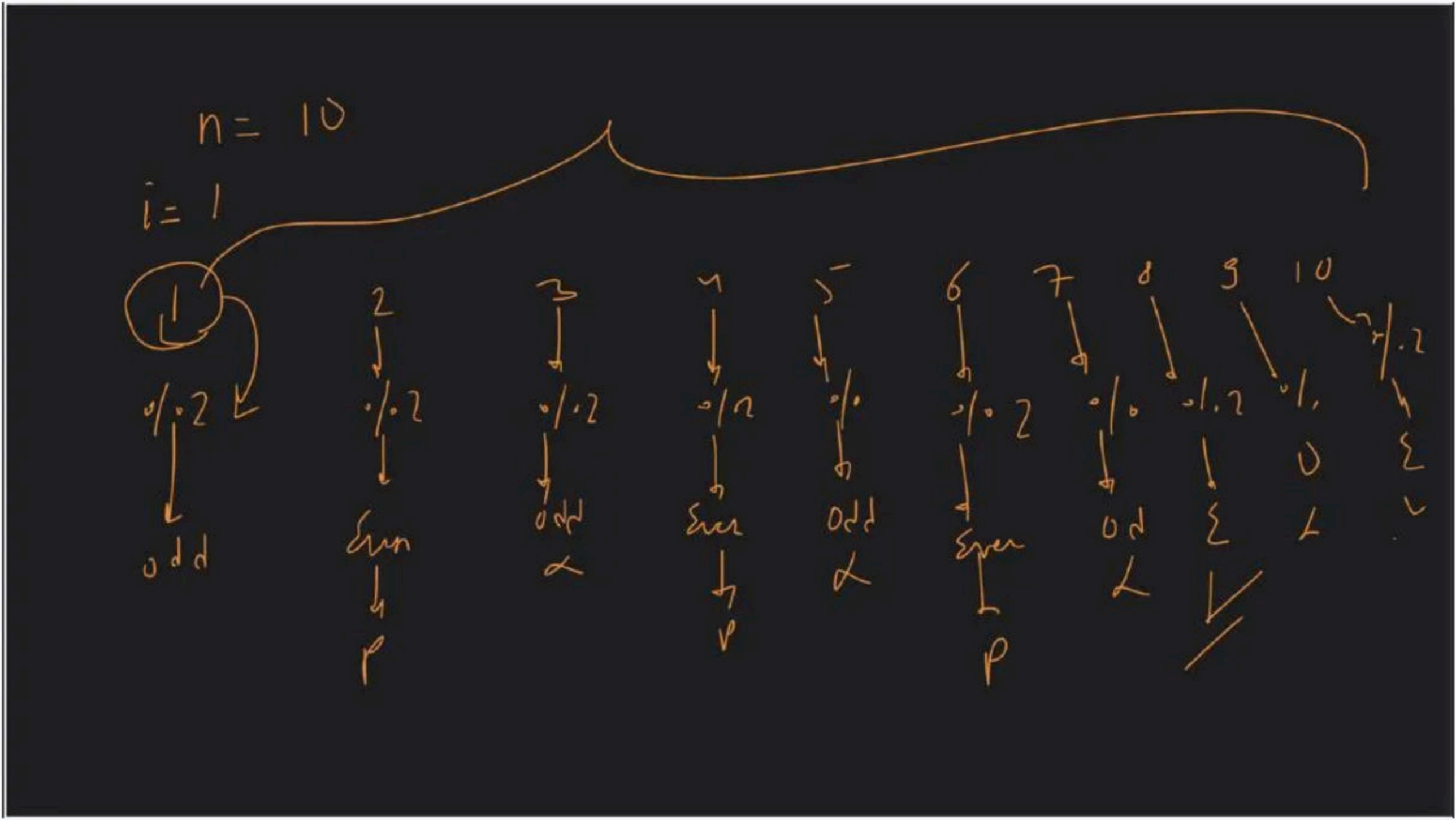
Sum = 5 + 10

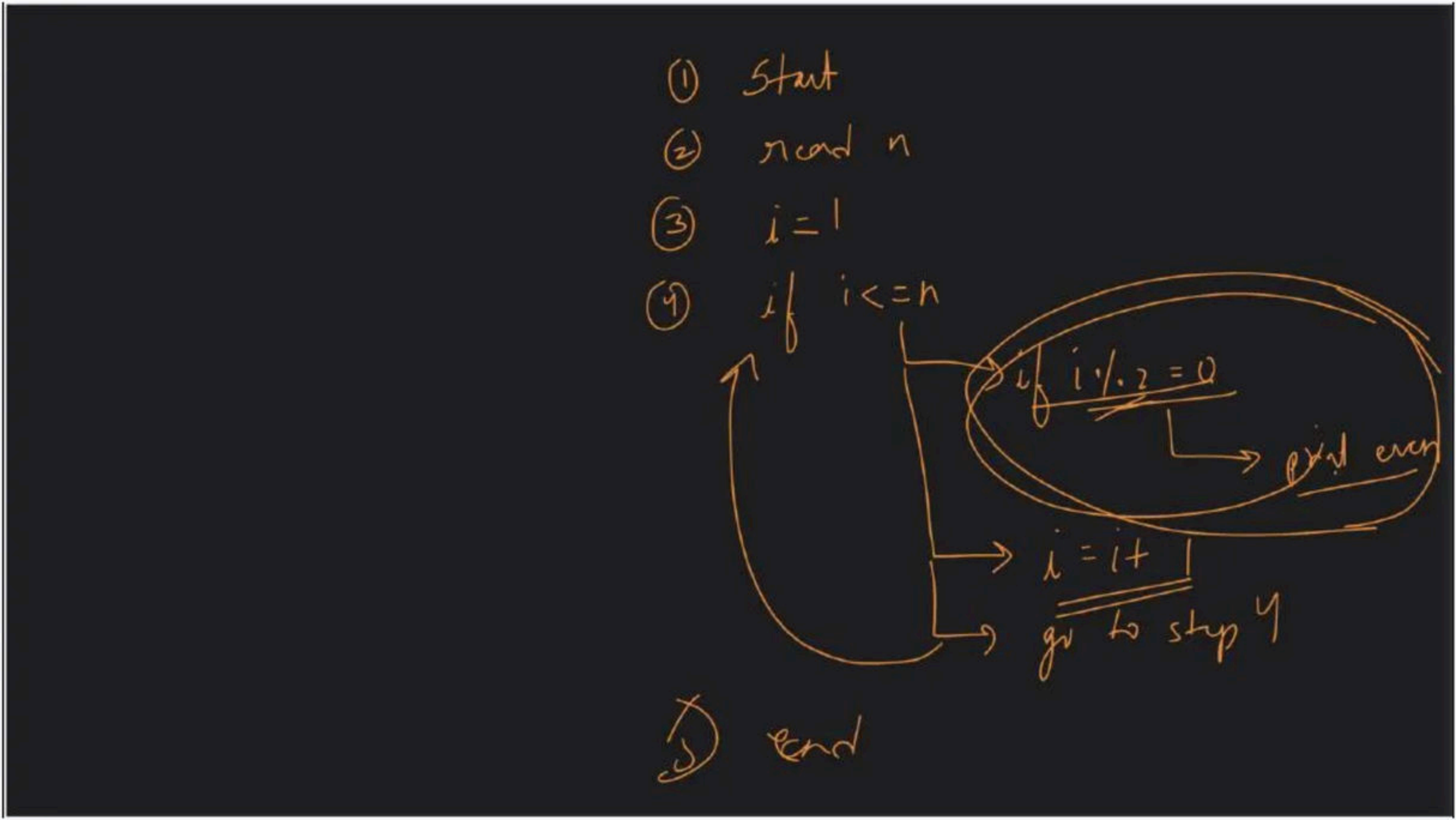
Sum = 15

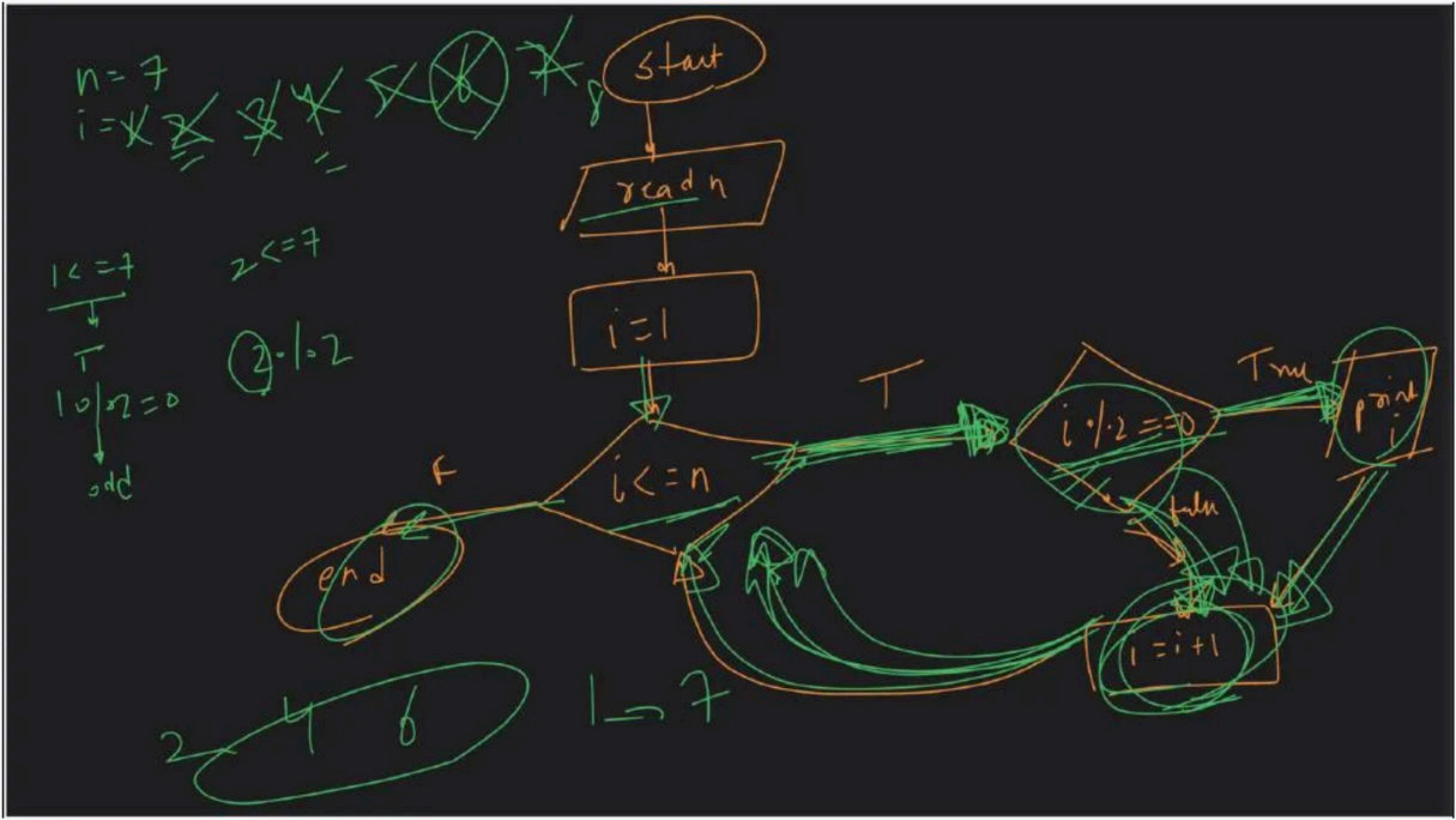

start
read n
n=4
i=1 sum=0
i<=n
false
true
print sum
110
end
read num
sum = sum + num
i = i+1
i=1 2 3 4 5
sum = 0+10+20+30+50
= 110
1<=4
T
num=10
2<=4
T
num=20
3<=4
T
num=30
4<=4
T
5<=4
F

Design Flowchart - Print Even numbers from 1 to N




n = 10
i = 1
1
1
1 % 2
odd
2
2 % 2
Even
3
3 % 2
odd
4
4 % 2
Even
5
5 % 2
odd
6
6 % 2
Even
7
7 % 2
odd
8
8 % 2
Even
9
9 % 2
odd
10
10 % 2
Even


1 Start
2 read n
3 i = 1
4 if i<=n
if i%2 = 0
print even
i = i+1
go to step 4
5 end


n=7
i=
1<=7
T
i%2==0
2<=7
2.1.2
start
read n
i=1
i<=n
T
F
i%2==0
True
print i
false
i=i+1
end
2 4 6
1->7

10/2

12 %% 2

Modulo

Modulus

12 %% 2 = 0

rem

ans = 10 / 2 = 5

Rem = 10 % 2 = 0 remainder

num %% 2

→ 0 → Even

→ !0 → Odd

2) 10 (5

-10

0

Quo = 13 / 2

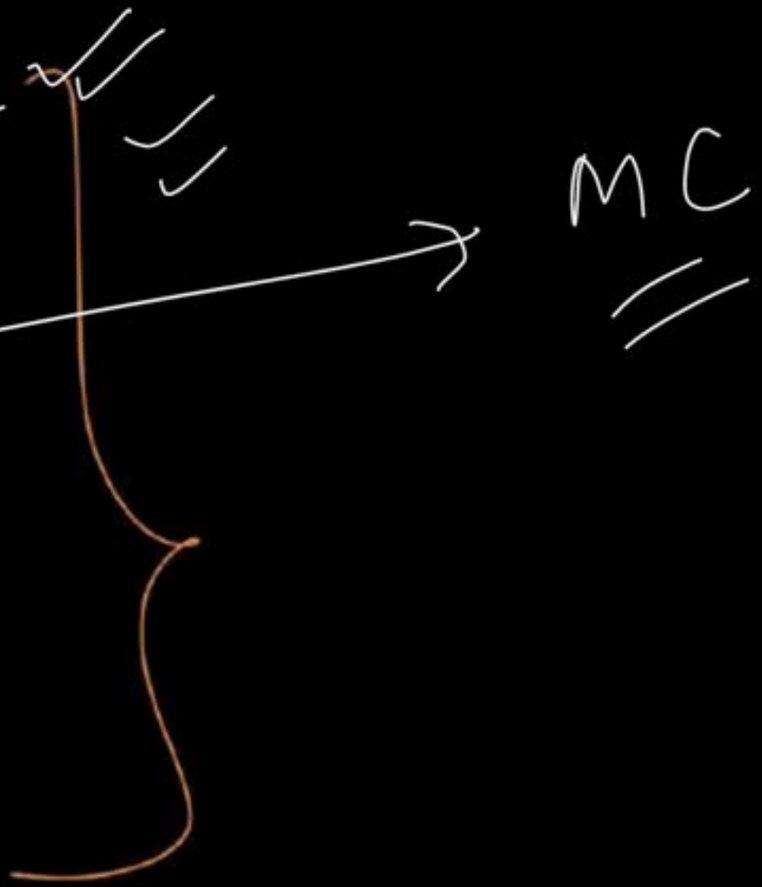
Rem = 13 % 2

⇒ dividend = divisor * Quotient + rem.

Assignment:

Design Flowchart for below:

- • Multiply 2 number after taking input
- • Perimeter of triangle
- • Find Simple Interest
- • Find Compound Interest
- • Print Counting from N to 1
- • Find Factorial of a Number
- • Check if a number is Prime or not
- • Check Valid Triangle or not
- • Print Max of 2 number



L2-Basics of Programming

Special class

Love Babbar • Apr 3, 2025

L2 - Basics of Programming

- Variables and Operators
- Write your 1st Program

—> by Codehelp

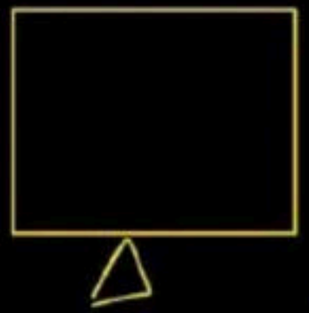
Programming Language ?

What ?

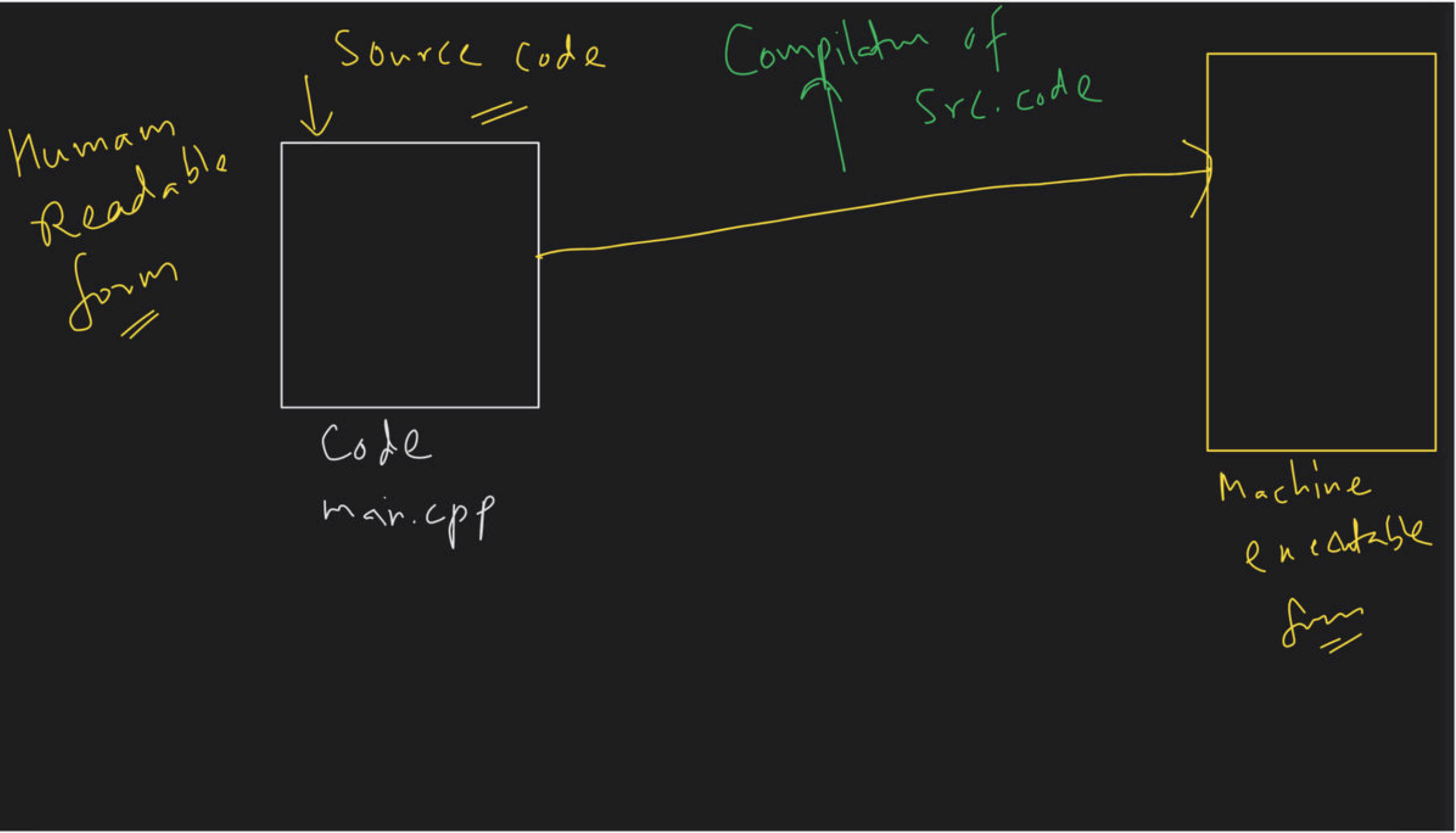
Why ?

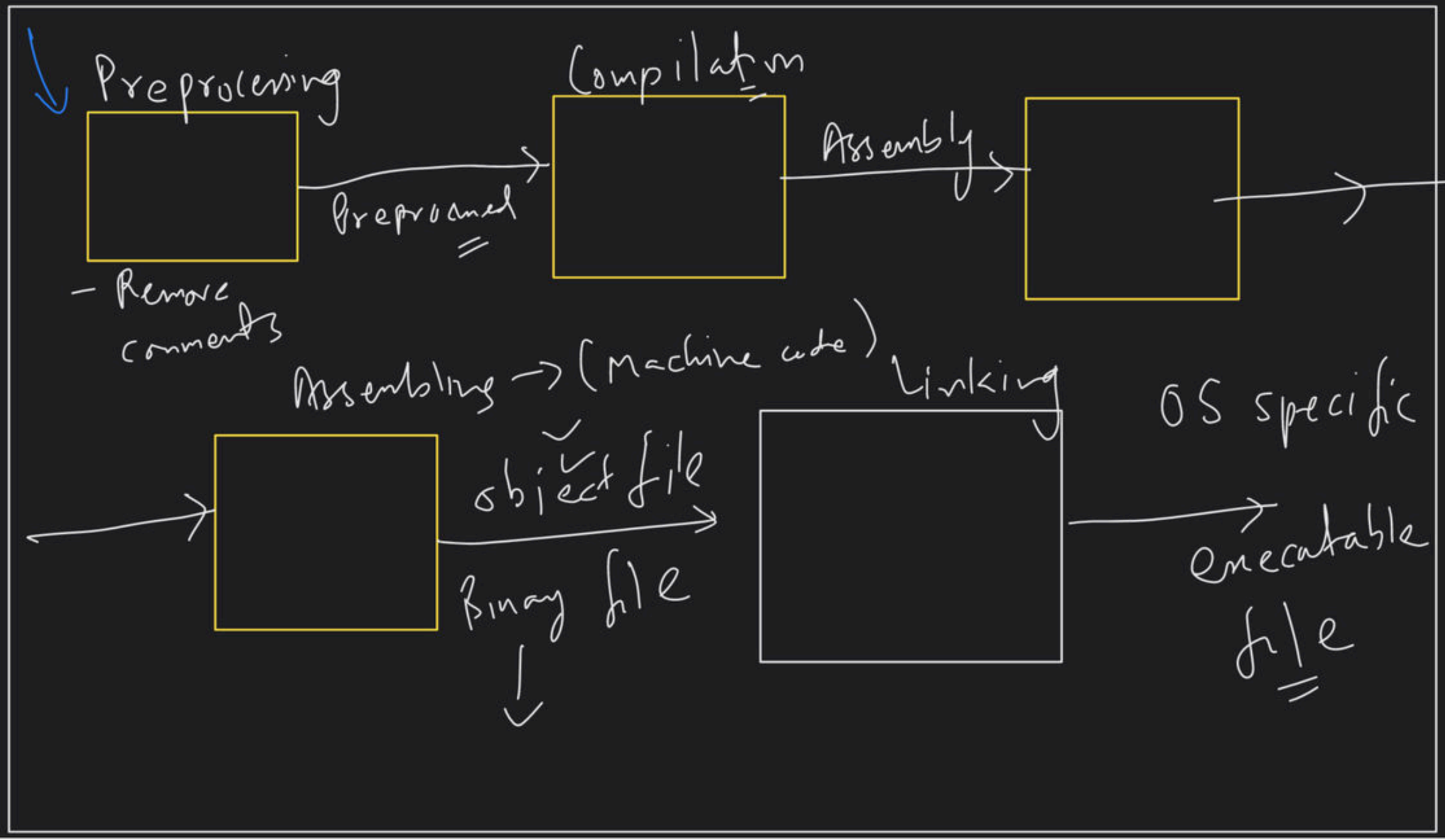
Standard

Programming language

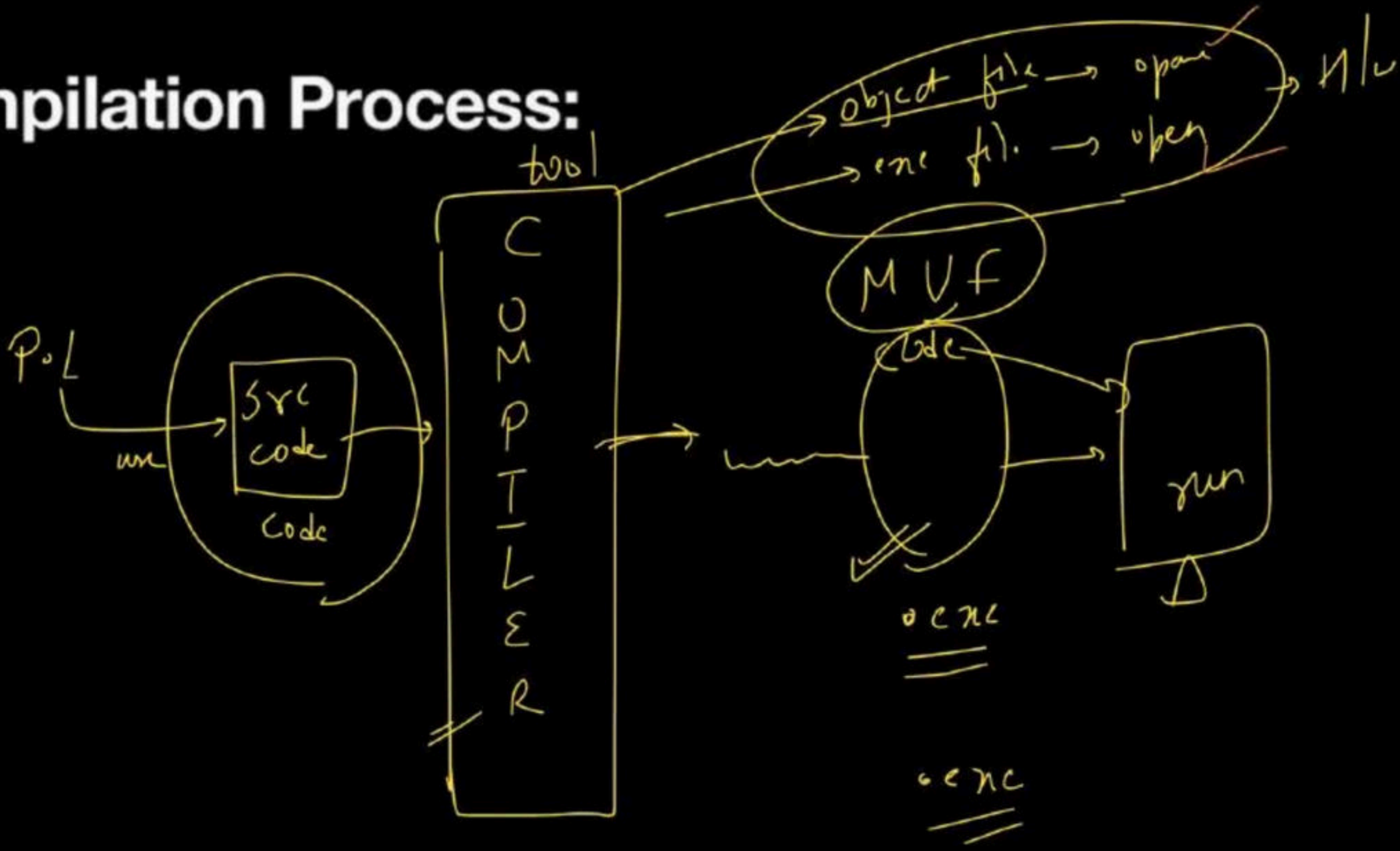


task

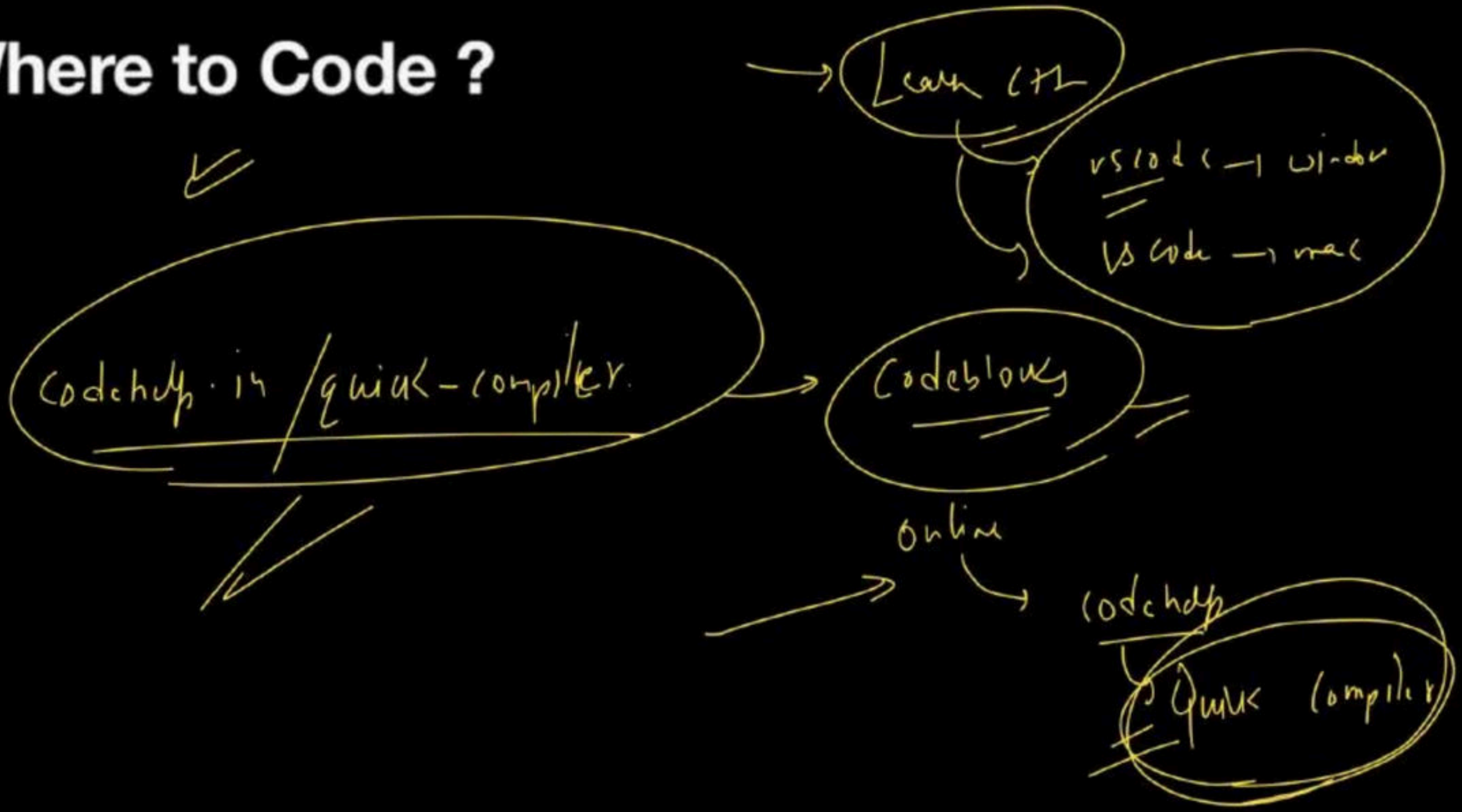

Source code
Compilation of Src. code
Human Readable form
Code
main.cpp
Machine executable form


Preprocessing
Compilation
Assembly
Preprocessed
- Remove comments
Assembling → (machine code)
object file
Binary file
Linking
OS specific
executable file

Compilation Process:



Where to Code ?



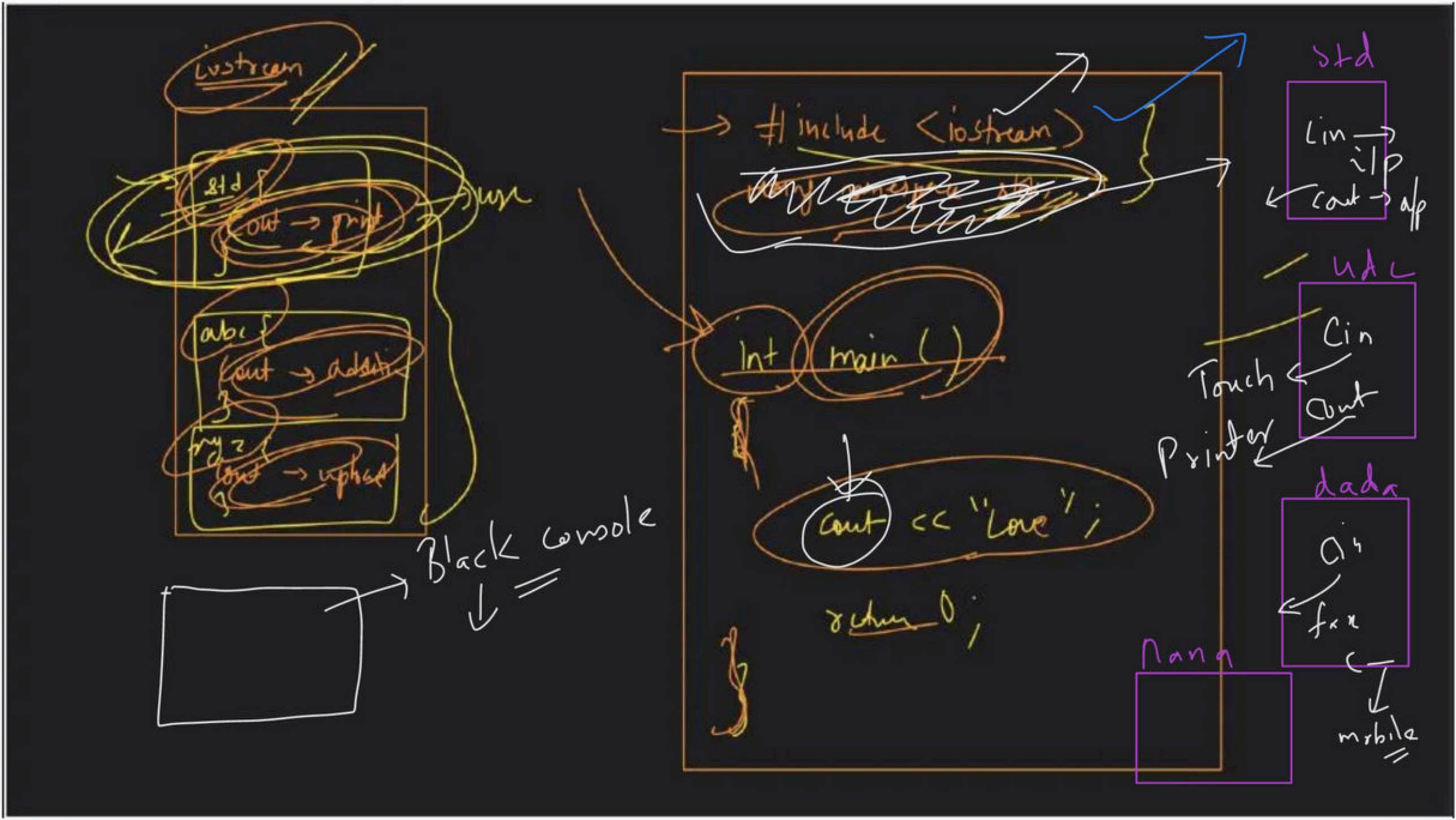
Your 1st Code:

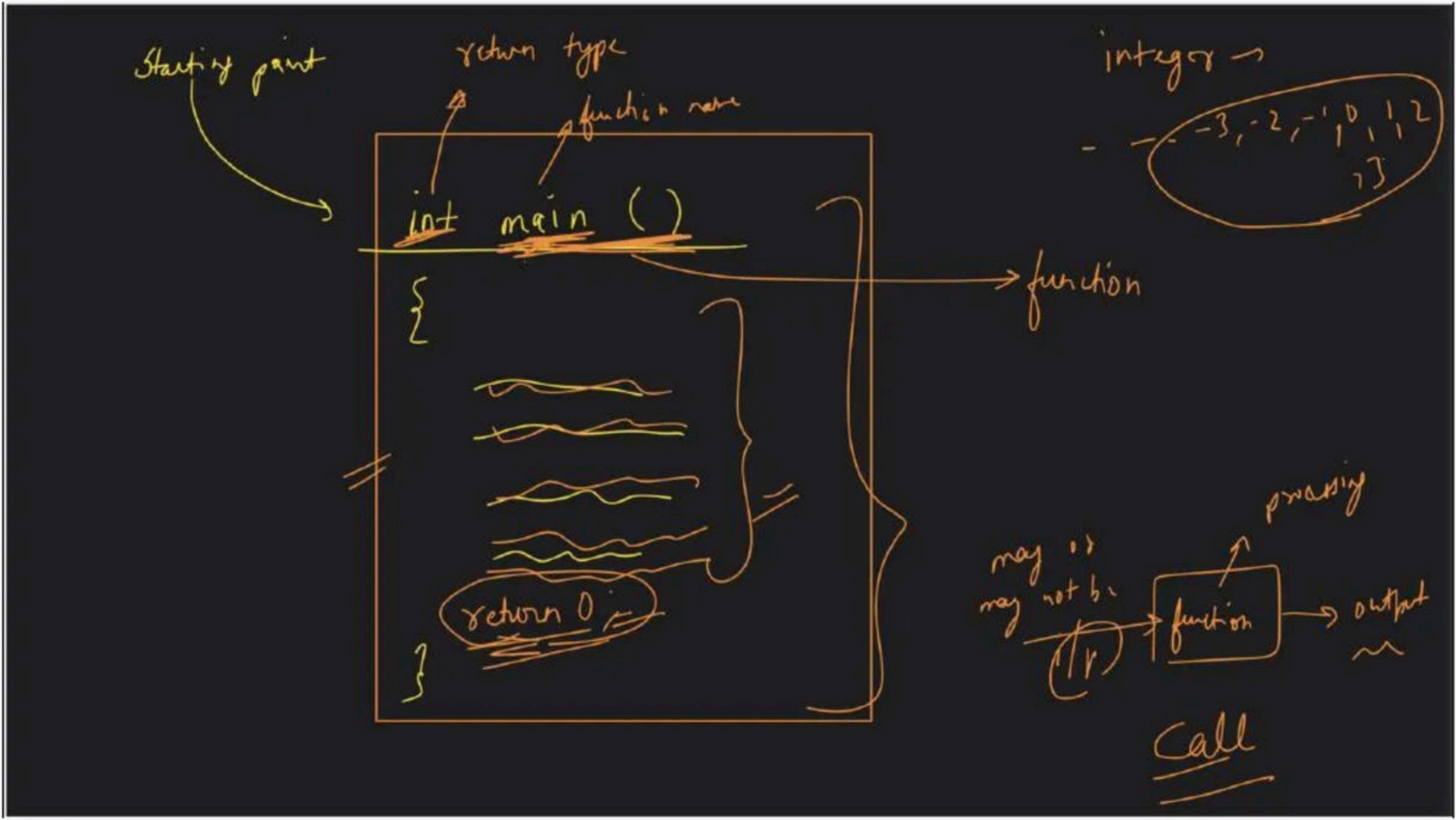
C++

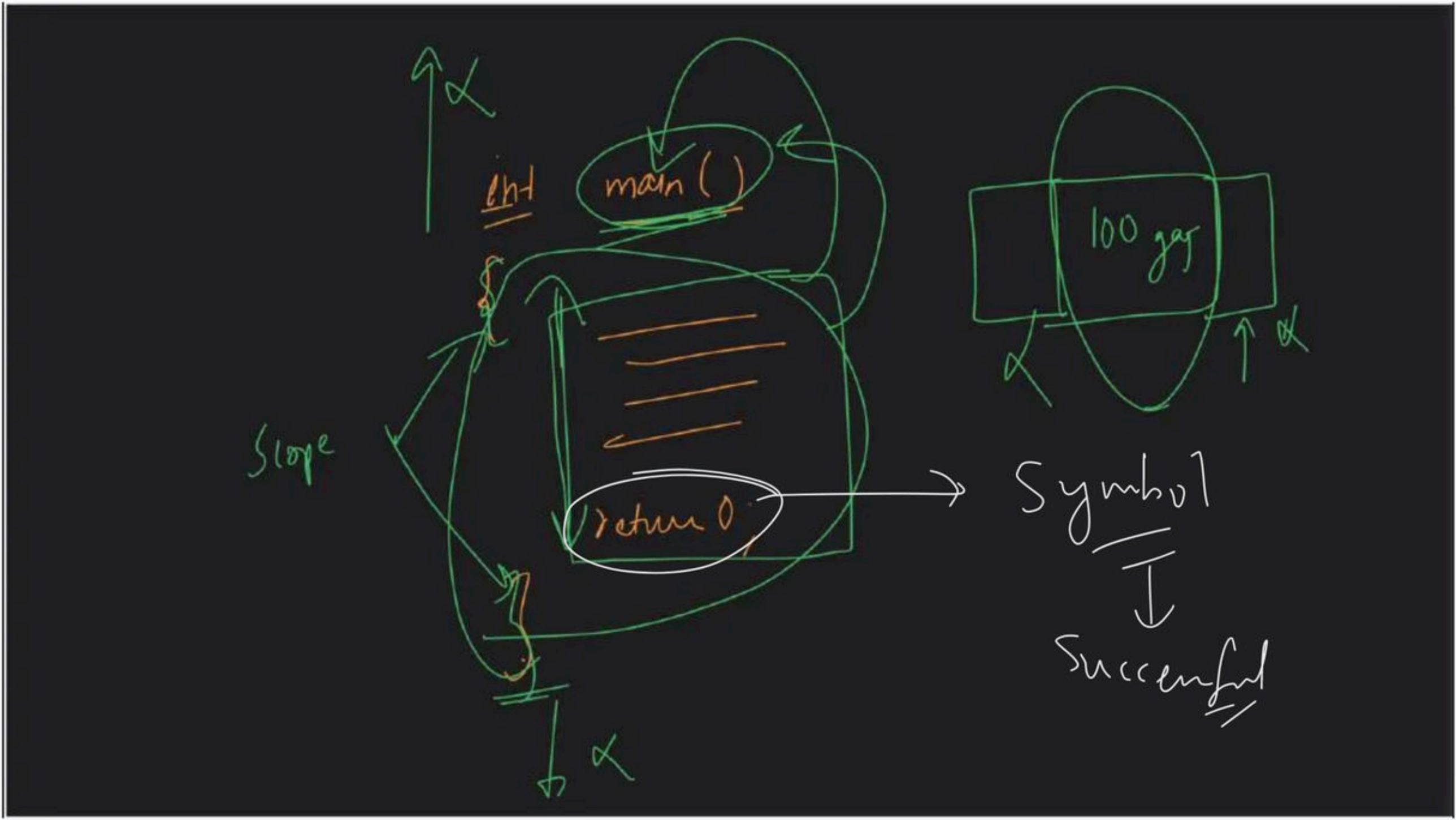
starting point

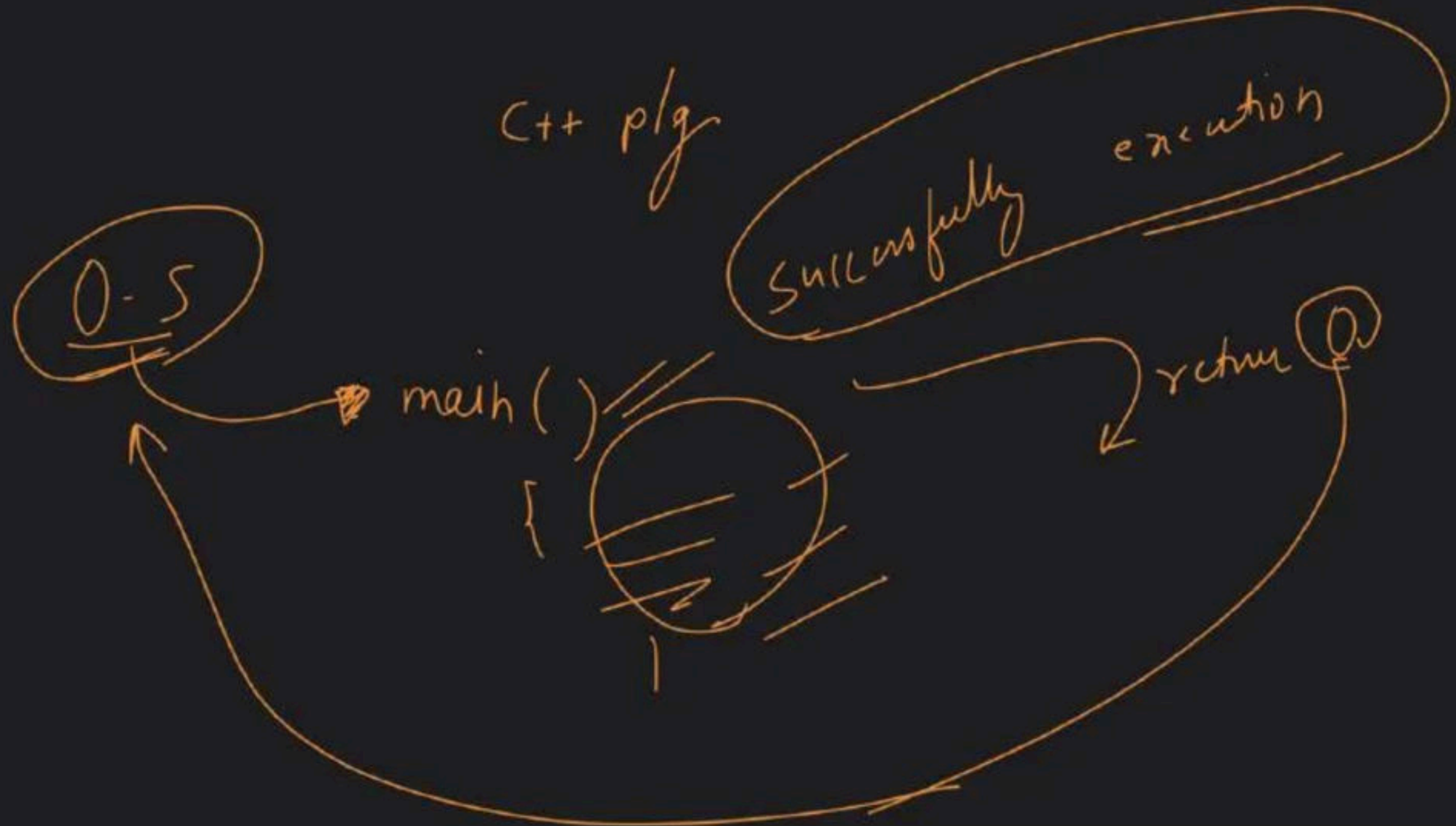
int main

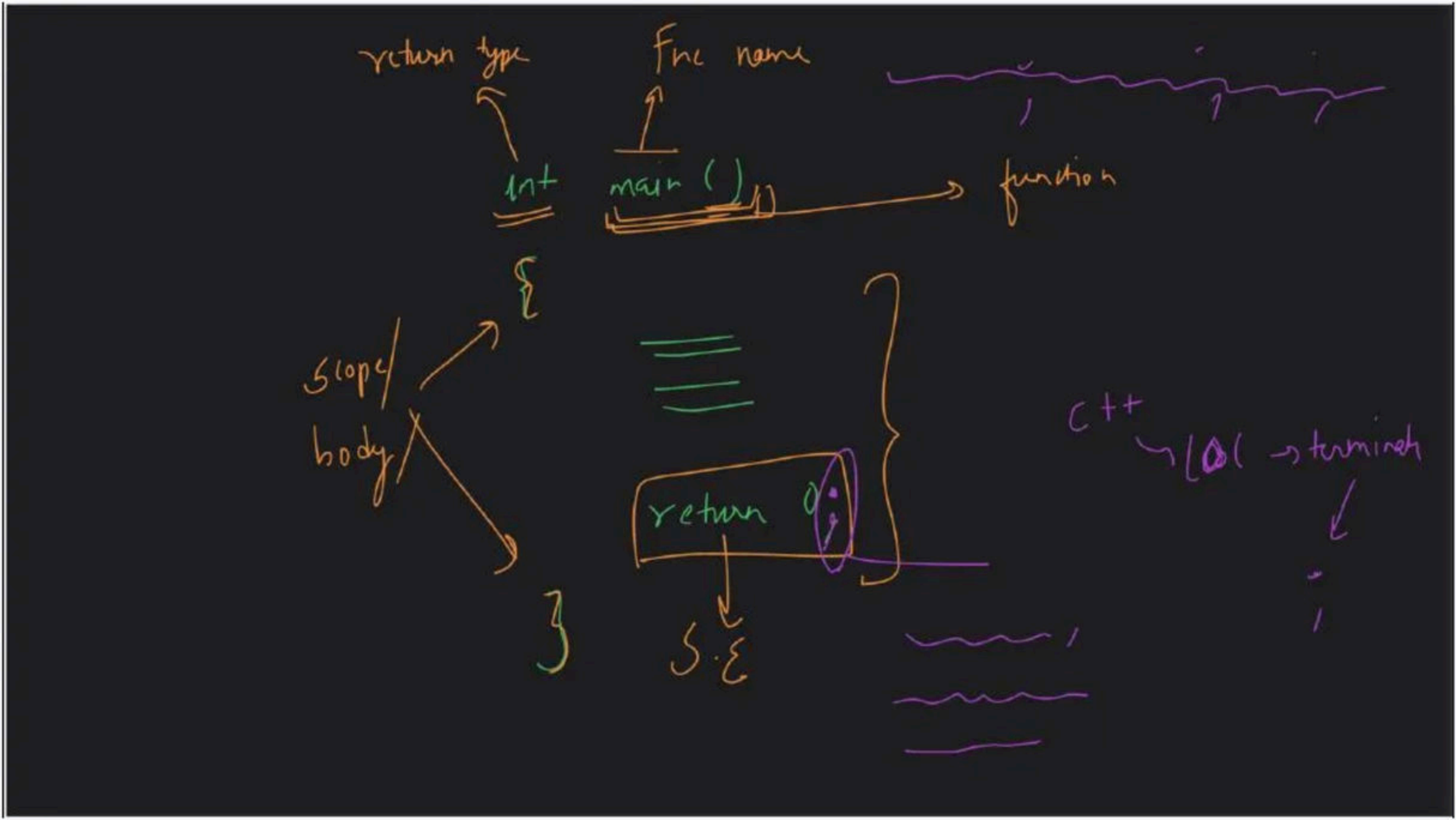
Starting point

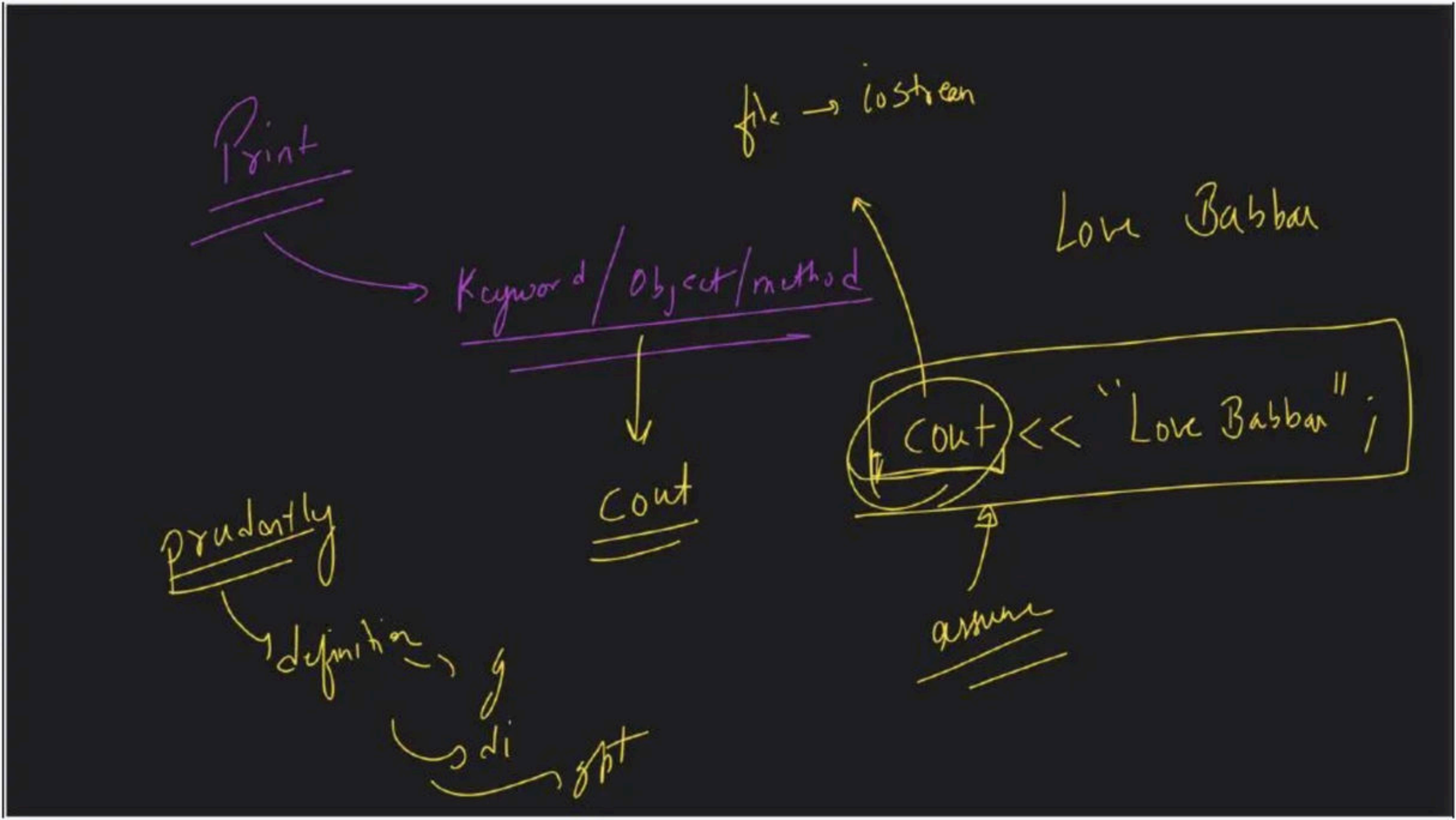

iostream
std
cout → print
abc
cout → address
xyz
cout → alphabet
Black console
#include <iostream>
int main ()
cout << "Love";
return 0;
std
Cin → i/p
cout → o/p
UAL
Cin
Touch
Printer
Count
dada
Ai
face
mobile
Nana

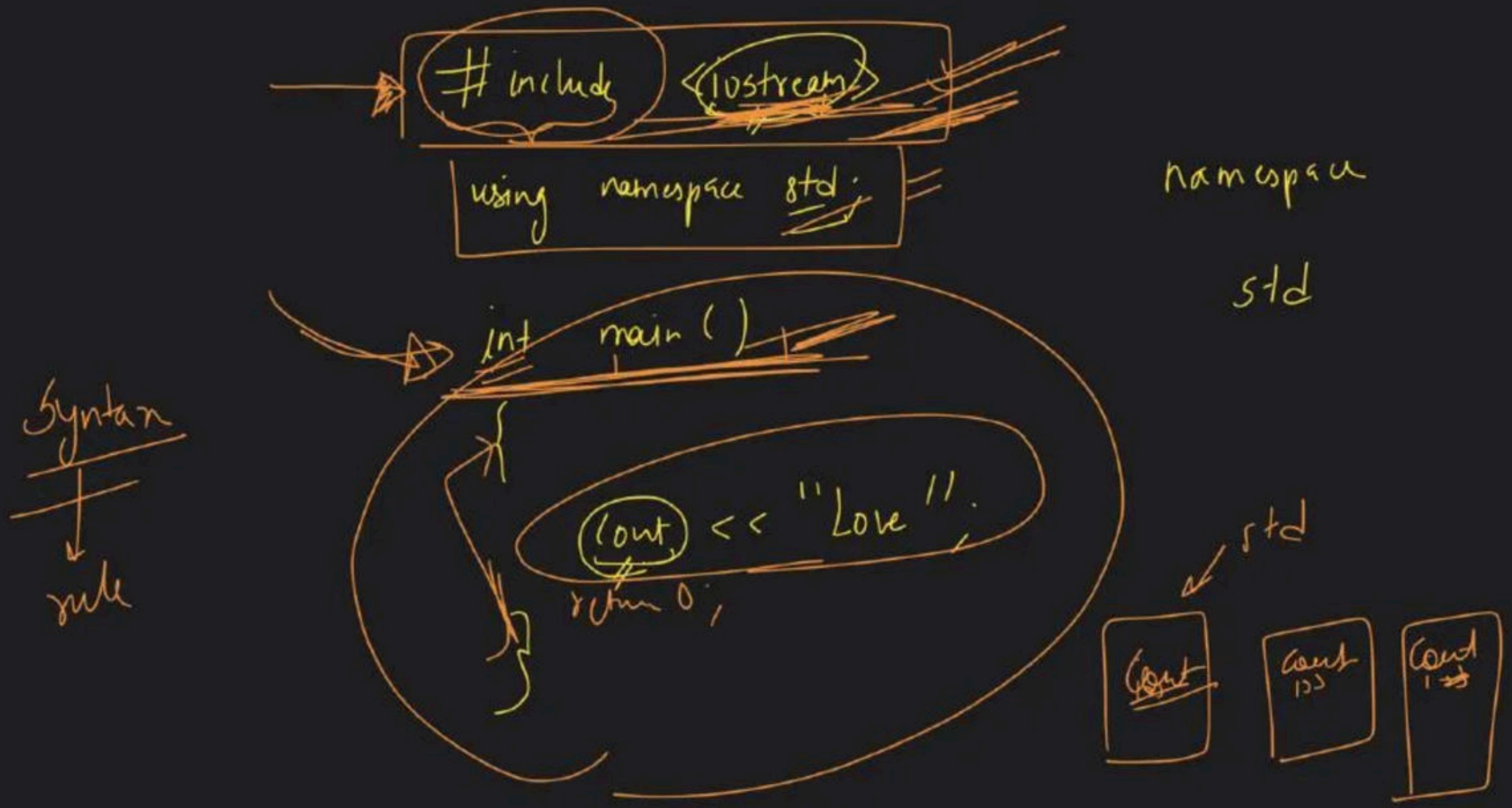

Starting point
return type
function name
integer →
-3, -2, -1, 0, 1, 2, 3
int main ()
{
return 0;
}
function
passing
may or may not be
function
output
Call

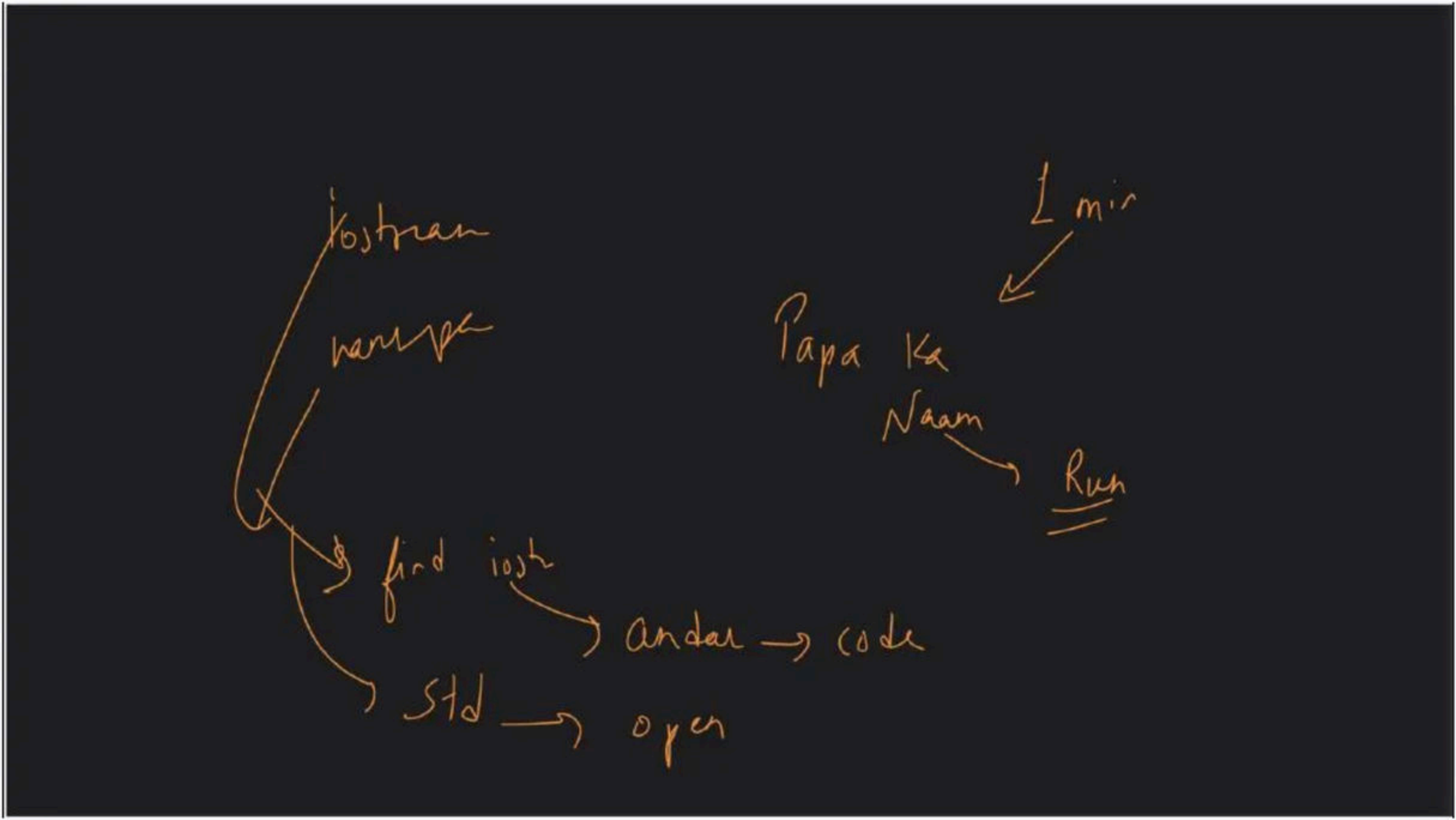

int
main()
{
return 0;
}
100 gaj
Slope
Symbol
Succesful

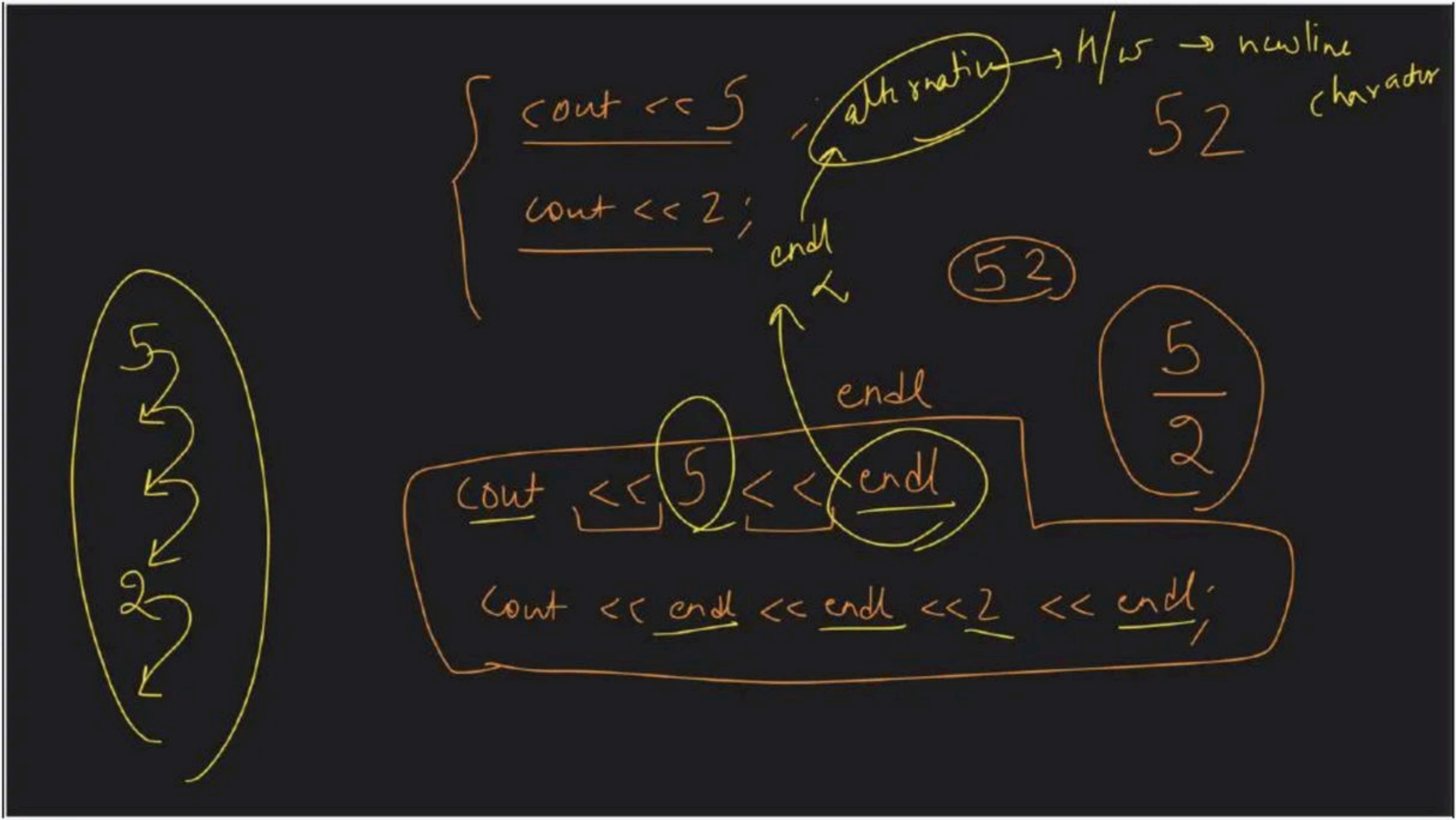

C++ p/g
O.S
main()
{
successfully execution
return 0;


return type
Fnc name
int
main ()
function
{
scope/
body
return 0;
}
S.E
c++
↳ 0 → terminate

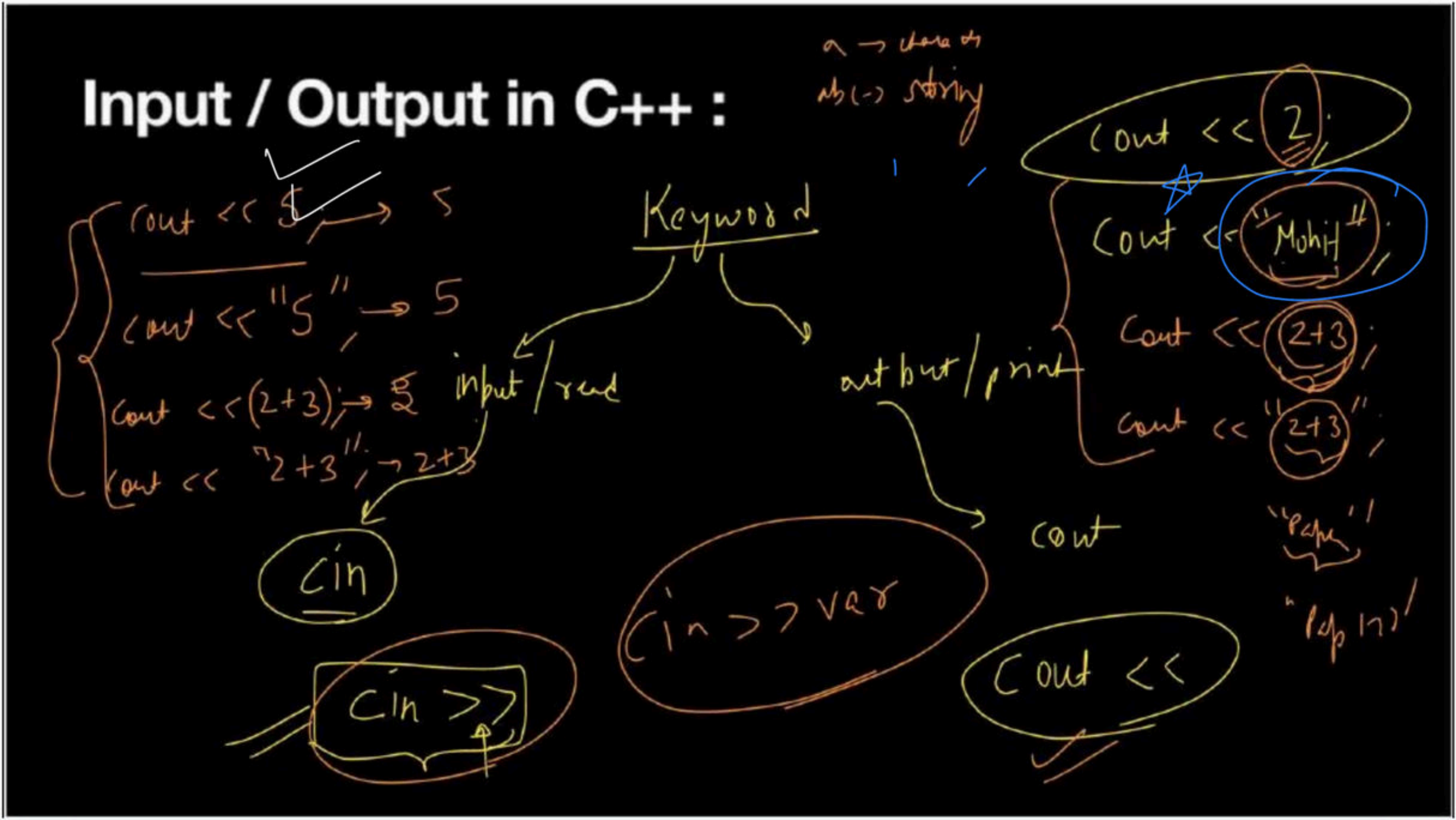

Print
file → iostream
Keyword / Object / method
Love Babbar
cout
cout << "Love Babbar" ;
Prudently
definition
gpt
assume

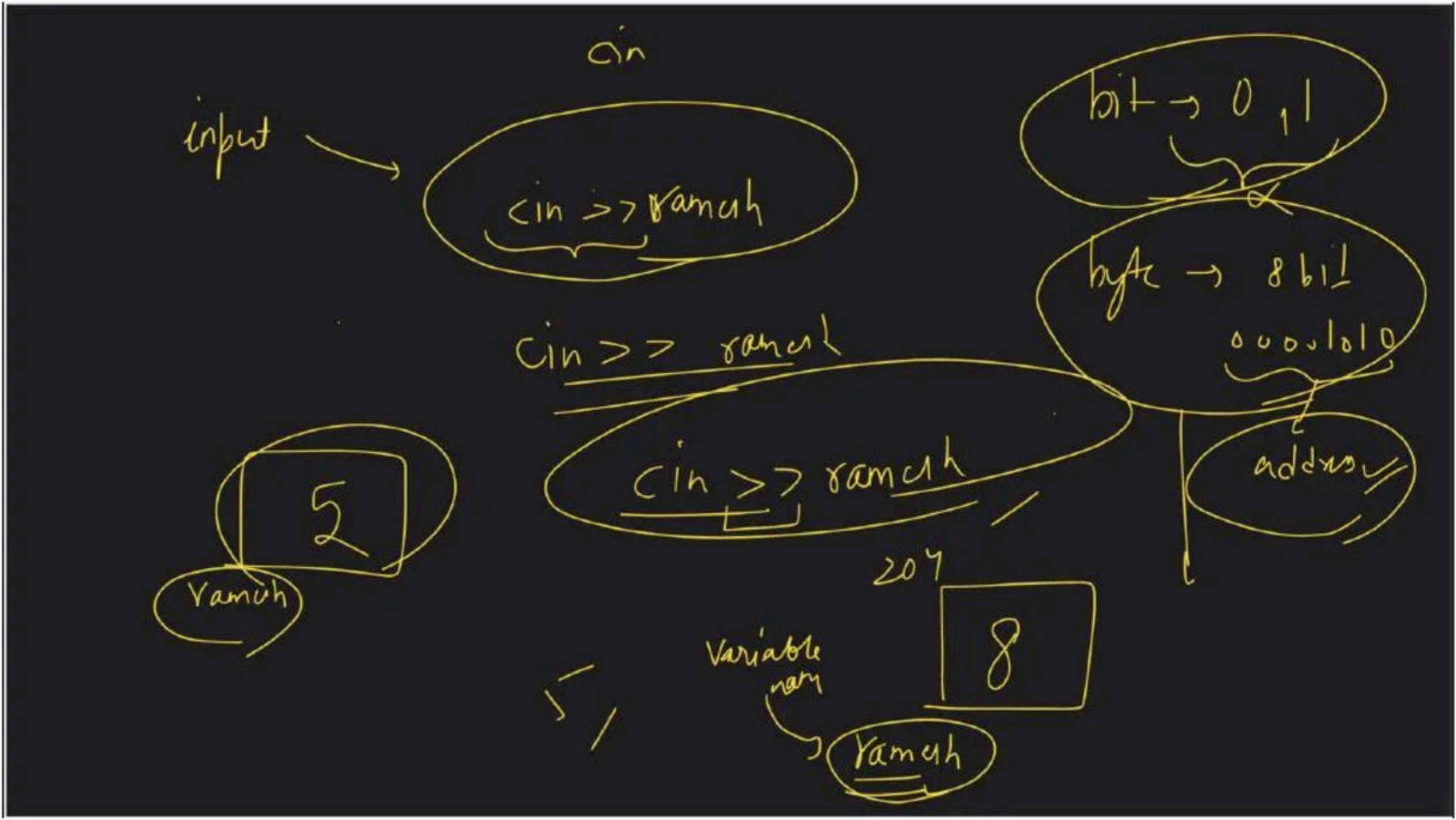

#include <iostream>
using namespace std;
int main()
cout << "Love";
return 0;
Syntax
rule
namespace
std
std
cout
cout
123
cout
1


iostream
namespace
Main
Papa ka
Naam
Run
find iosh
andar → code
std → open


cout << 5;
cout << 2;
alternative
\n/\r → newline character
52
endl
52
endl
cout << 5 << endl
5
2
5
5
2
2
cout << endl << endl << 2 << endl;

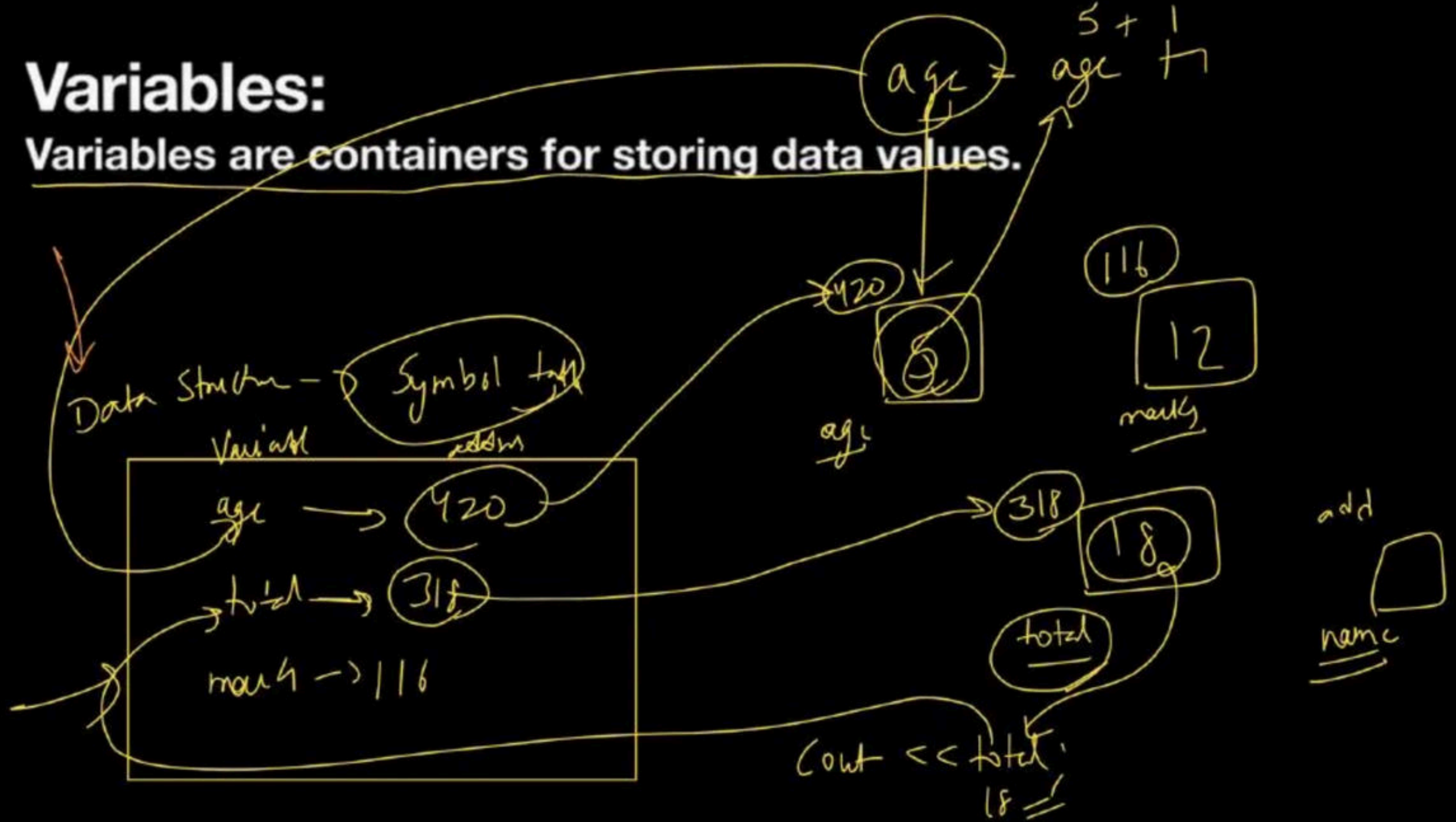
Input / Output in C++ :

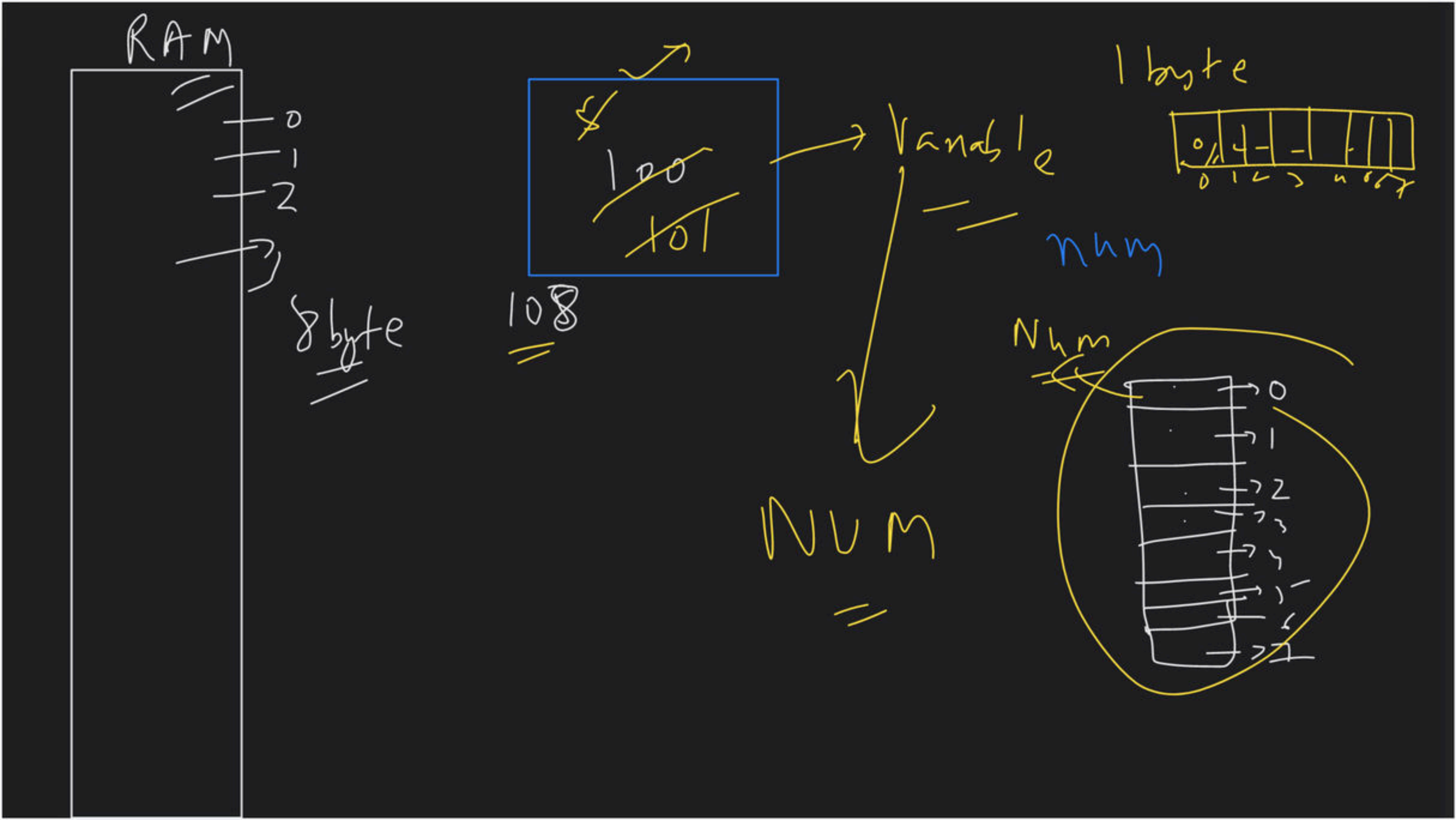


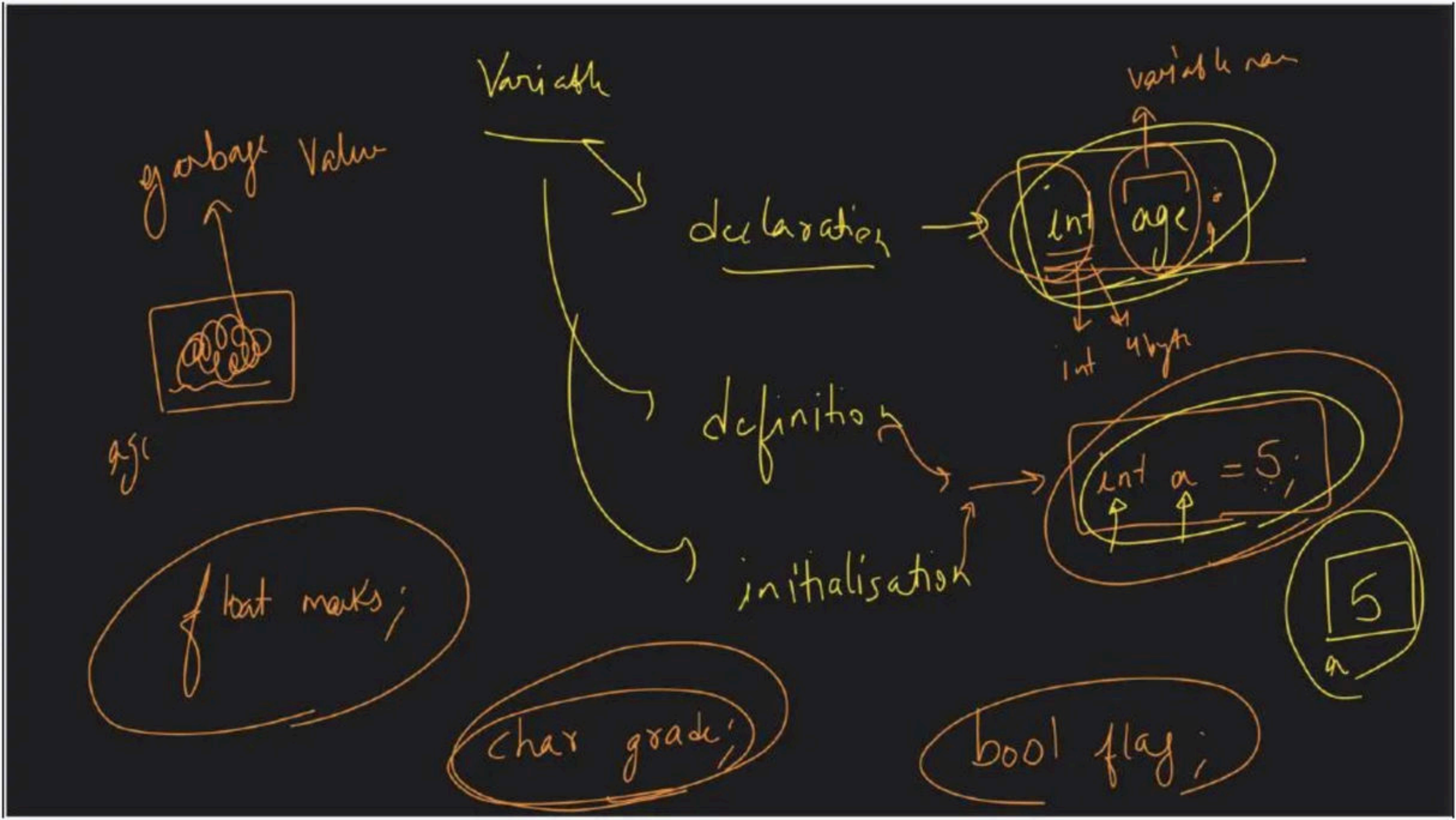

cin
input
cin >> ramesh
bit → 0 , 1
byte → 8 bit
00001010
address
cin >> ramesh
cin >> ramesh
5
Ramesh
204
8
Variable name
Ramesh

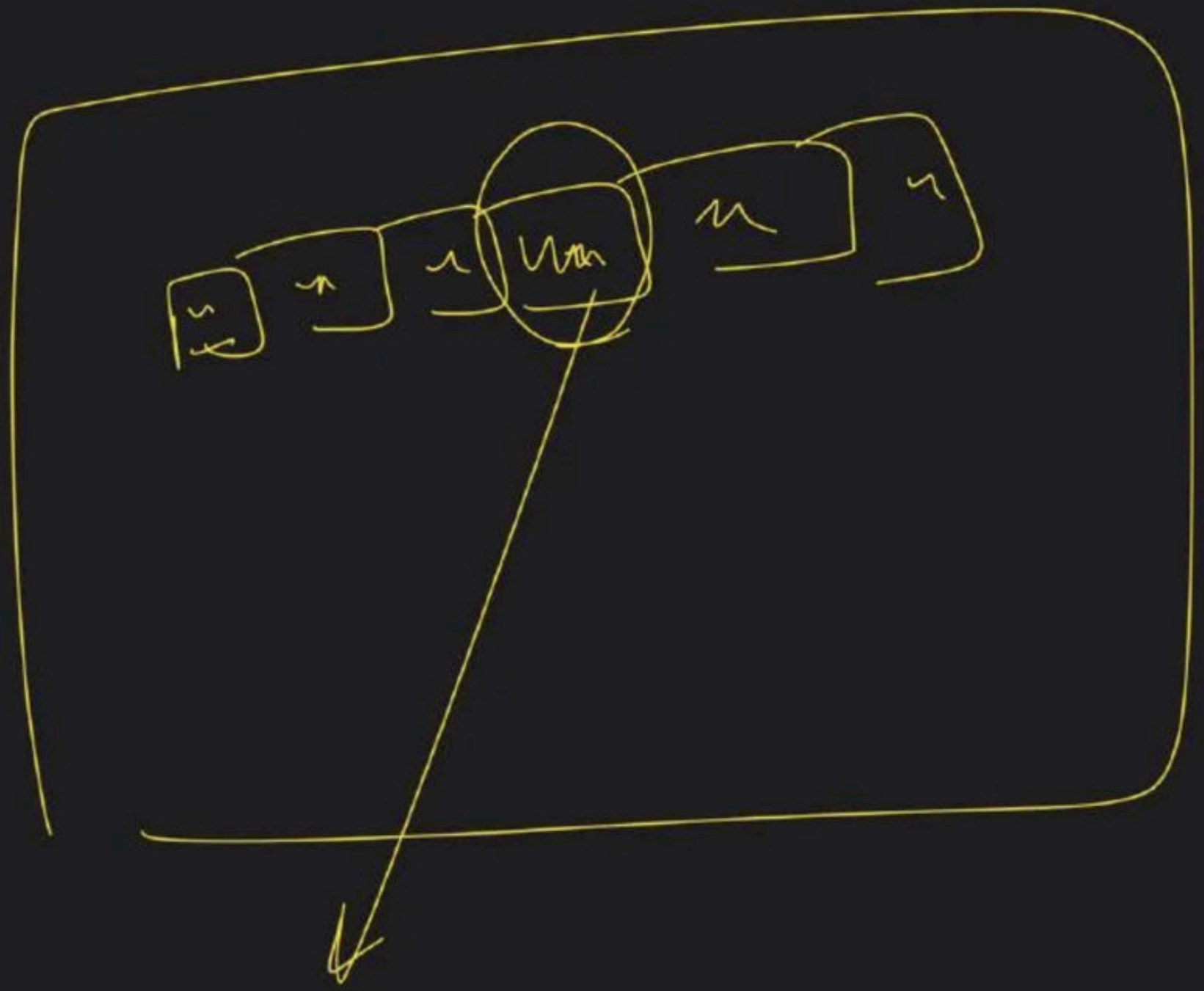
Variables:

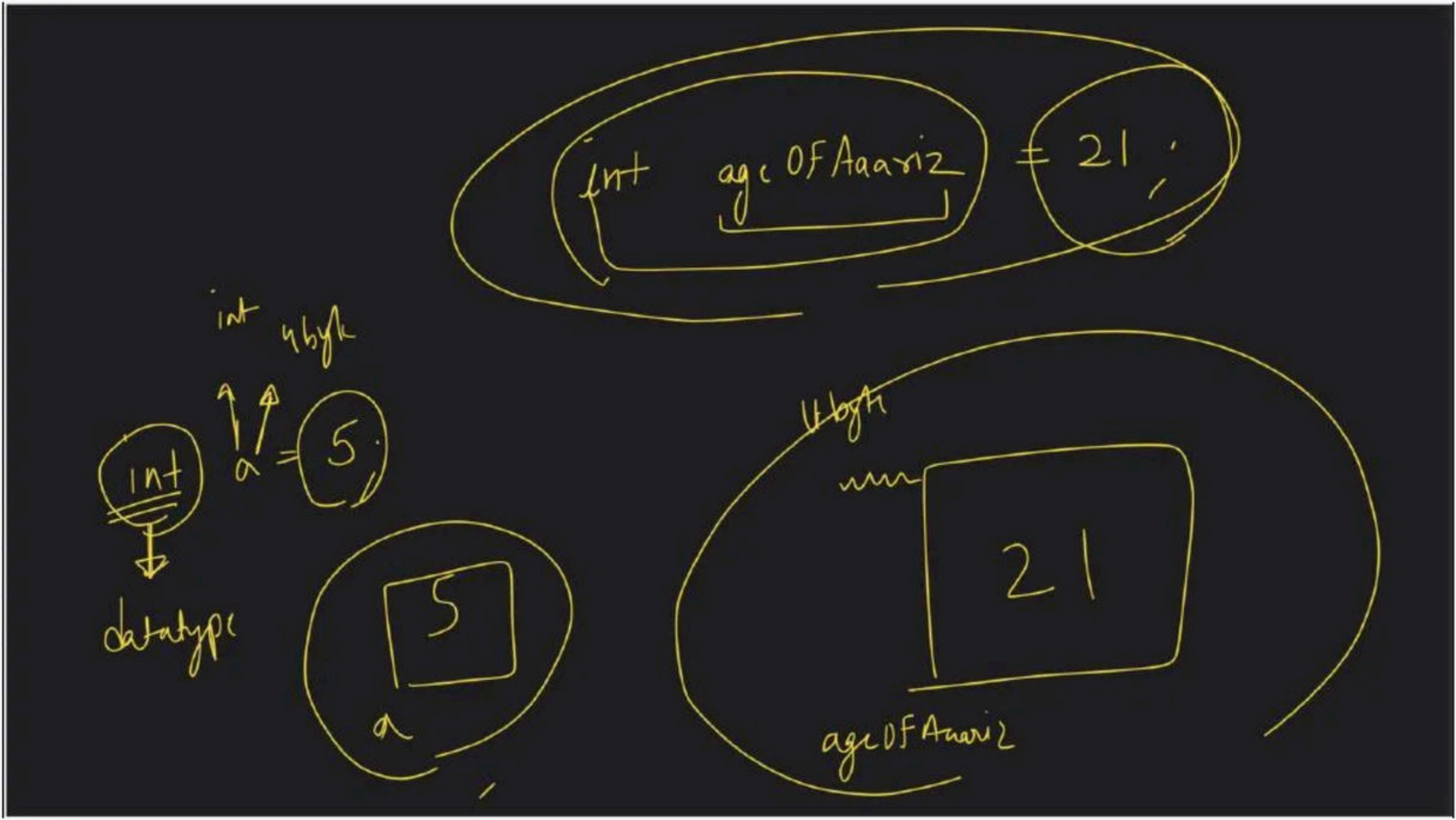
Variables are containers for storing data values.

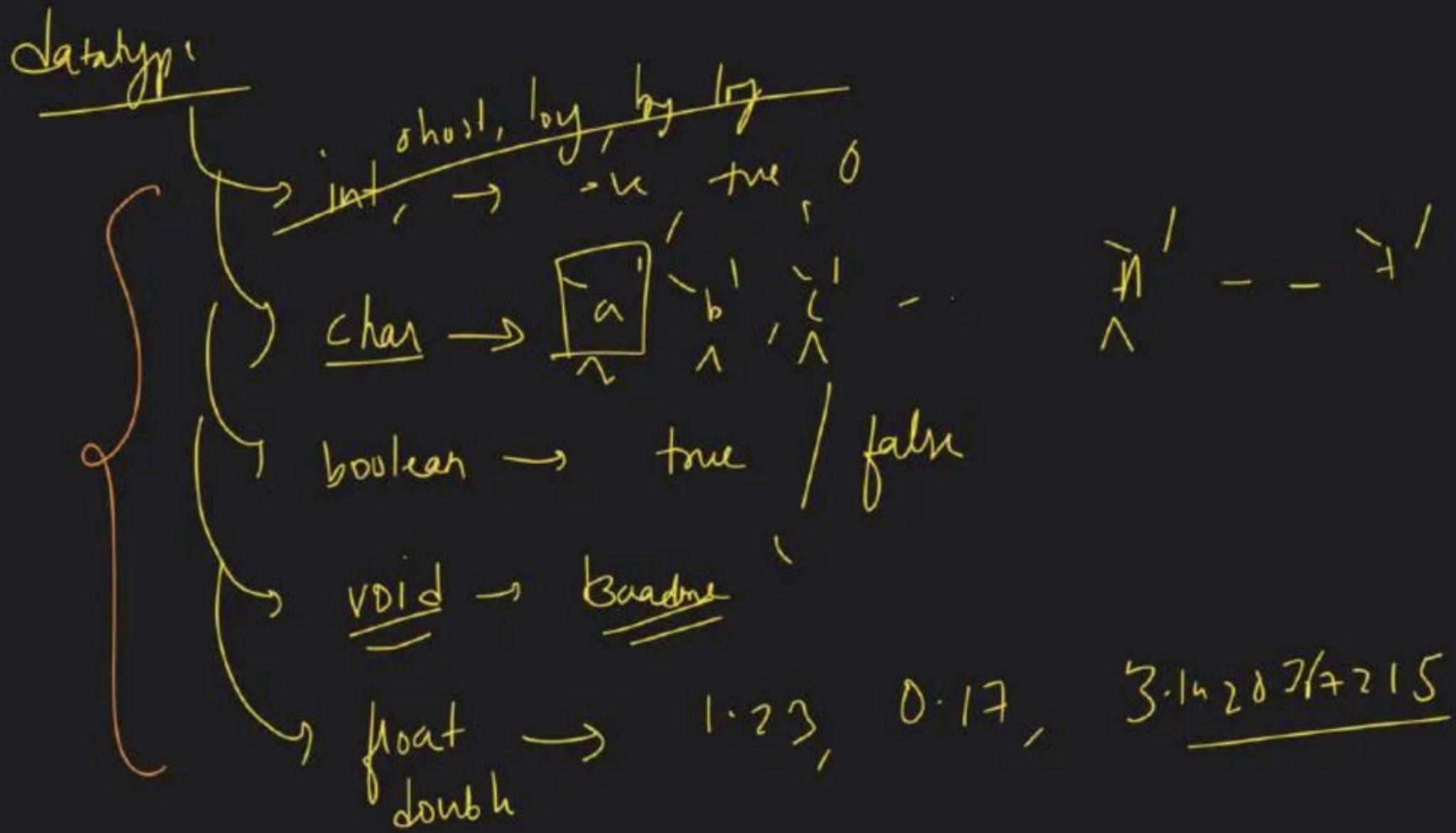



RAM
0
1
2
8byte
108
100
101
variable
num
1byte
Num
Num


Variable
garbage Value
garbage
declaration
variable name
int age;
int 4byte
definition
int a = 5;
initialisation
5
a
float marks;
char grade;
bool flag;




int ageOfAaariz = 21;
int
4byte
int
a = 5;
datatype
5
a
4byte
21
ageOfAaariz


datatype:
int → -ve true 0
char → 'a'
boolean → true / false
void → Baadme
float → 1.23, 0.17, 3.142857
double

```
bool babban = true;
            = 1;
```

true ↔ 1

false ↔ 0

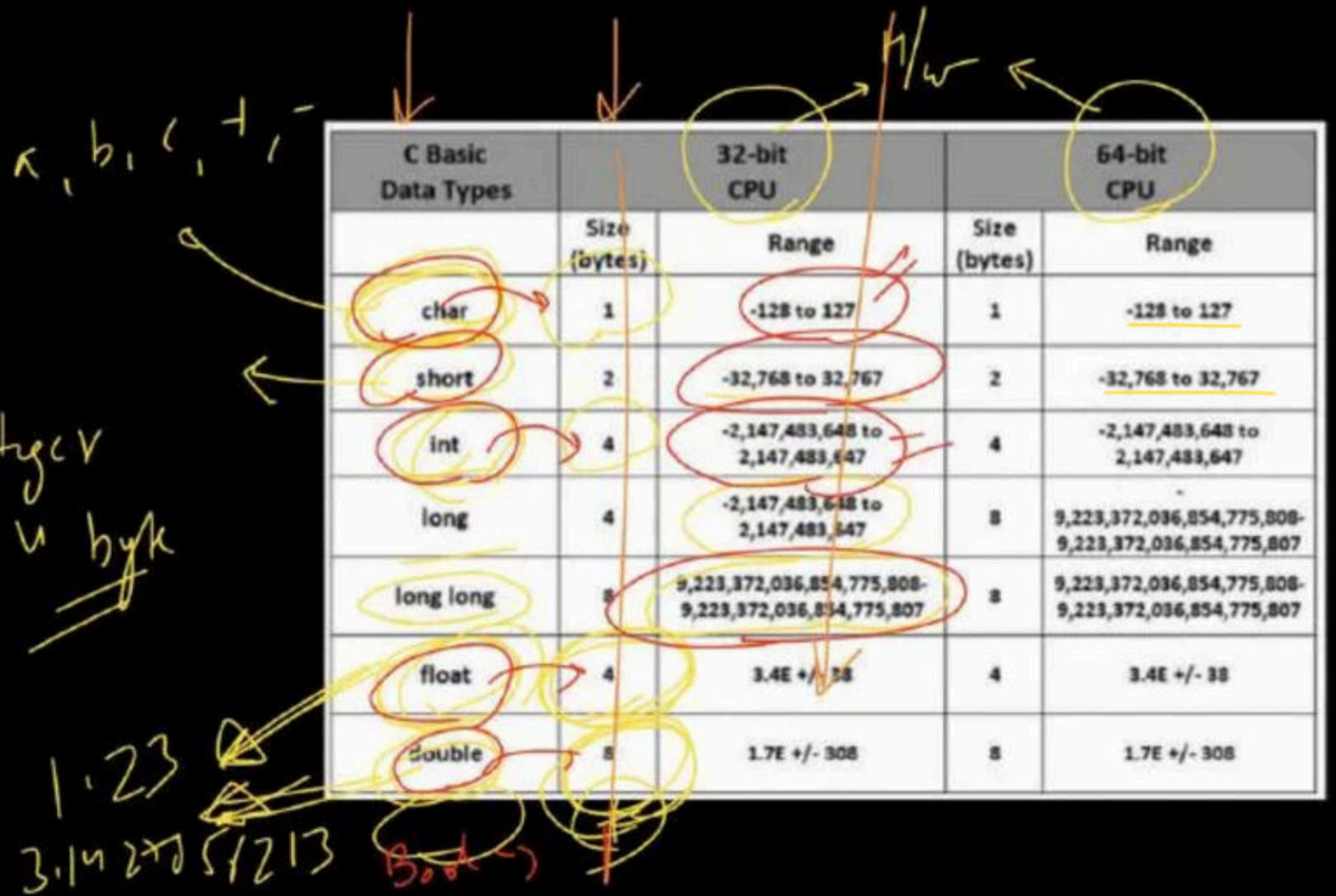
```
char grade = '+';
```

Datatypes:

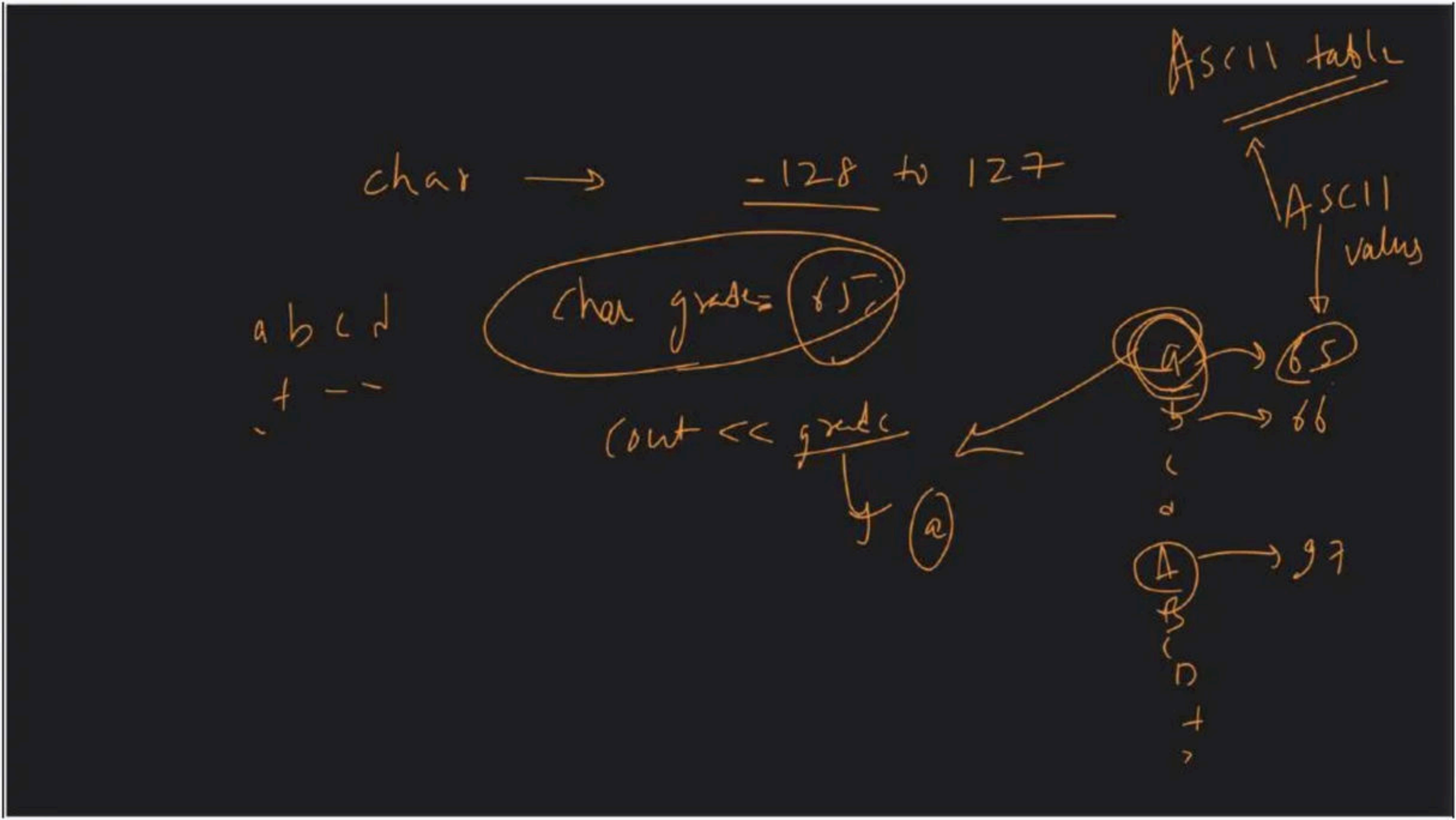
type of data → size of data

The data type specifies the size and type of information the variable will store

- Diagram

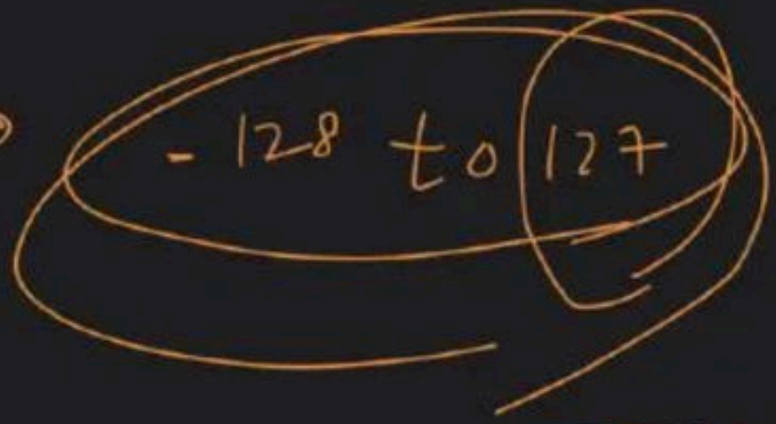


C Basic Data Types	32-bit CPU		64-bit CPU	
	Size (bytes)	Range	Size (bytes)	Range
char	1	-128 to 127	1	-128 to 127
short	2	-32,768 to 32,767	2	-32,768 to 32,767
int	4	-2,147,483,648 to 2,147,483,647	4	-2,147,483,648 to 2,147,483,647
long	4	-2,147,483,648 to 2,147,483,647	8	-9,223,372,036,854,775,808-9,223,372,036,854,775,807
long long	8	9,223,372,036,854,775,808-9,223,372,036,854,775,807	8	9,223,372,036,854,775,808-9,223,372,036,854,775,807
float	4	3.4E +/- 38	4	3.4E +/- 38
double	8	1.7E +/- 308	8	1.7E +/- 308

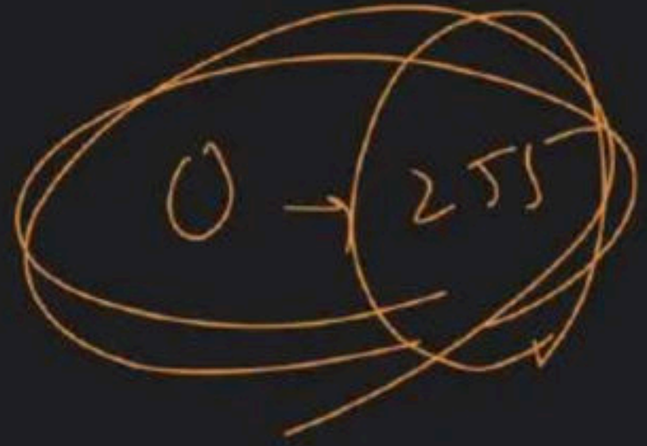

Ascii table
char ⟶ -128 to 127
ASCII values
char grade = 65
abcd
cout << grade
65
66
97
@

char ch = 'a'

signed



unsigned char = 'a'



datatype

→ range

n = no of bits

char → 1byte → 8 bits

n = 8

Signed → $-2^{n-1} \longrightarrow 2^{n-1}-1$

unsigned → $0 \longrightarrow 2^{n}-1$

signed = $-2^{8-1} \longrightarrow 2^{8-1}-1$

$-2^{7} \rightarrow 2^{7}-1$

$-128 \rightarrow 128-1$

$-128 \rightarrow 127$

$0 \rightarrow 2^{8}-1 \rightarrow 0 \rightarrow 255$

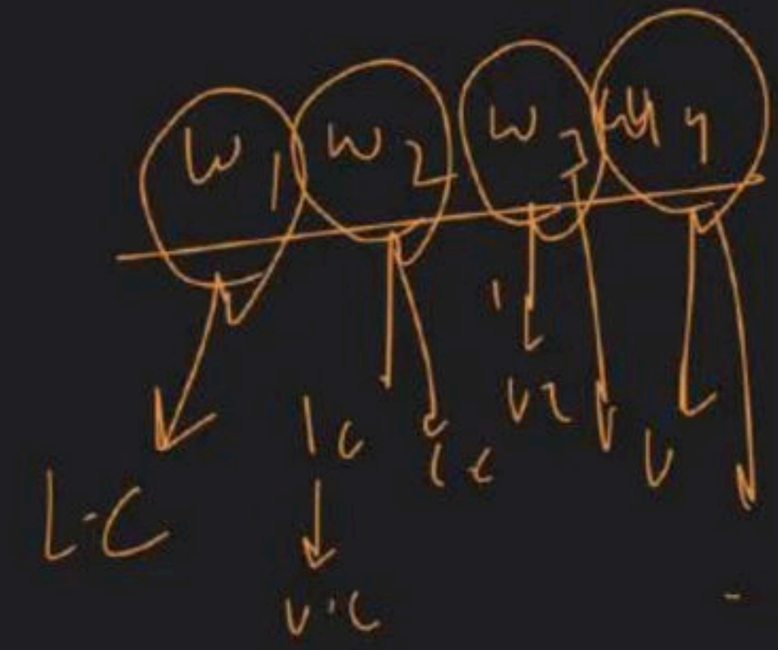
$0 \rightarrow 255$

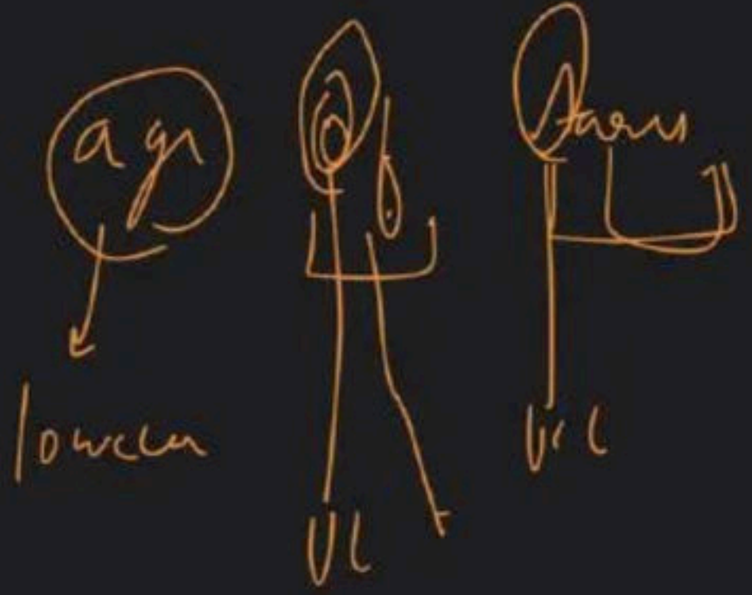
Variable Naming Conventions:

Naming Conventions rules for Variables are:

- It should begin with an alphabet.
- There may be more than one alphabet, but without any spaces between them.
- Digits may be used but only after alphabet.
- No special symbol can be used except the underscore (_) symbol. When multiple words are needed, an underscore should separate them.
- No keywords or command can be used as a variable name.
- All statements in C++ language are case sensitive. Thus a variable A (in uppercase) is considered different from a variable declared a (in lowercase).

```
int love=5;
int Love=5;
```





age Of Aaariz

lowBabbar

lowBabbar

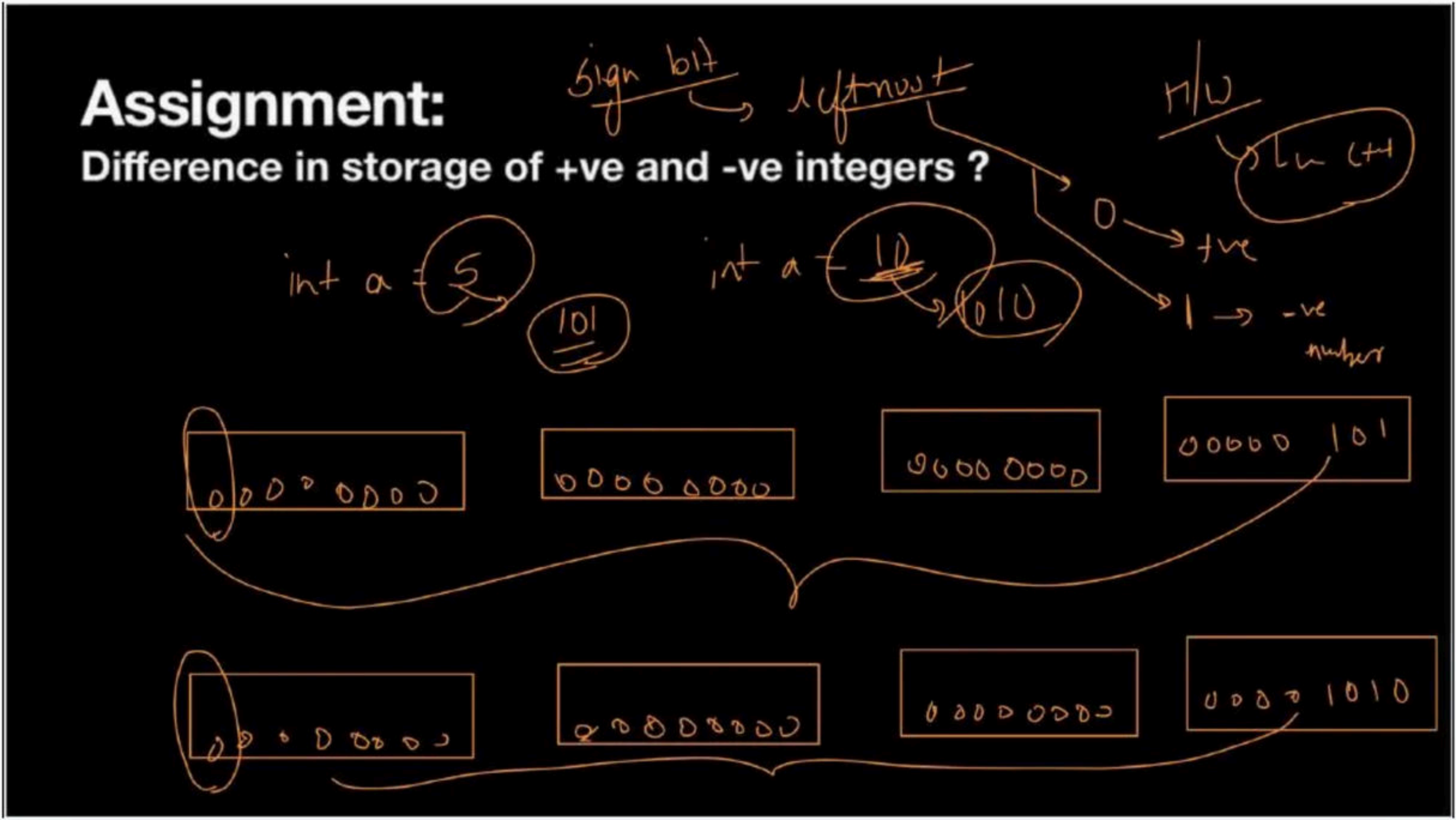
harsh Kapoor

adityaKumar

love_babbar

Assignment:

Difference in storage of +ve and -ve integers ?



1's complement

2's complement

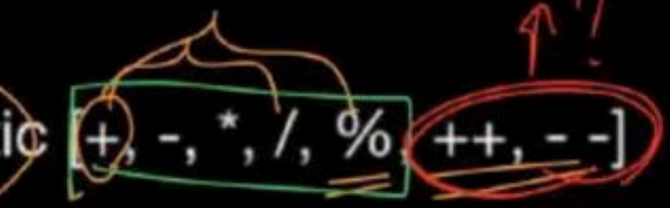
+ve - vev

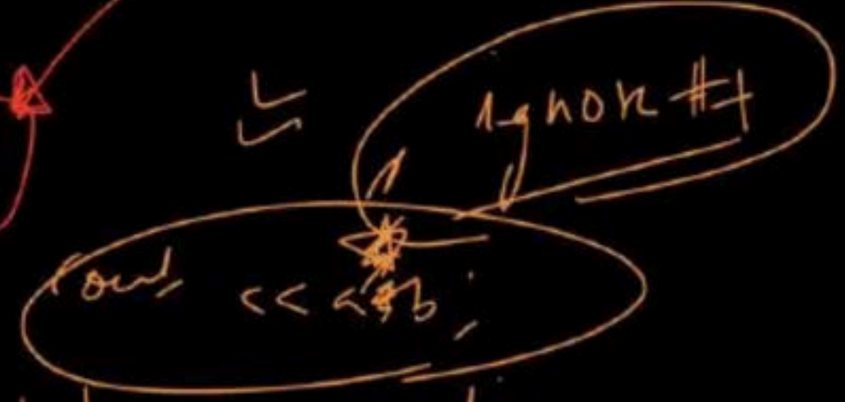
Signed vs Unsigned Integers:

Operators:

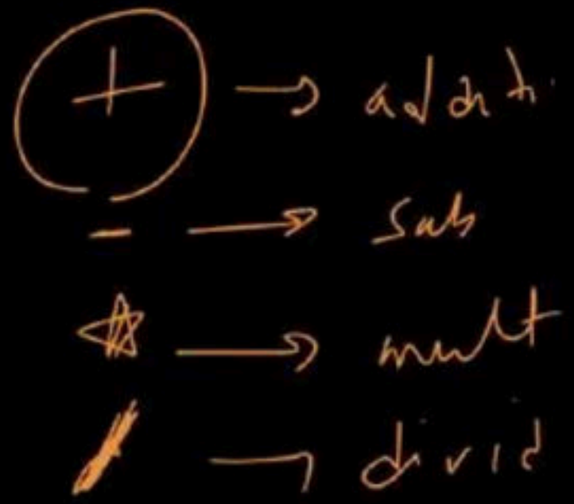
Operators are used to perform operations on variables and values.

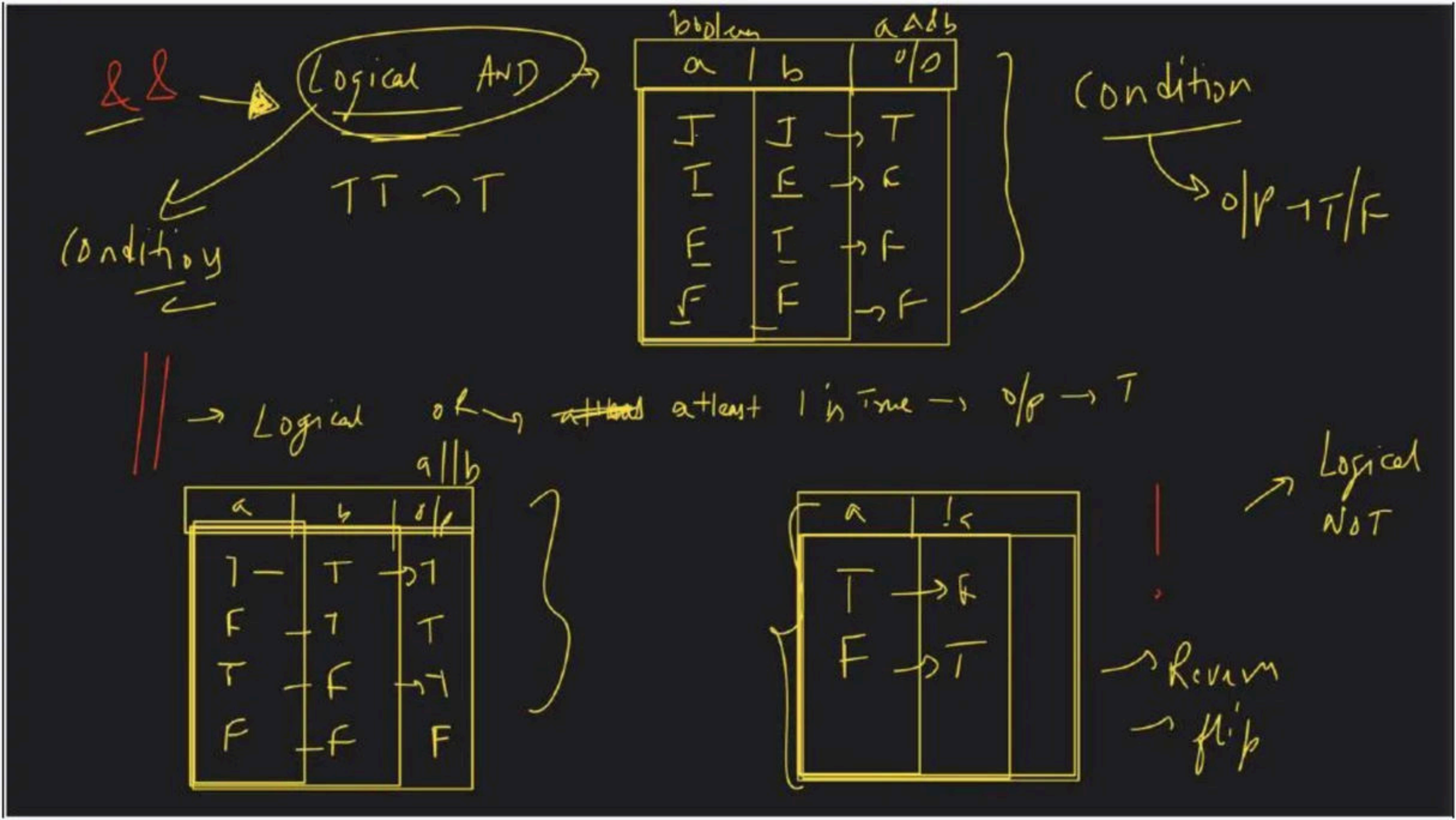
- Arithmetic [+, -, *, /, %, ++, - -]
- Relational [==, !=, >, <, >=, <=]
- Assignment [=]
- Logical [&&, ||, !]
- Bitwise [&, |, ~, ^, >>, <<]

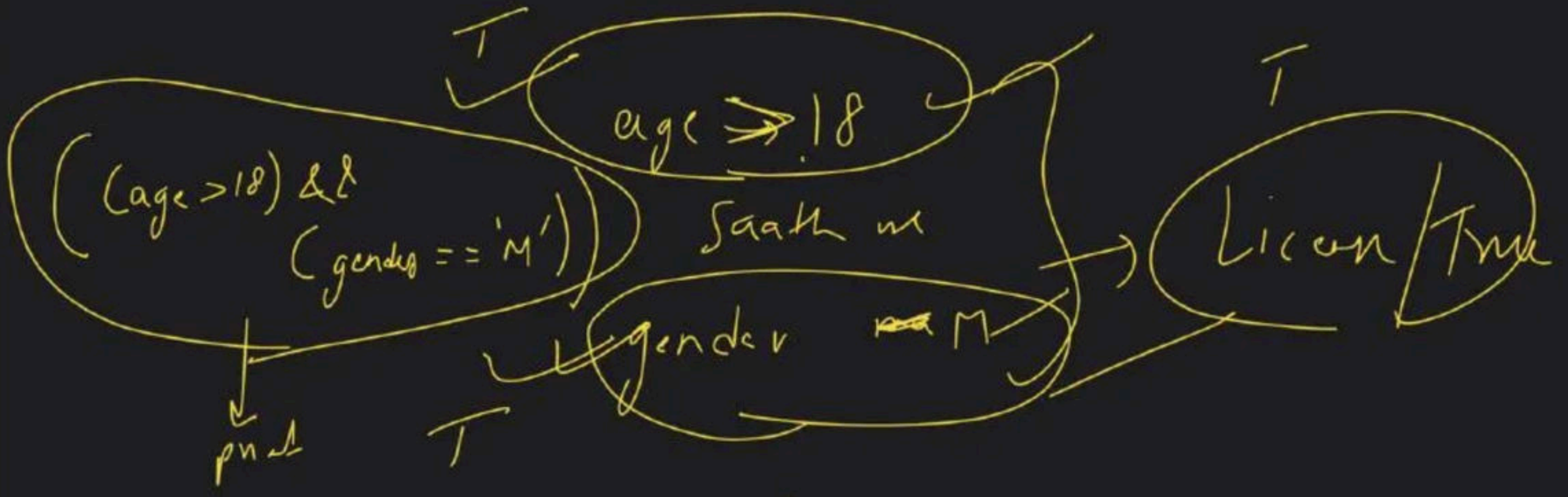




5.0/0.5

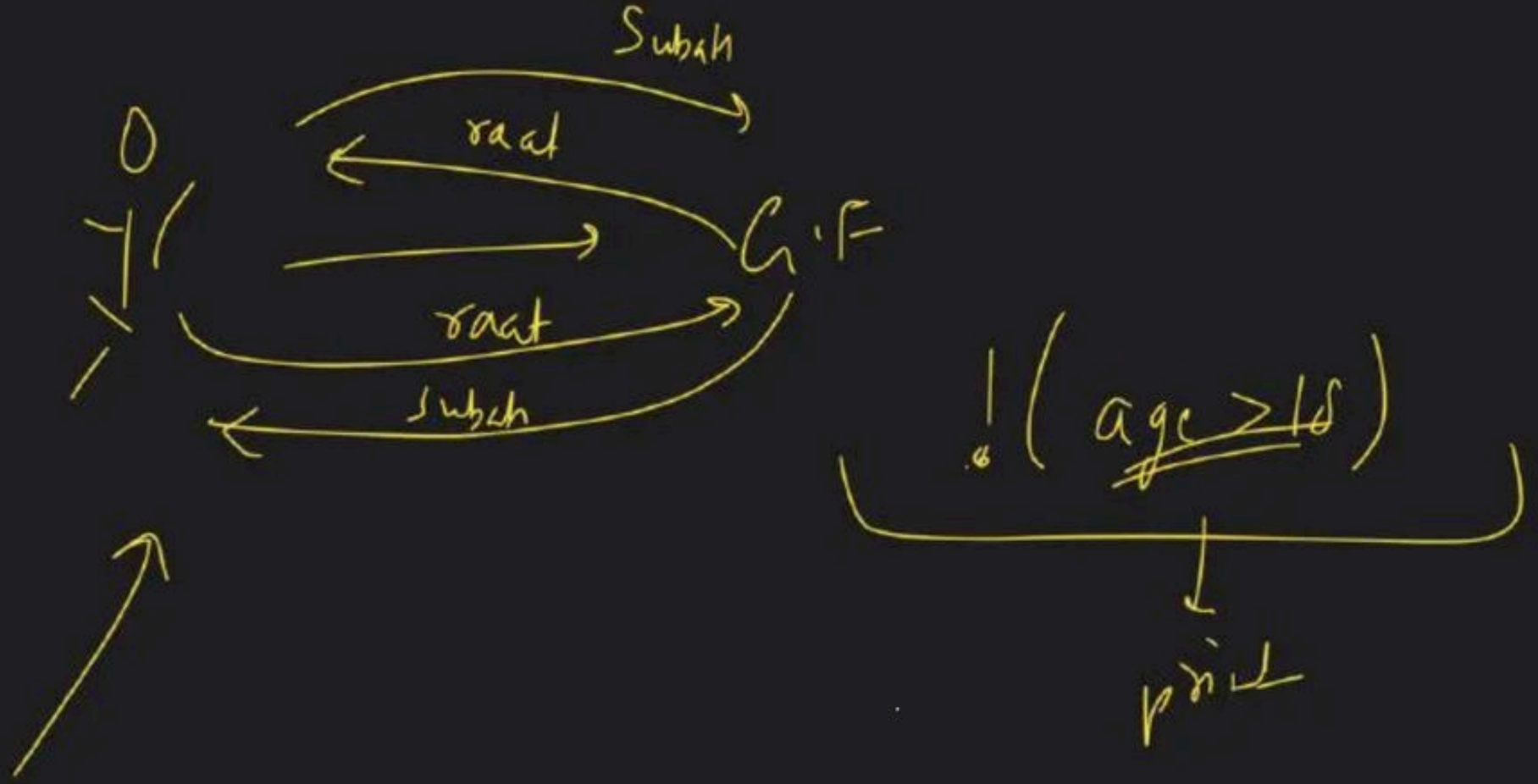



&&
Logical AND
TT → T
Condition
Condition
o/p → T/F
boolean
a && b
a | b | o/p
T | T → T
T | F → F
F | T → F
F | F → F
|| → Logical OR → atleast 1 is true → o/p → T
a || b
a | b | o/p
T | T → T
F | T → T
T | F → T
F | F → F
! → Logical NOT
a | !a
T → F
F → T
→ Reverse → flip



else → False

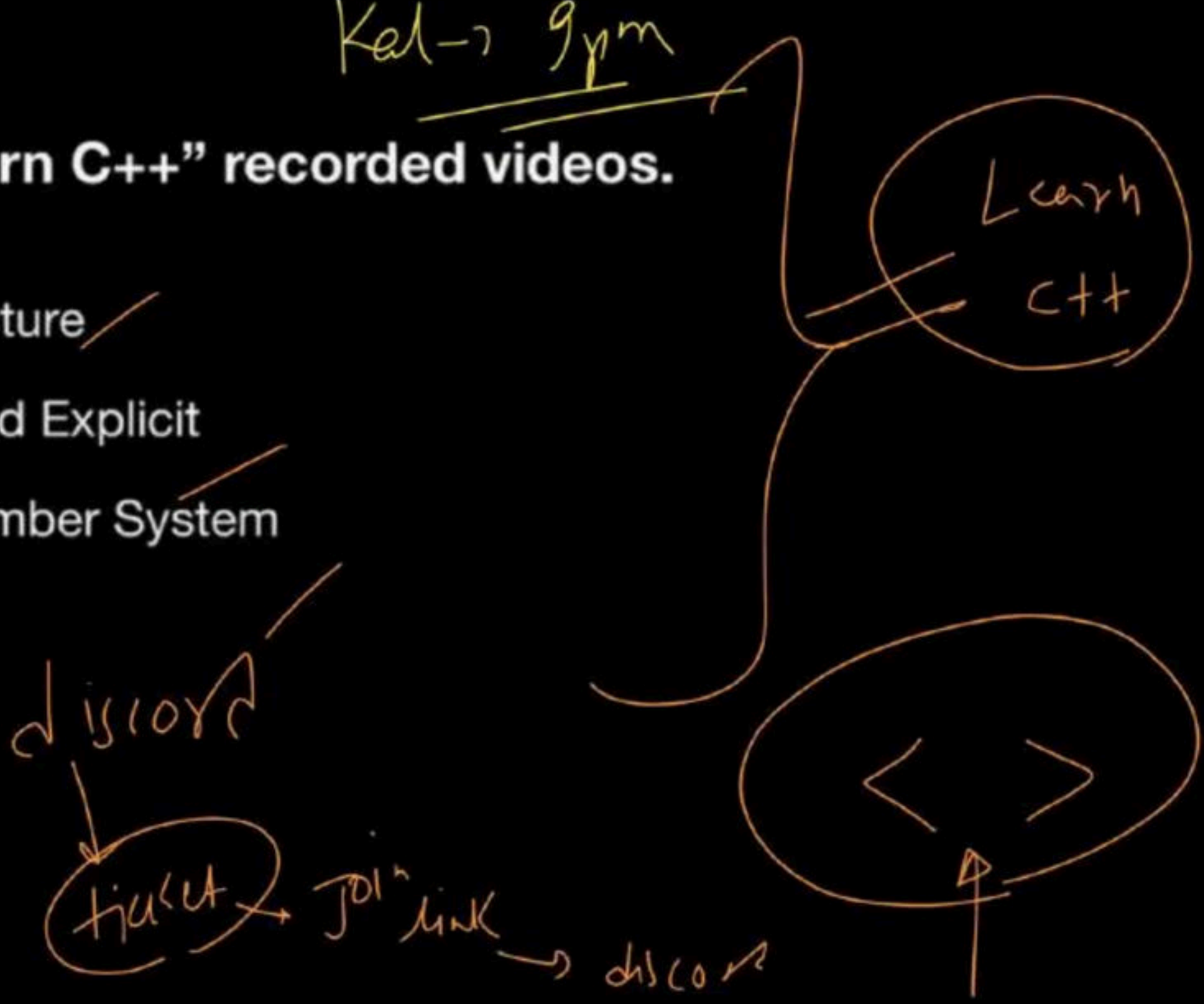
((age > 18) || (gender == 'M'))


Subah
raat
raat
subah
!(age>18)

Assignment:

Complete week-"Learn C++" recorded videos.

- 32 bit vs 64 bit Architecture
- Typecasting: Implicit and Explicit
- Binary and Decimal Number System



L3-Basics of Programming

Special class

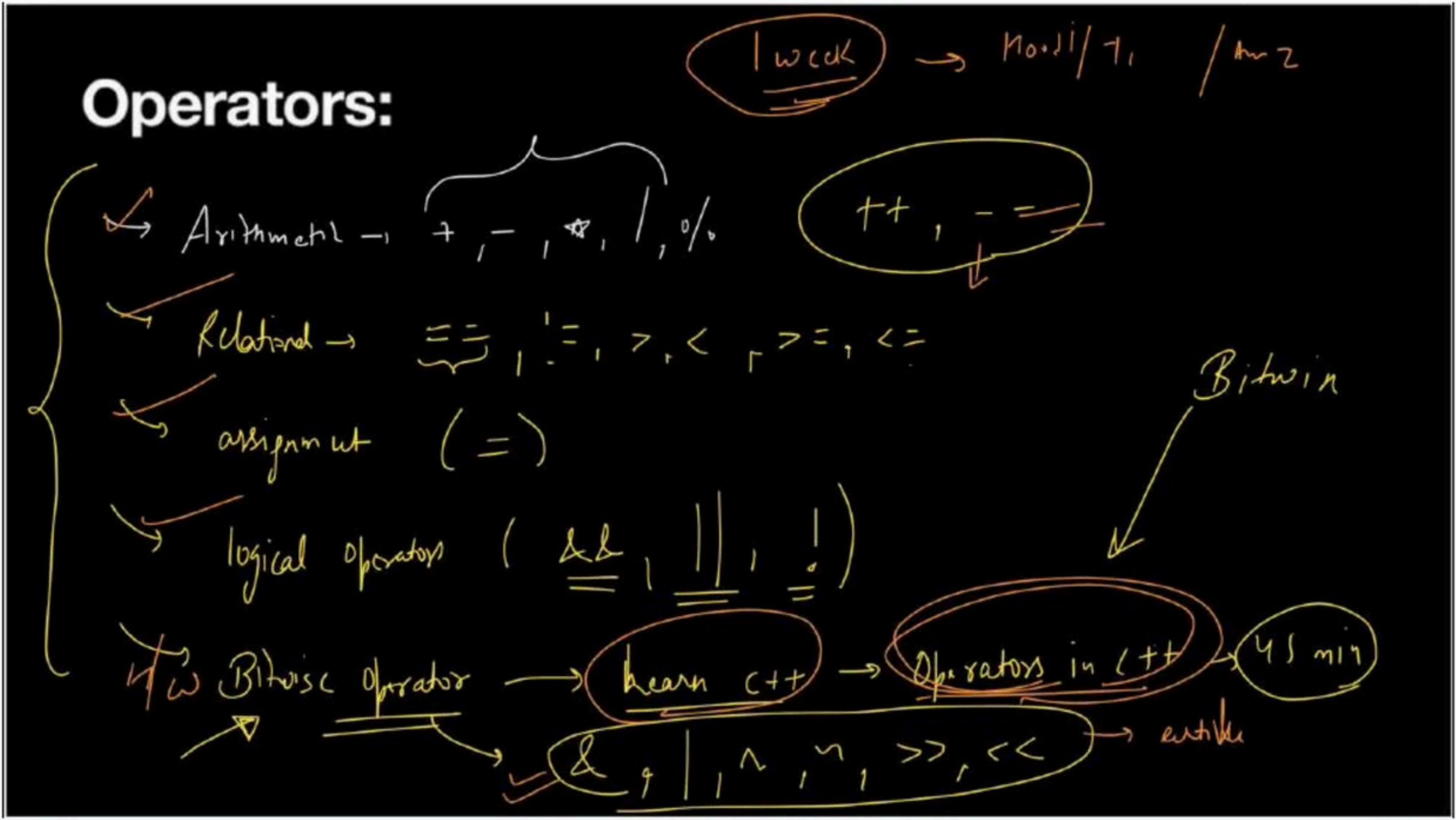
Love Babbar • Apr 4, 2025

L3 - Basics of Programming

Operators, Conditionals & Loops

—> by Codehelp

Operators:


Week → Modi 1, 2
Arithmetic → +, -, *, /, %
++, --
Relational → ==, !=, >, <, >=, <=
assignment (=)
logical operators (&&, ||, !)
Bitwise
Bitwise Operator → learn C++ → Operators in C++ → 45 min
&, |, ^, ~, >>, <<
extra

Arithmetic

Binary Op

2 operands

+
-
*
/
%

operands

operator = +

Unary Op

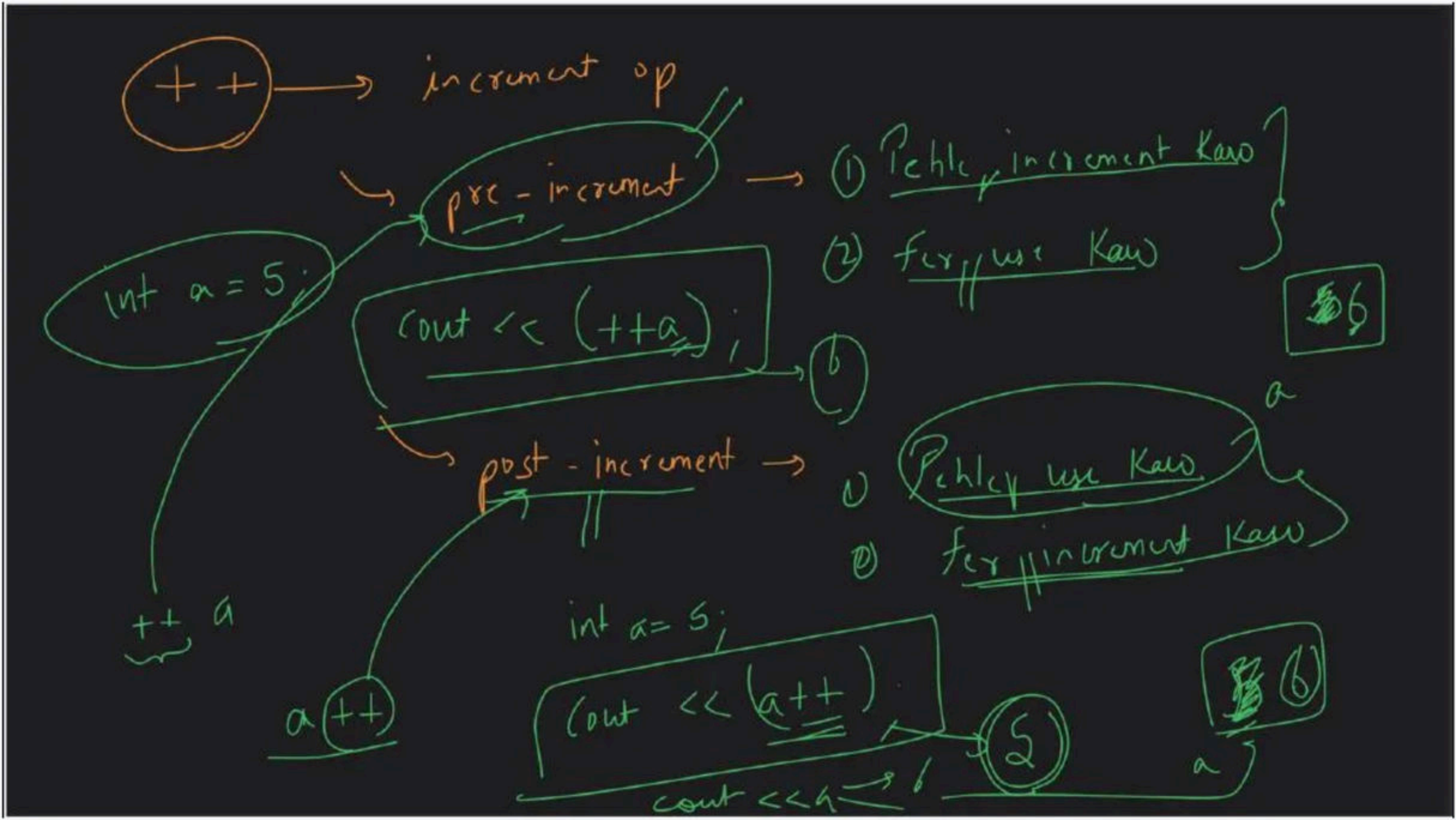
1 operand

++ --

MCQs

Comment

Links


++ → increment op
pre - increment → ① Pehle increment karo
② fir use karo
int a = 5;
cout << (++a);
6
b
post - increment →
① Pehle use karo
② fir increment karo
a
++ a
int a = 5;
a ++
cout << (a++)
5
6
a
cout << a → 6

-- → decrement

- pre-decrement → (--a)
- post-decrement → (a--)

pre-decrement → ① Pehla decrement karo
② fir use karo

int a = 5

cout << (--a) → 4

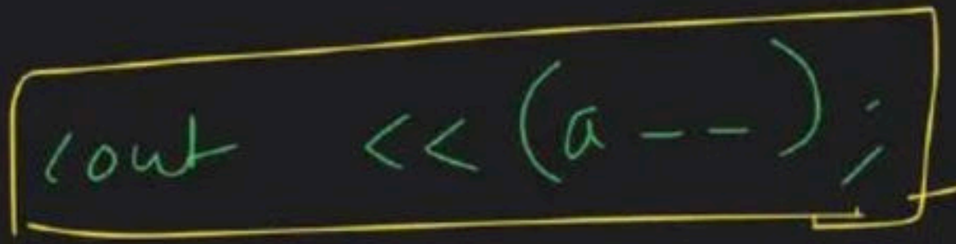
→ cout << a → 4

post - decrement → ① Pehle use Karo

② Fir decrement Karo

int a = 5;

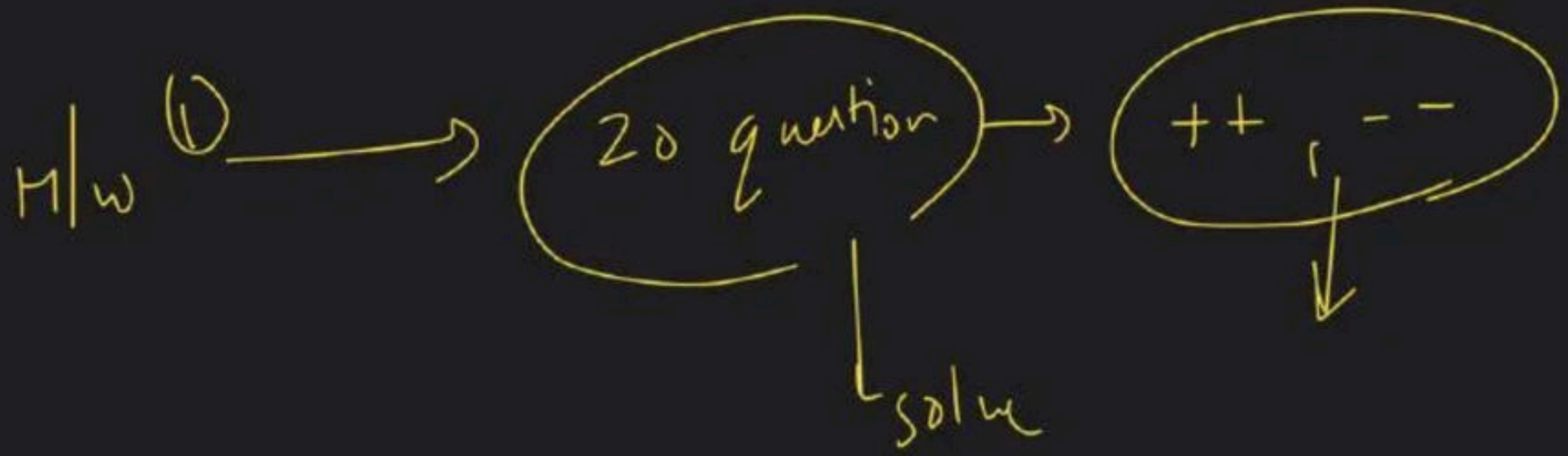
cout << (a--);



→ 5

a

cout << a+5; → 9



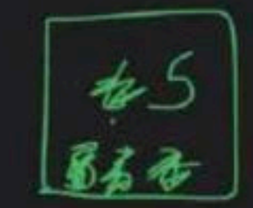
2 Red Quiz → unlocked

Question :-

2 min

short - circuit condition

```
int a = 5;
cout << (++a);   → 6
cout << a;       → 6
cout << (a++);   → 6
cout << a;       → 7
cout << (--a);   → 6
cout << "\n";    → (
cout << (a--);   → 6
cout << a;       → 5
```



a

any → lakh bar a = 5

H/W

int val =

60

(++a) * (a++) + (--a) * (a--)

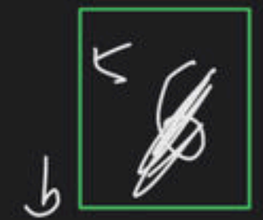
cout << val;

```
int a = 5;
cout << (++a) ⇒ 6
,,     a      ⇒ 6
     a++      ⇒ 6
     a        ⇒ 7
   (--a)      ⇒ 6
     a        ⇒ 5
   (a--)      ⇒ 6
   (a++)      ⇒ 5
    ^         ⇒ 6
```

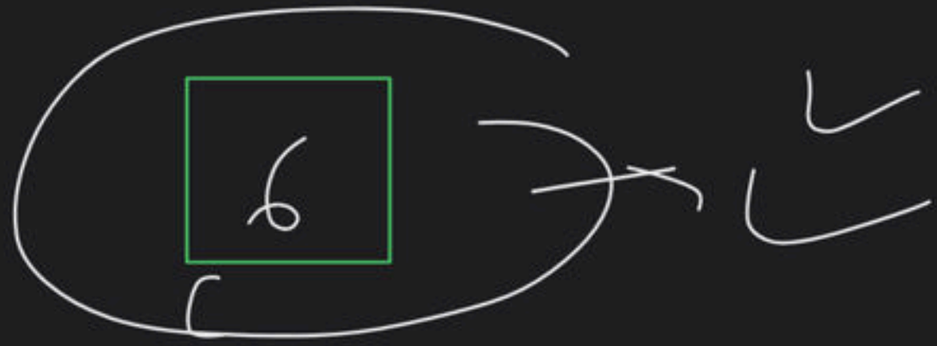
a = 1

b = (a++) + 5;

c = (--b) + 1;



value (c) ? ⇒



b = (11 + 5)

c = 5 + 1

val = (++a) * (a++) + (--a) * (a--);

① Ambiguous

② Undefined Behaviour

garbage value → Tech. Undefn

int a;

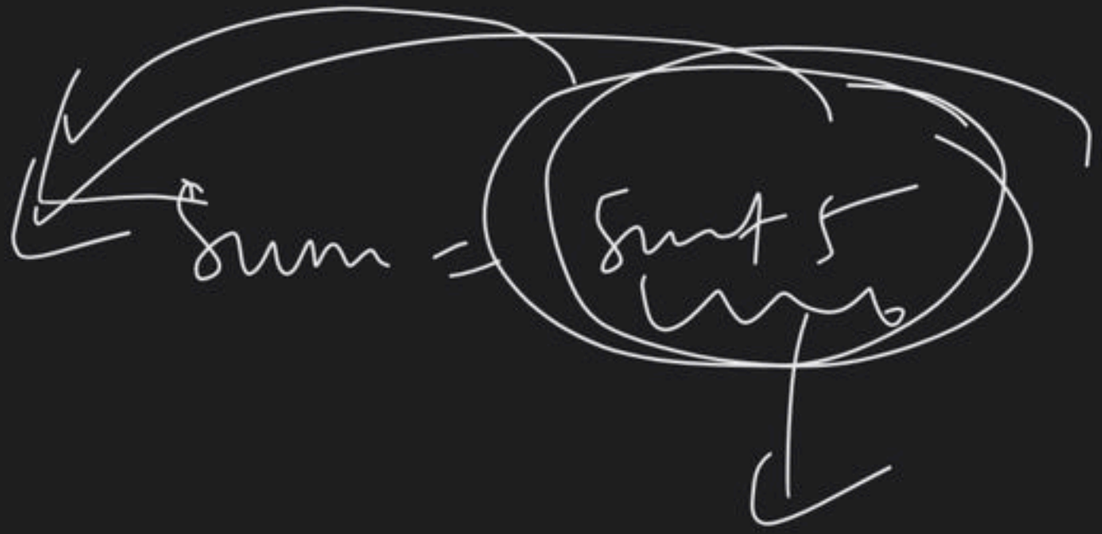
b = a + 10;

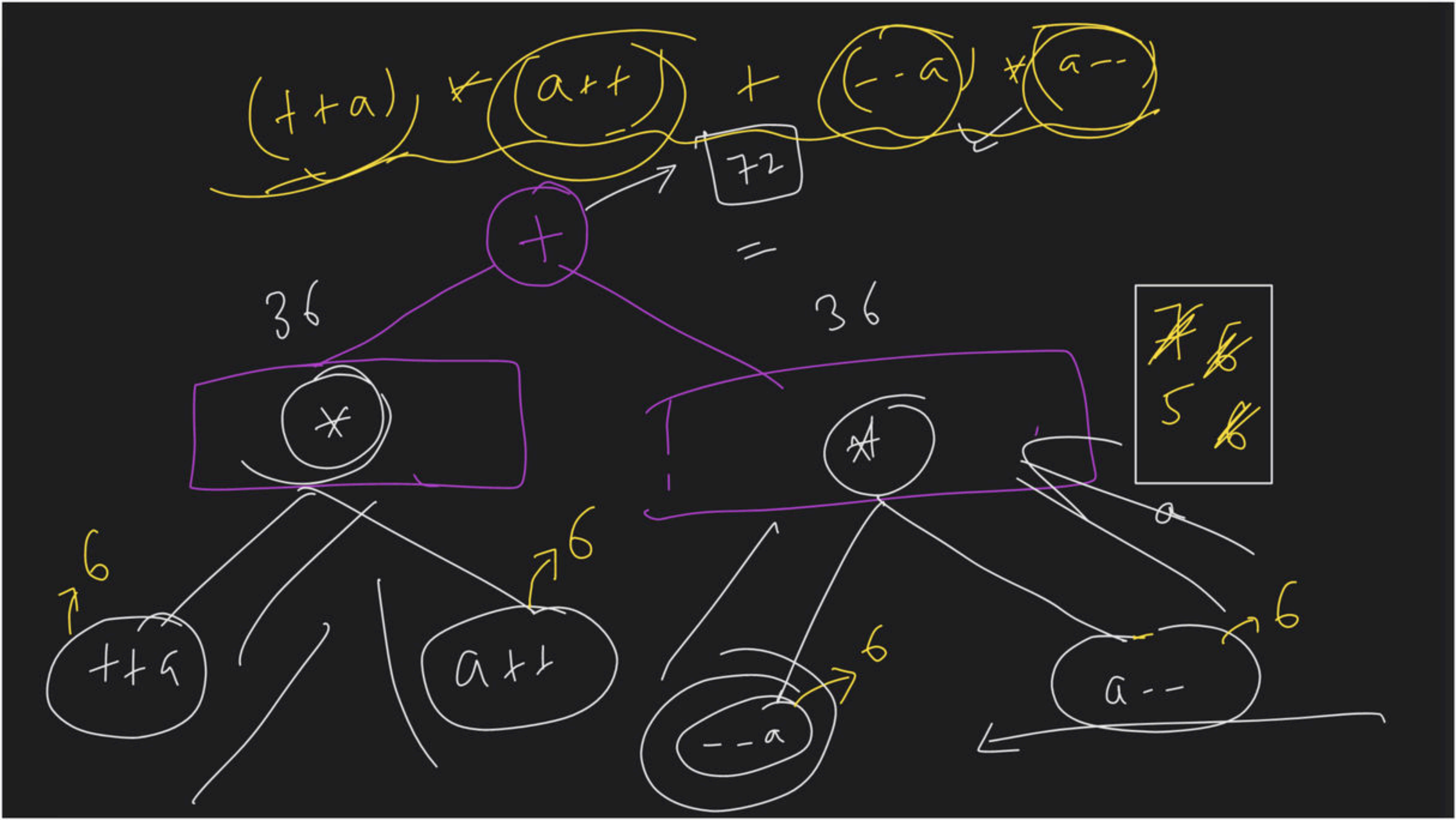
a = 0

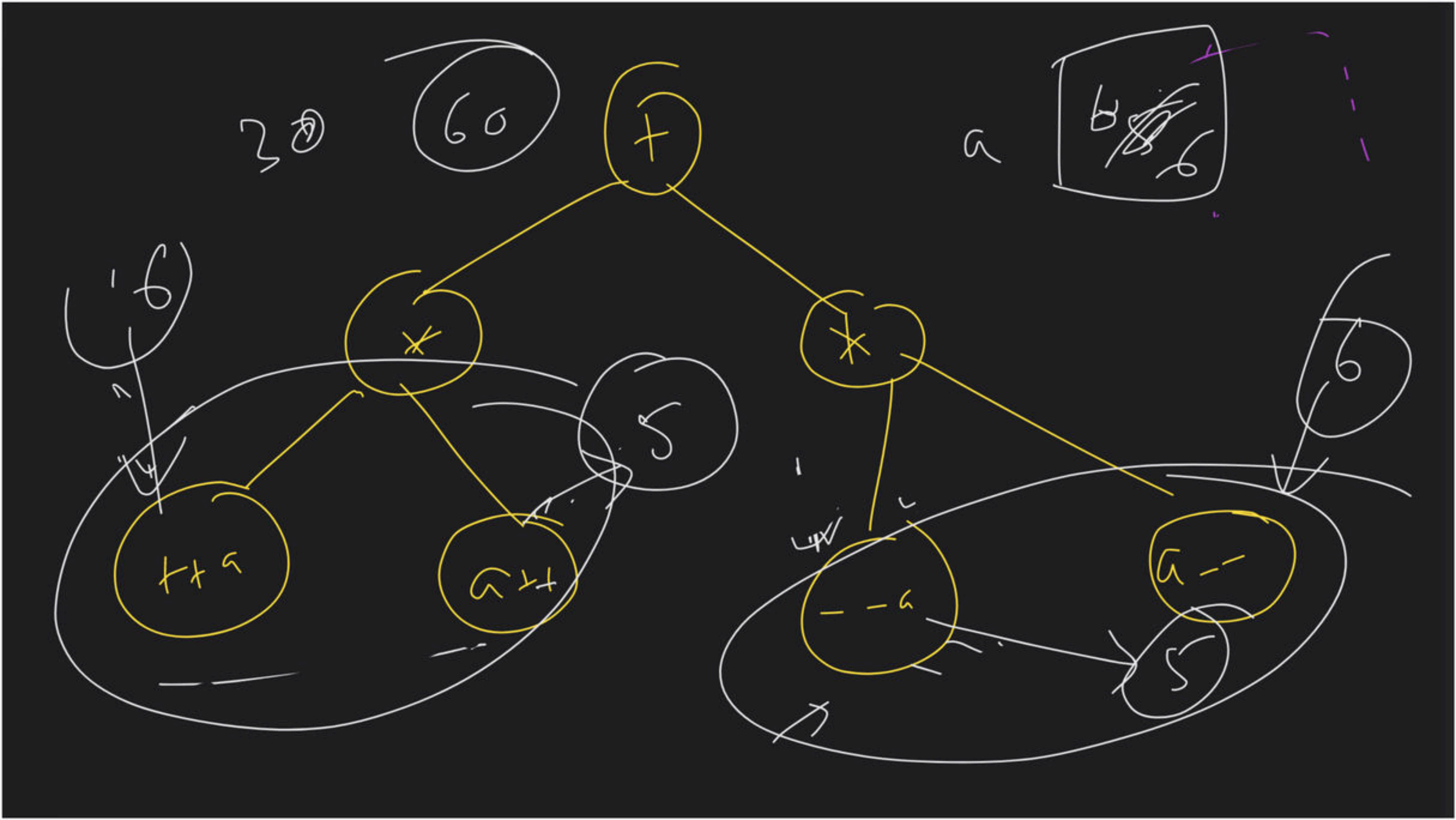
b = 10

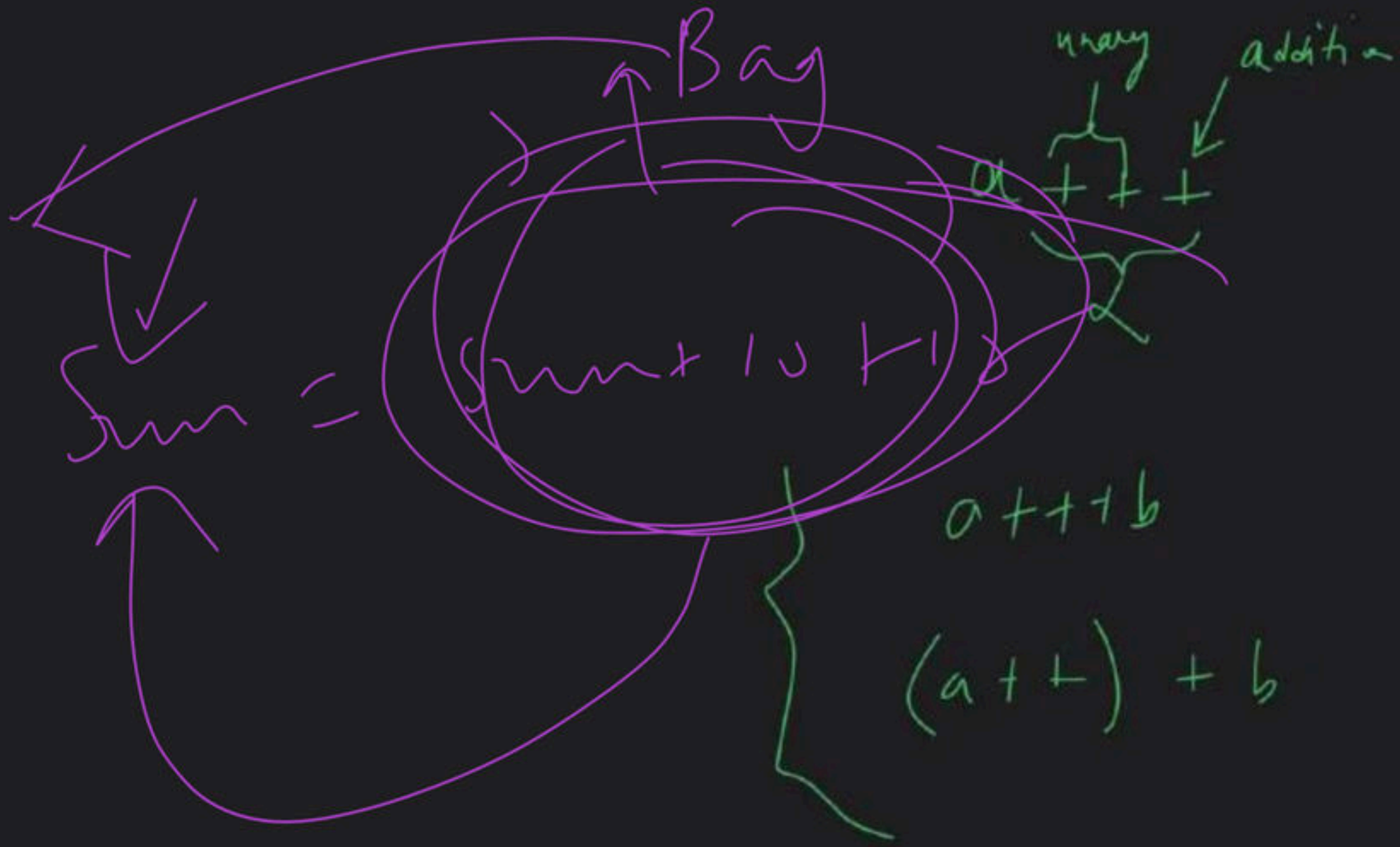
int a;

(++a) + 5




(++a) * a++ + --a * a--
72
+
=
36
36
*
*
7 6
5 6
a
6
++a
6
a++
6
--a
6
a--


20
60
+
a
b
6
10
*
*
5
6
++a
a++
--a
a--
5

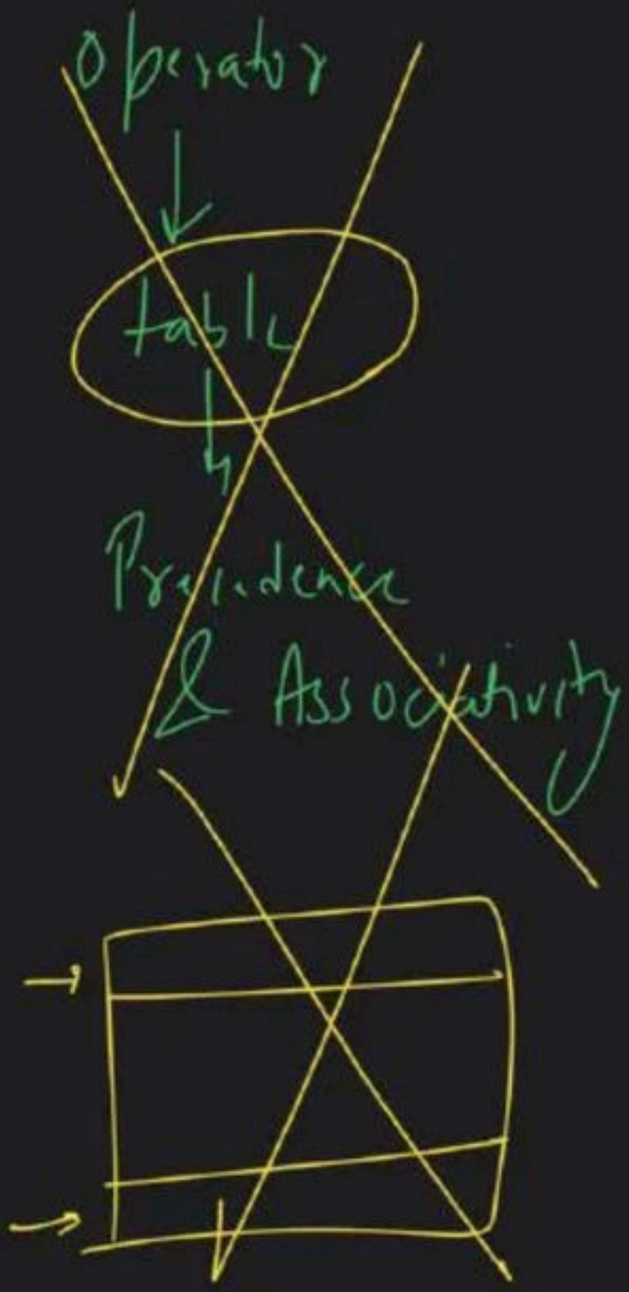

Bag
sum =
sum + 1
unary
addition
a + + + b
(a + +) + b

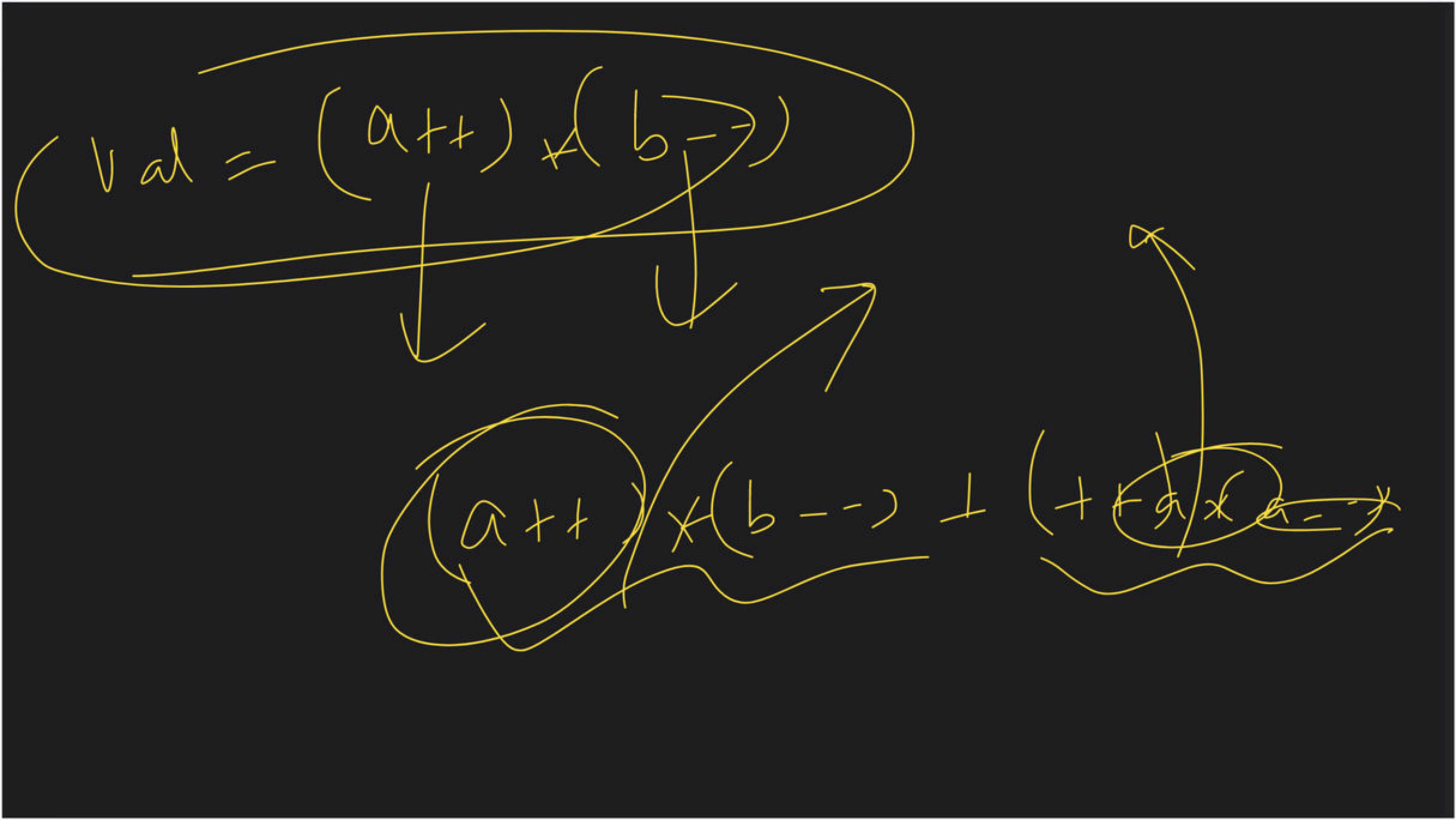
val = (a++) * (++a)

a++ * ++a

[[2 + (3 × 4)] + 5]

exp → Use Brackets




val = (a++) * (b--)
(a++) * (b--) + (++a * a--)

a = 4 ~~5~~ , b = ~~7~~ 6

(--a) * (++b) * 5

4 × 6 × 5

4 × 40

160

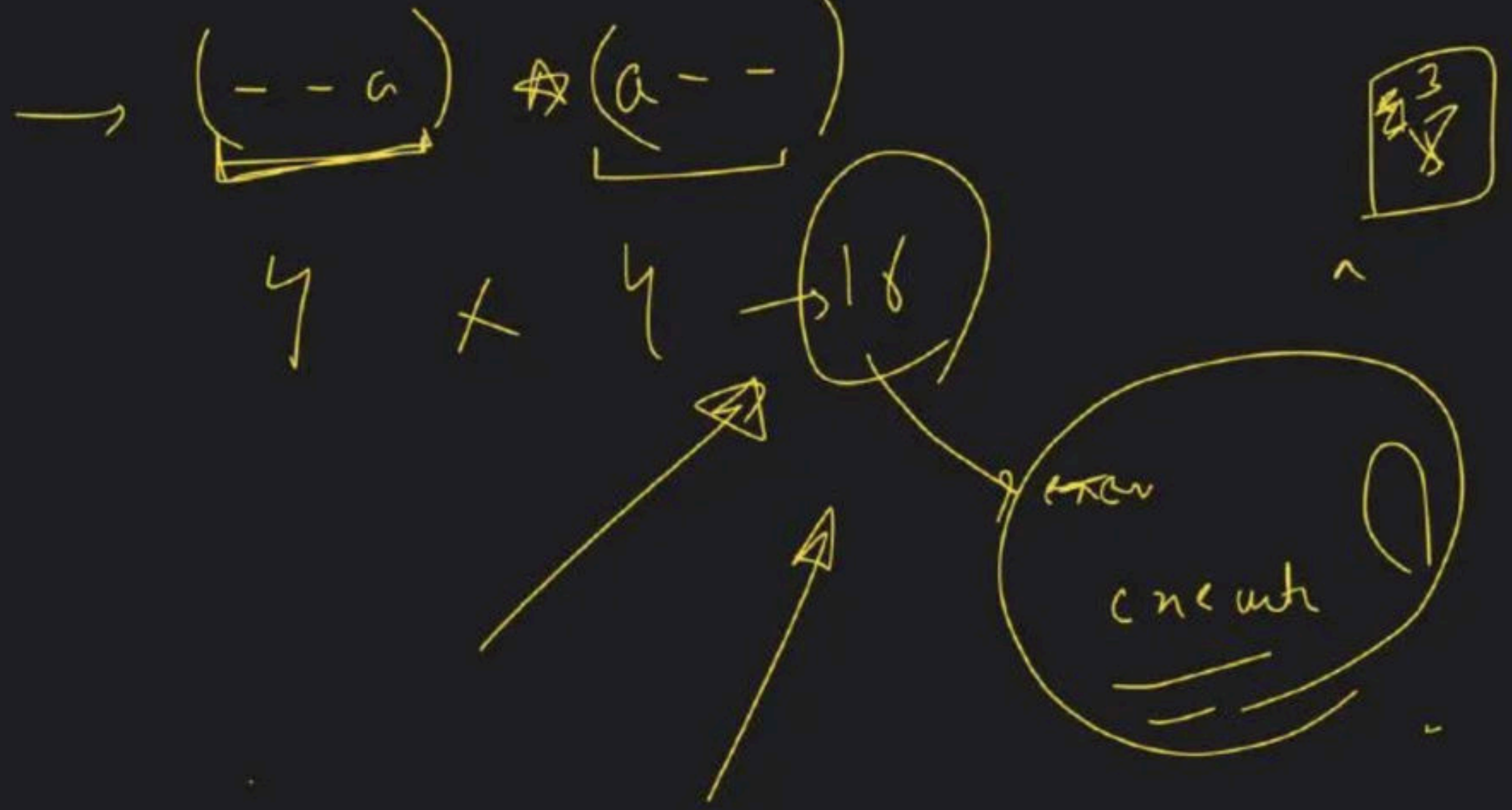
a = 4 ~~5~~ , b = ~~X~~ 7

(--a) * (b++)

4 × 6

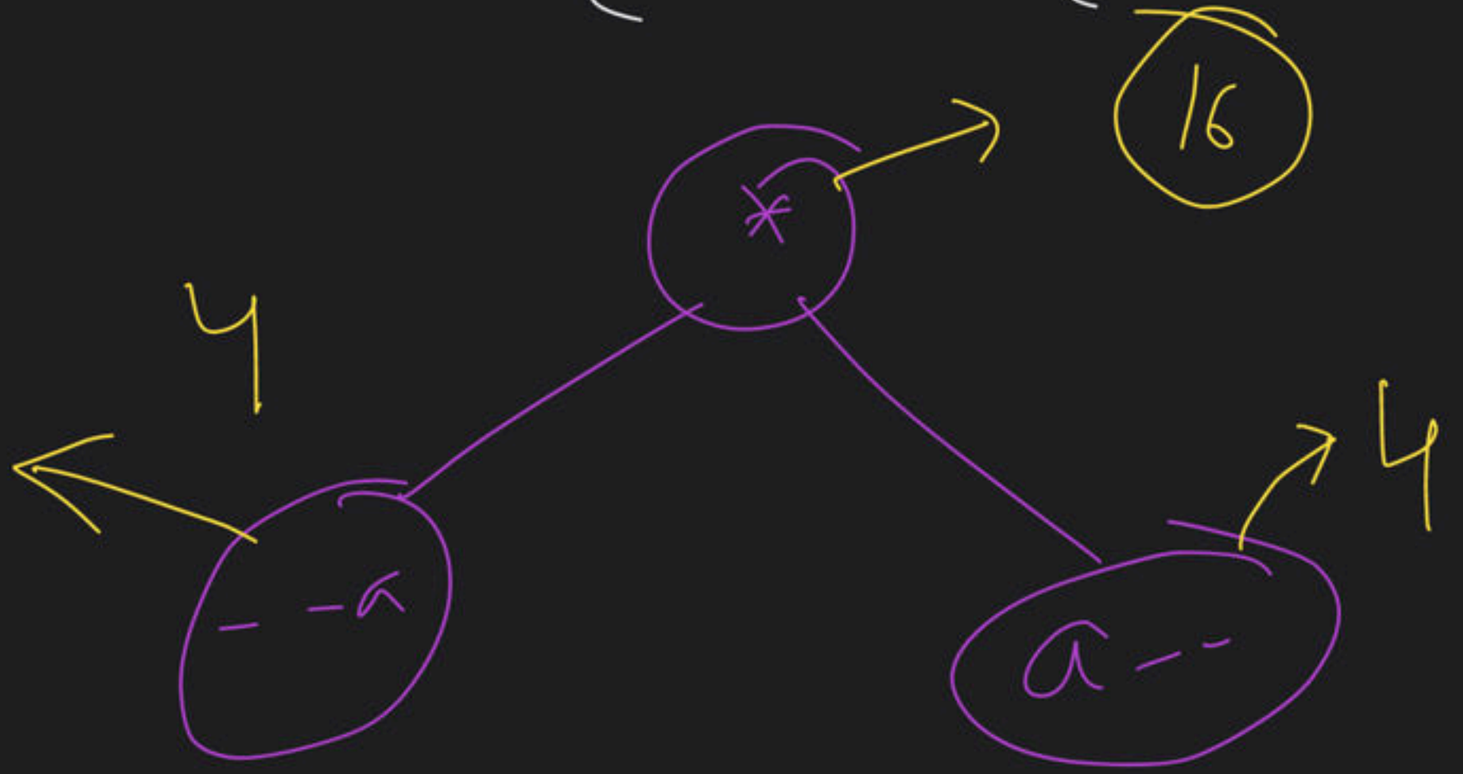
24

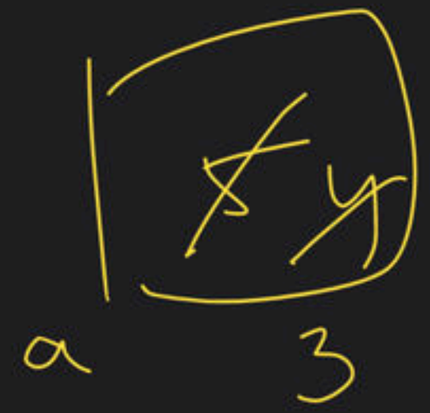
a = 5



$(--a) * (a--)$

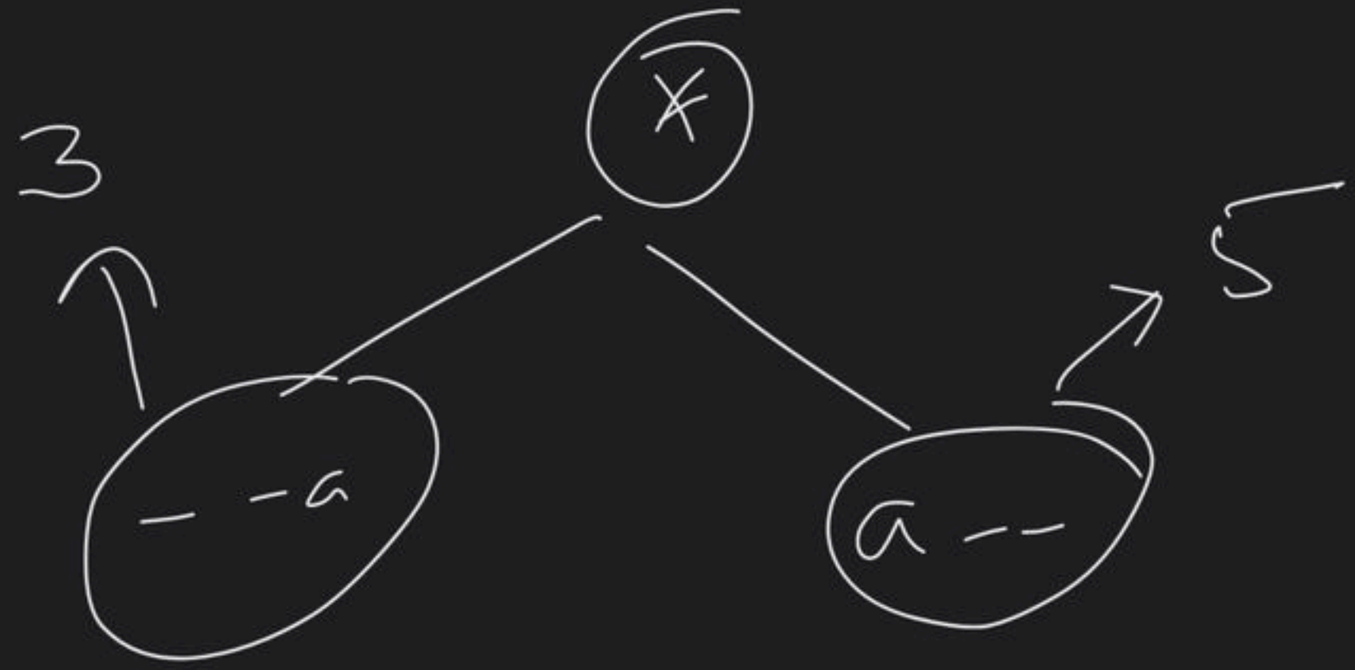
$a = 5$

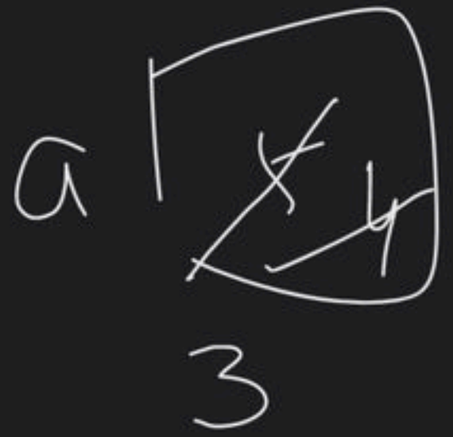


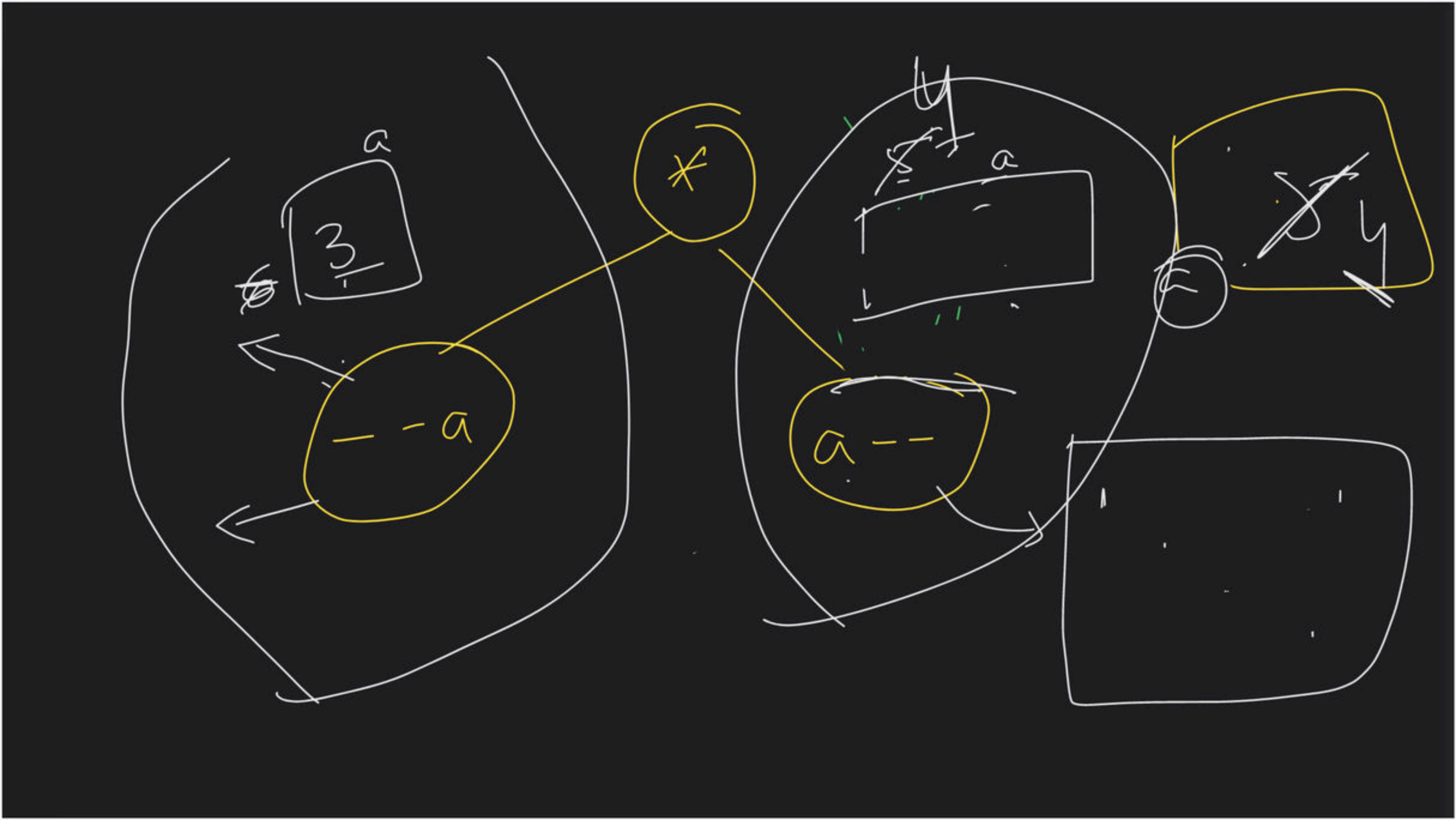


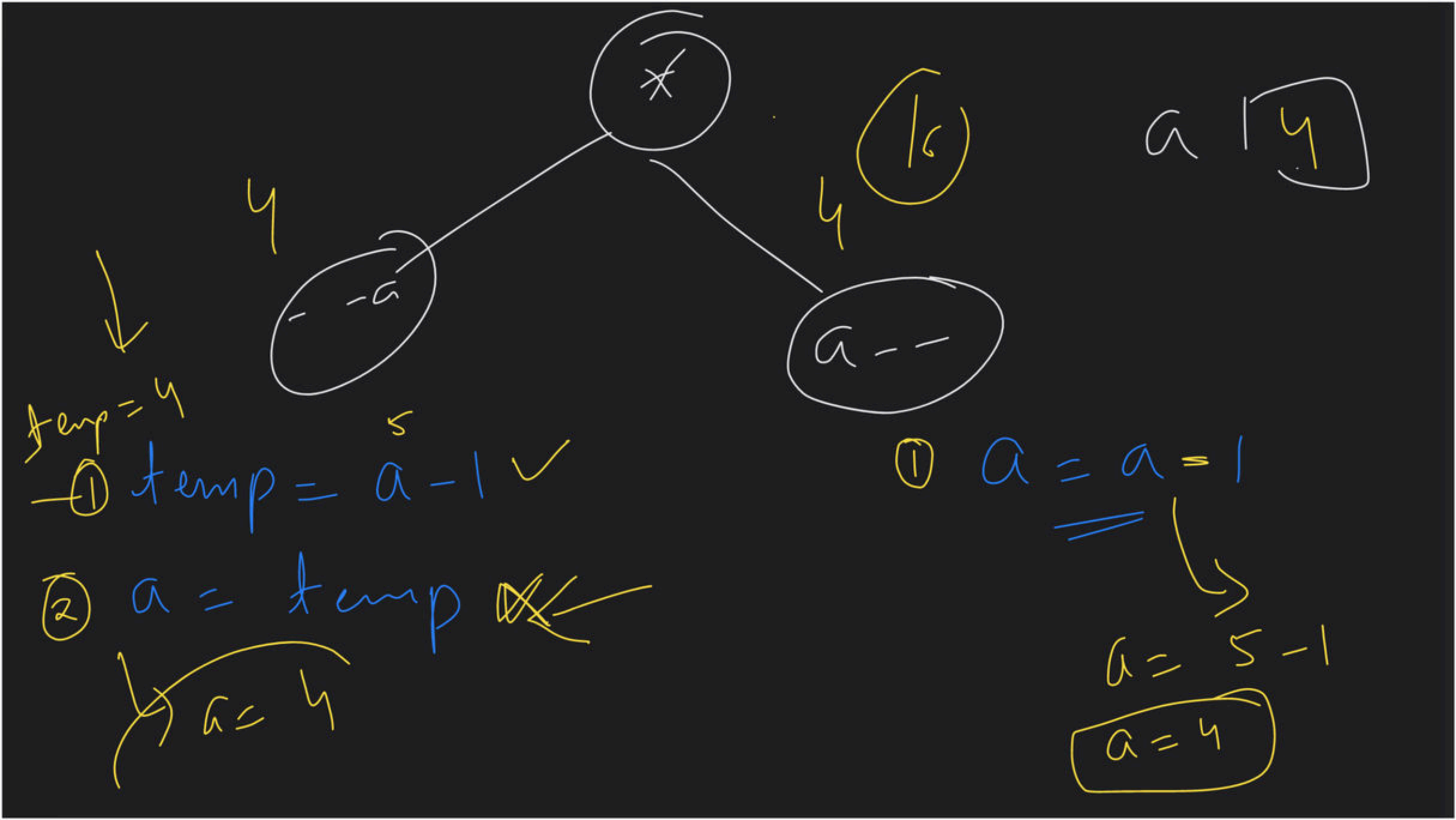
$(--a) * (a--)$

$\rightarrow 15$





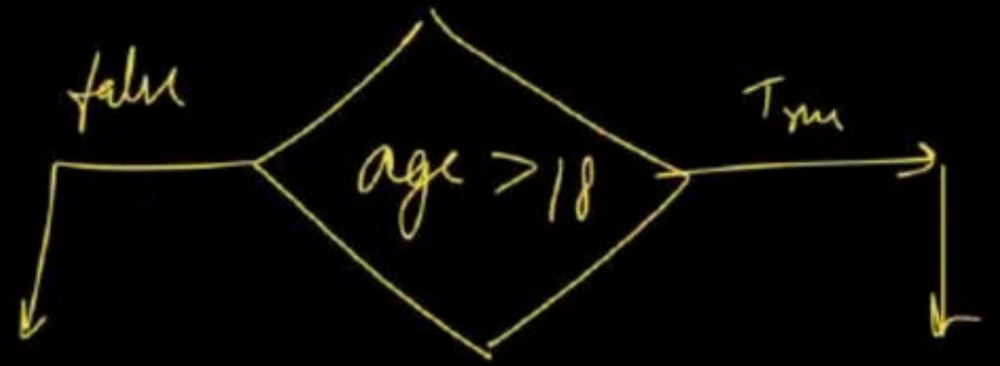

a
3
a
a
a
y

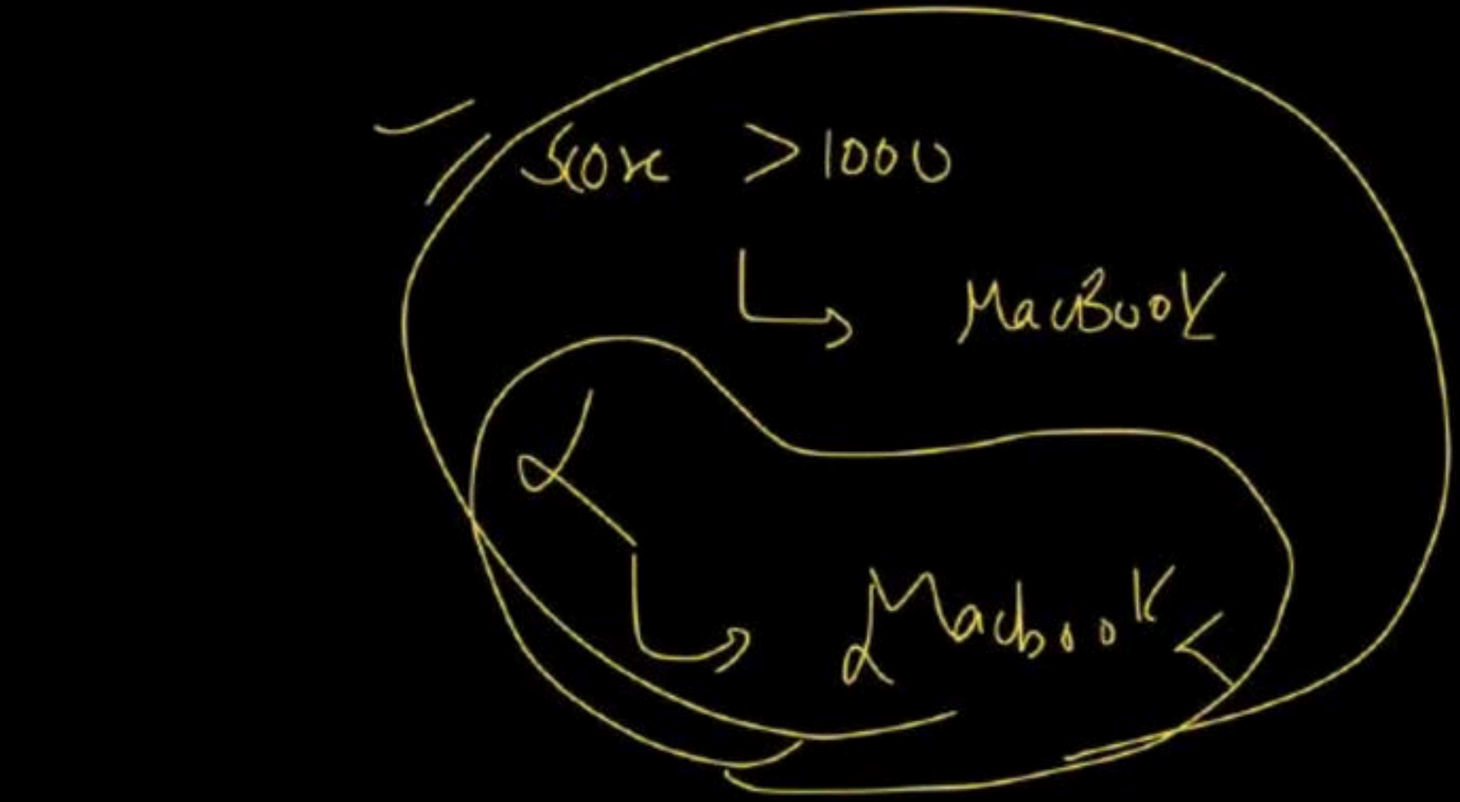

*
- -a
a--
4
4
5
temp = 4
① temp = a - 1 ✓
② a = temp
a = 4
16
a | 4
① a = a - 1
a = 5 - 1
a = 4

$$\frac{99999\ldots\ldots9}{13} \rightarrow ?$$

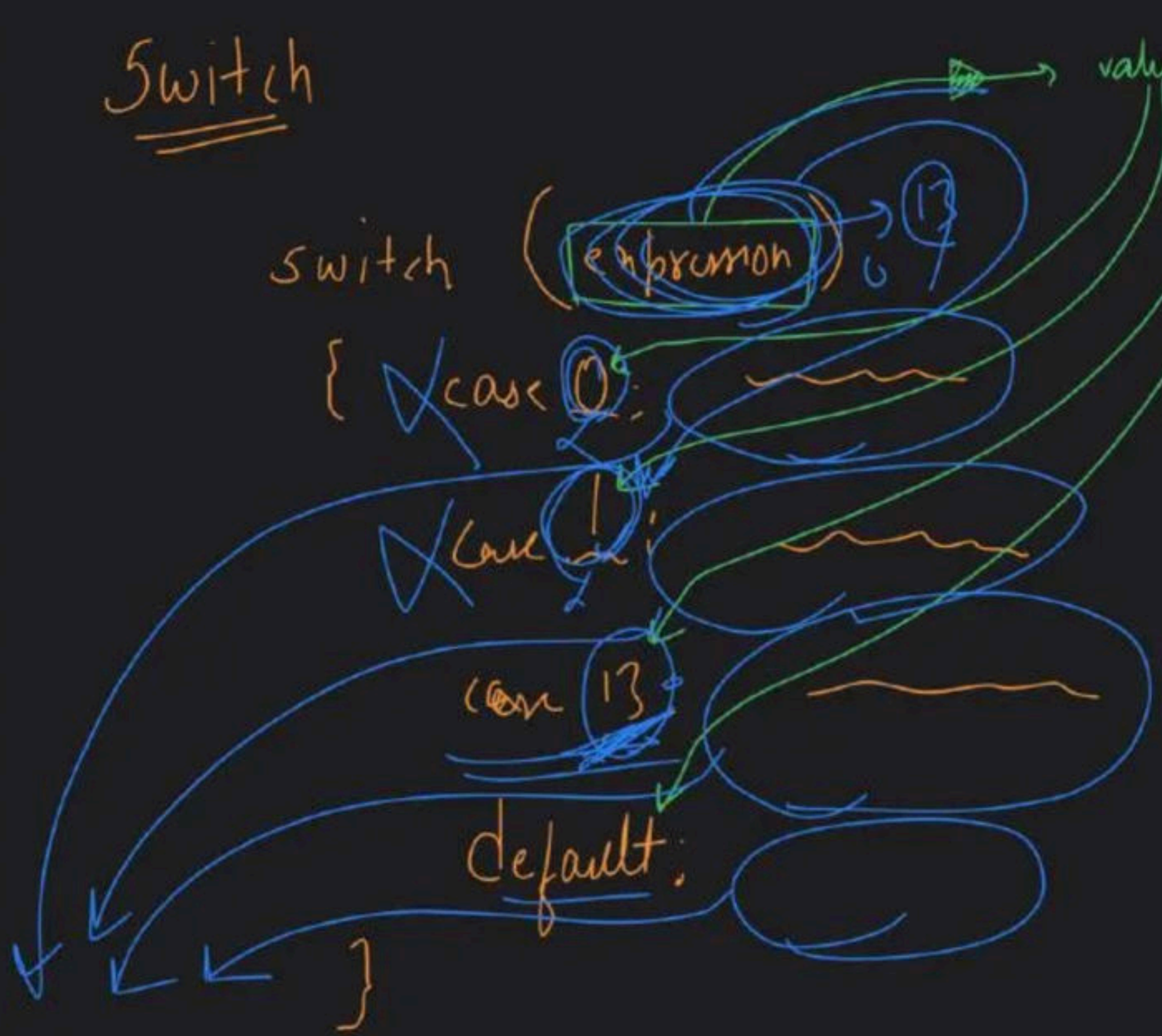
Conditionals:

- Basic if block
- If-else block
- If-elseif-else block
- Nested if else
- Switch case
- Ternary Operator



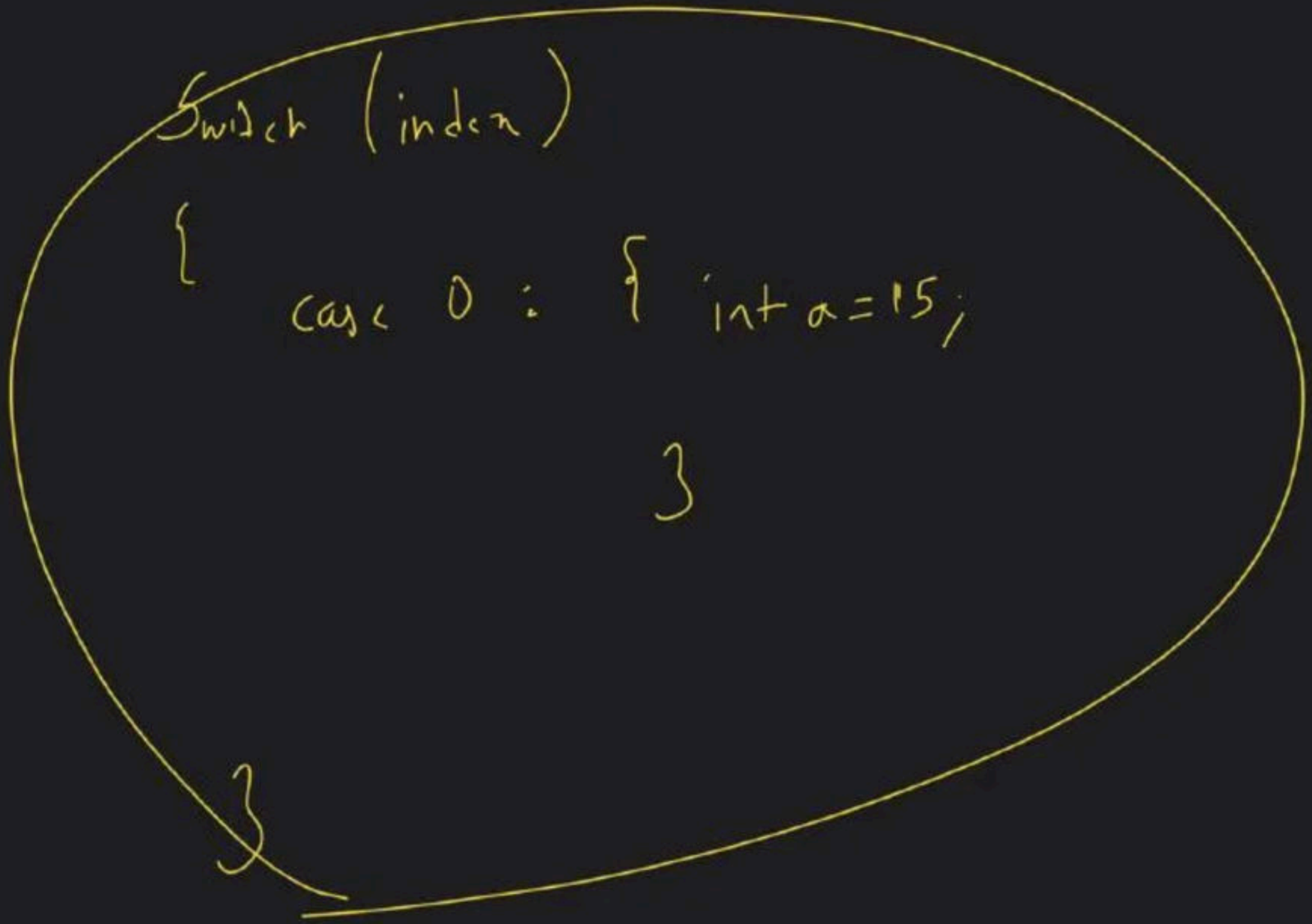


Switch



1, 2, 3, 4, 5, 6, 7

```
switch (index)
{
case 1:
    cout << "Monday";
case 2:
    cout << "Tuesday";
default:
    cout << "Sunday";
}
```


switch (index)
{
case 0 : { int a=15;
}
}

Switch

Switch (expression) → kya kya → Rules

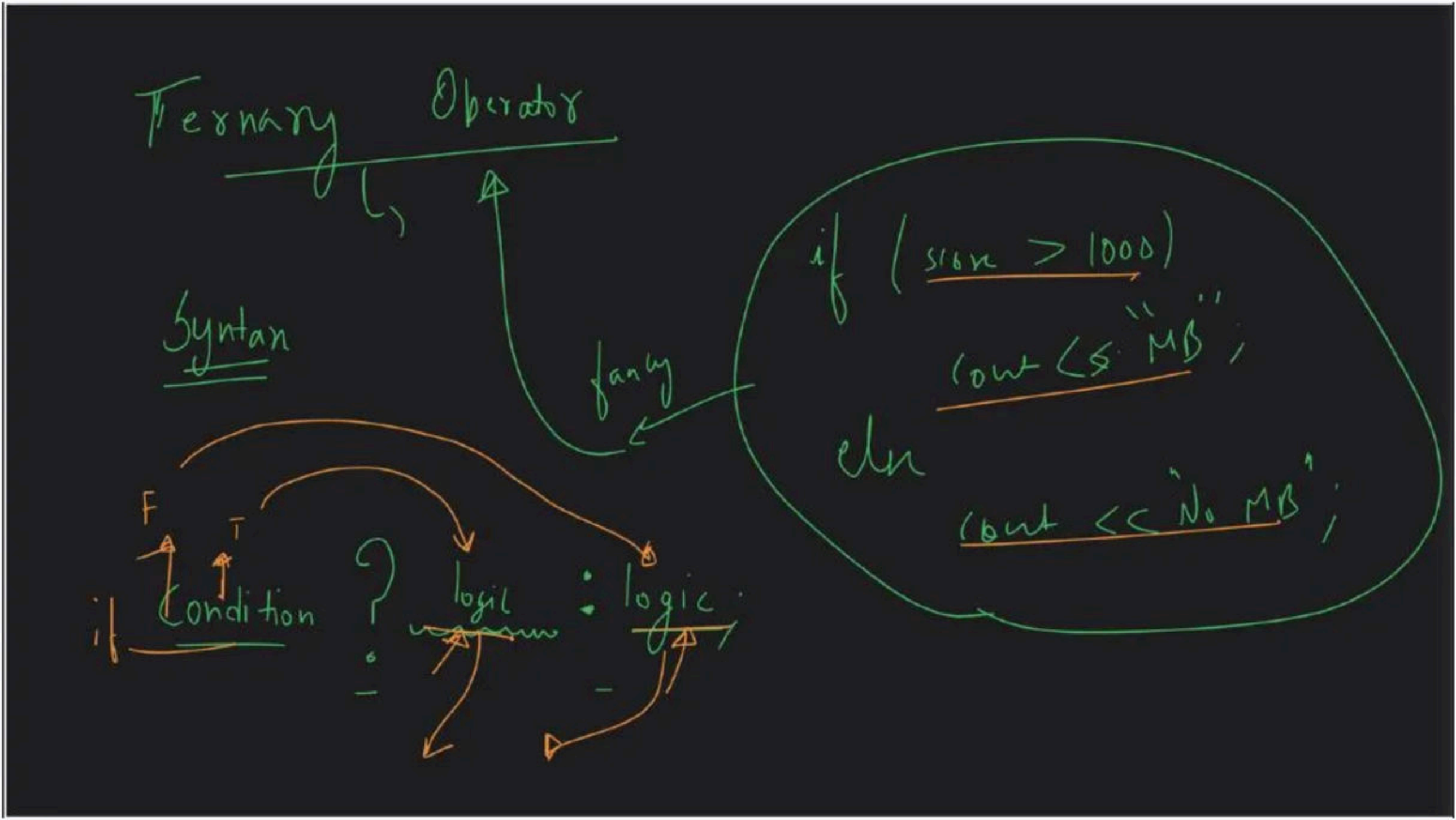
case

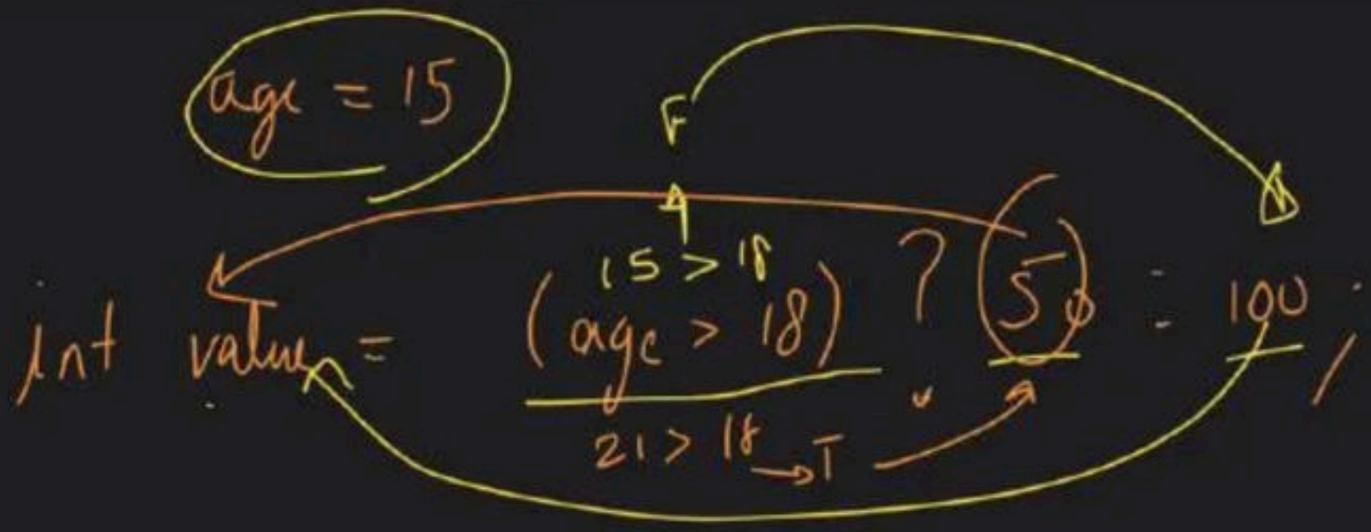
→ Rules

Learn C++

Switch case → Video

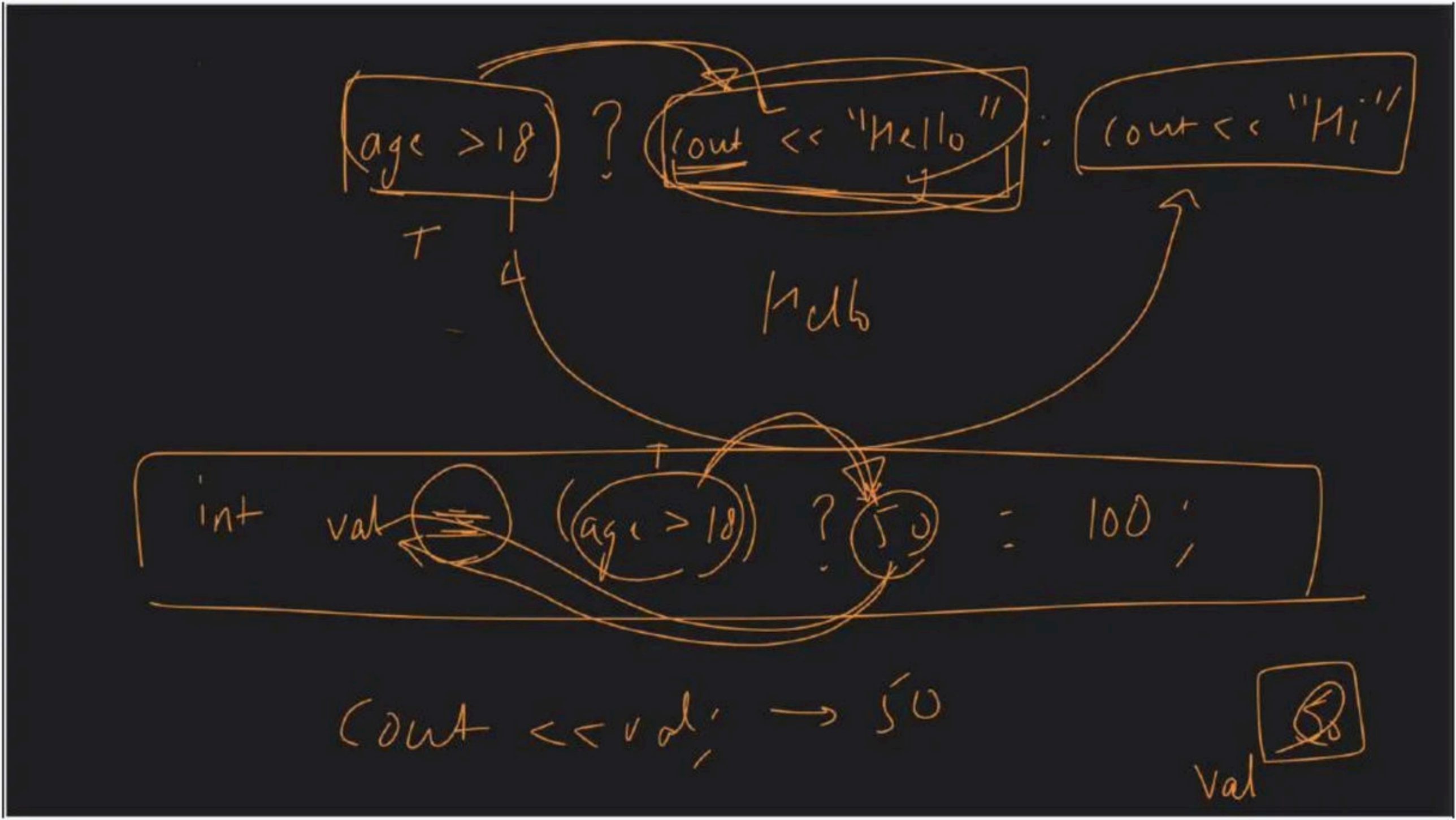
Ternary Operator



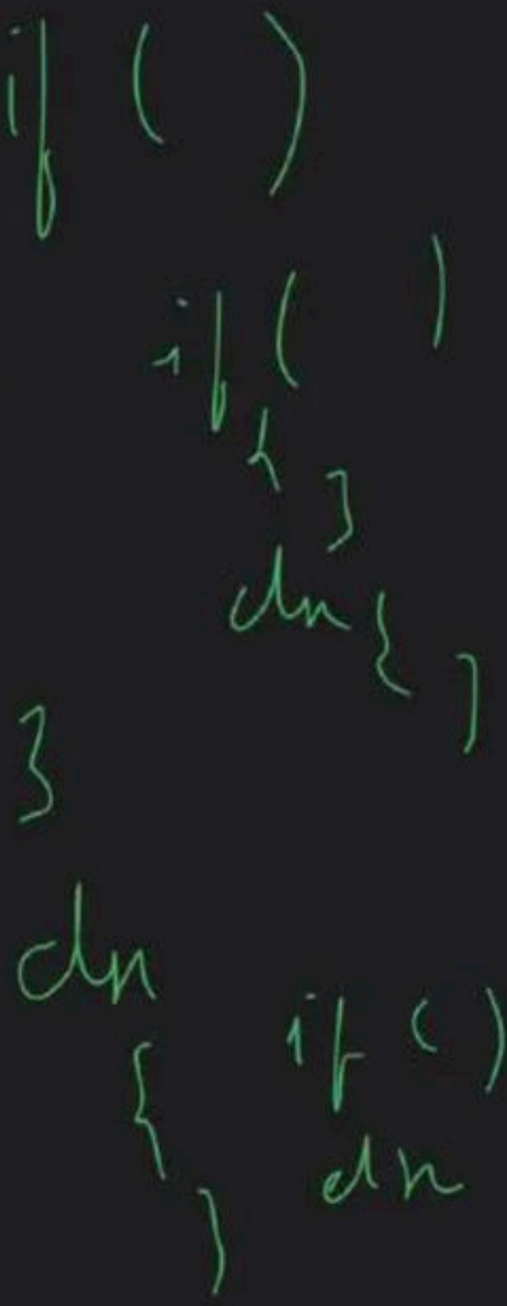


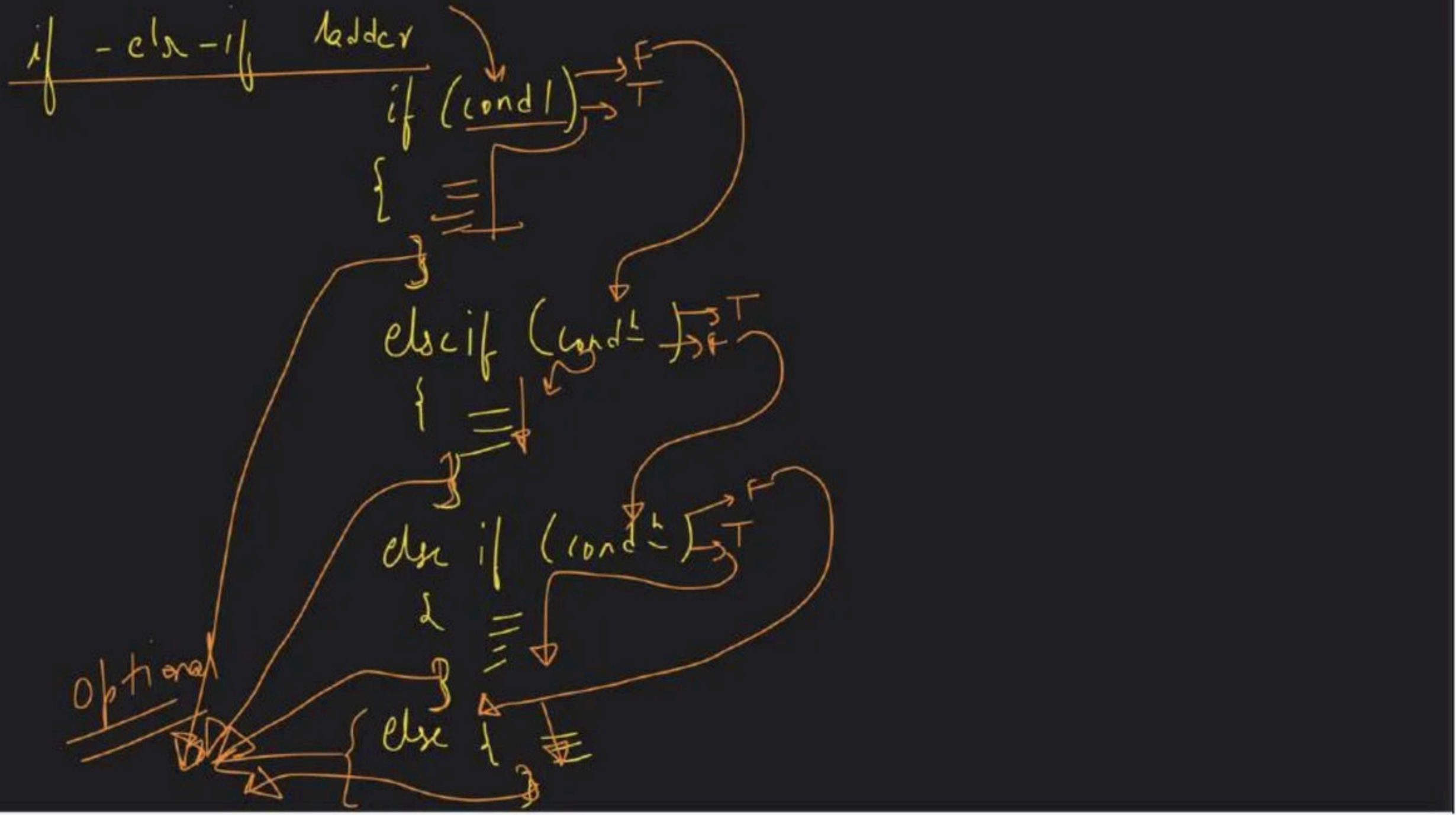
age = 21

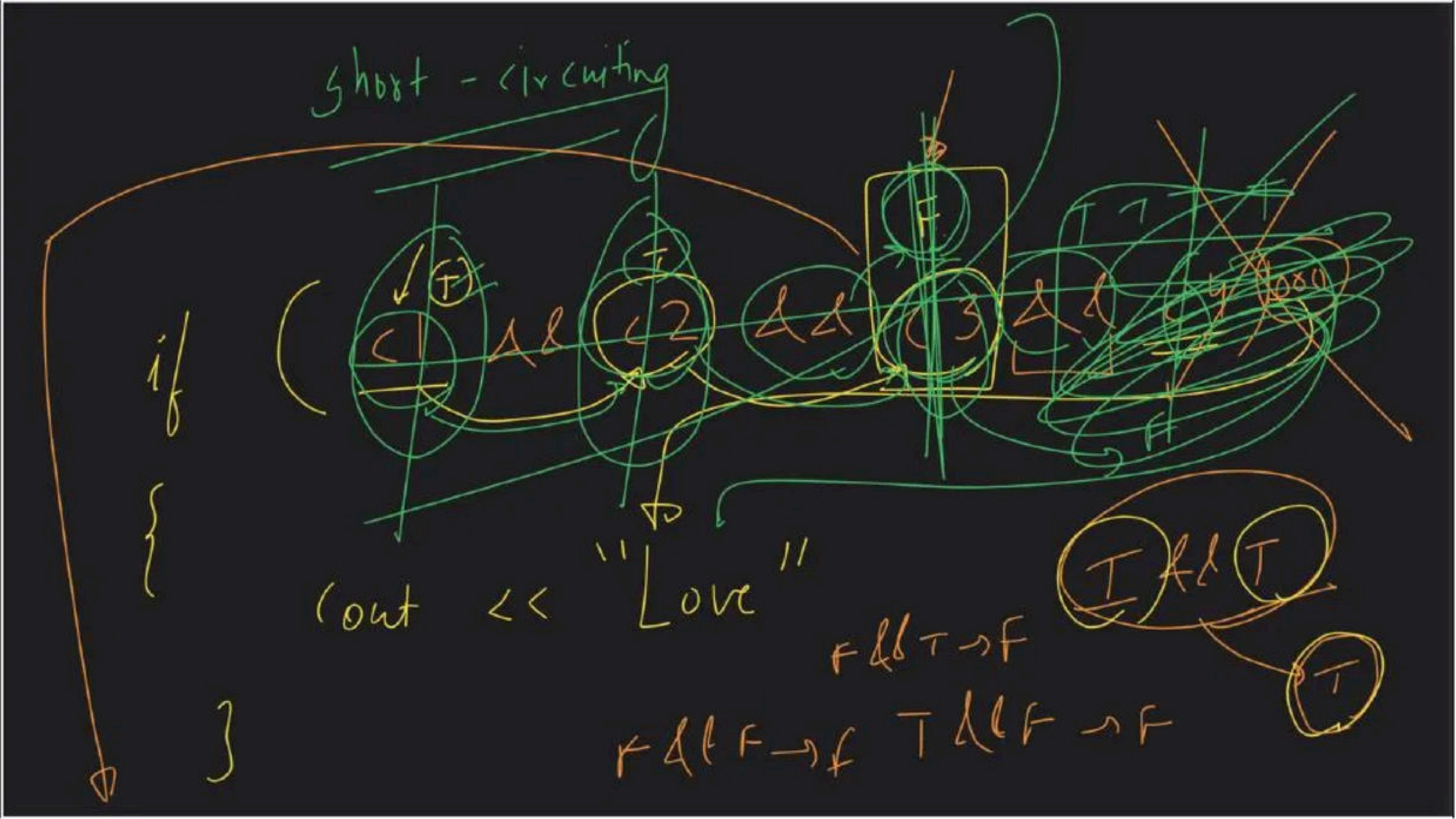
void value


age >18 ? cout << "Hello" : cout << "Hi"
T
F
Hello
int val (age >18) ? 50 : 100;
cout << val; → 50
val
50

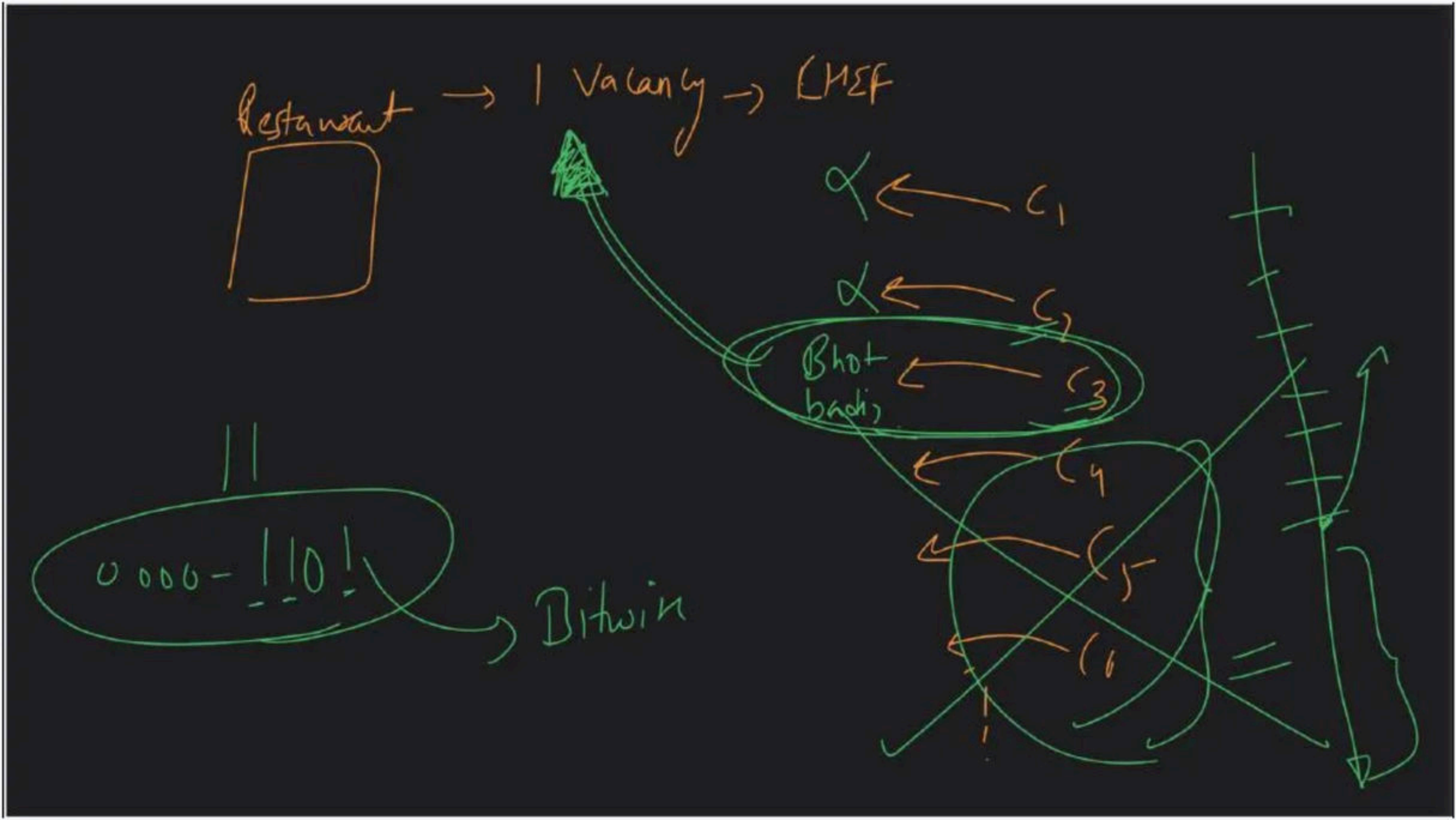

if ( )
{ if ( )
{ if ( )
{
}
}
3

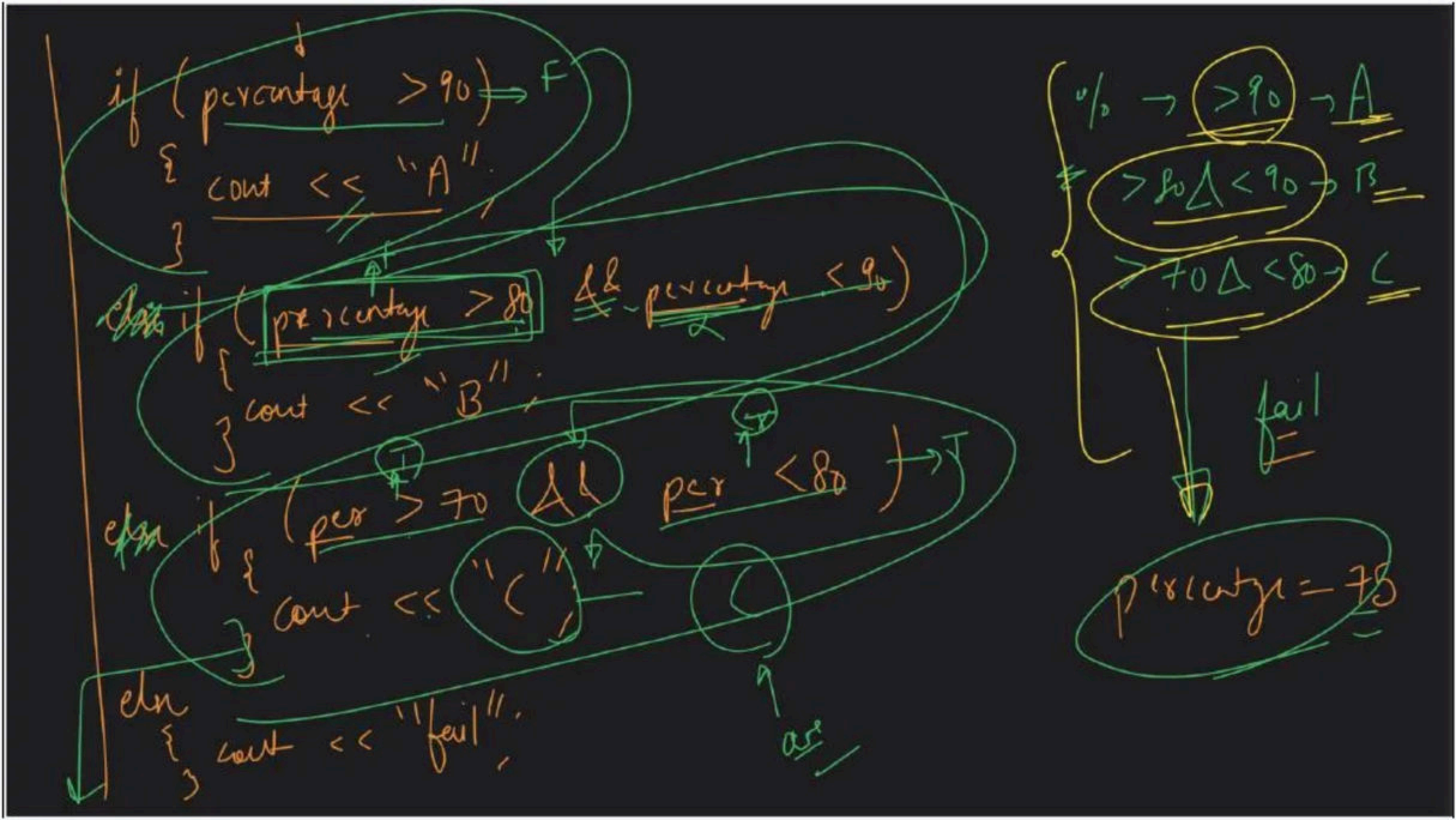

if ( )
{ if ( )
{
}
else {
}
3
else
{ if ( )
{
} else

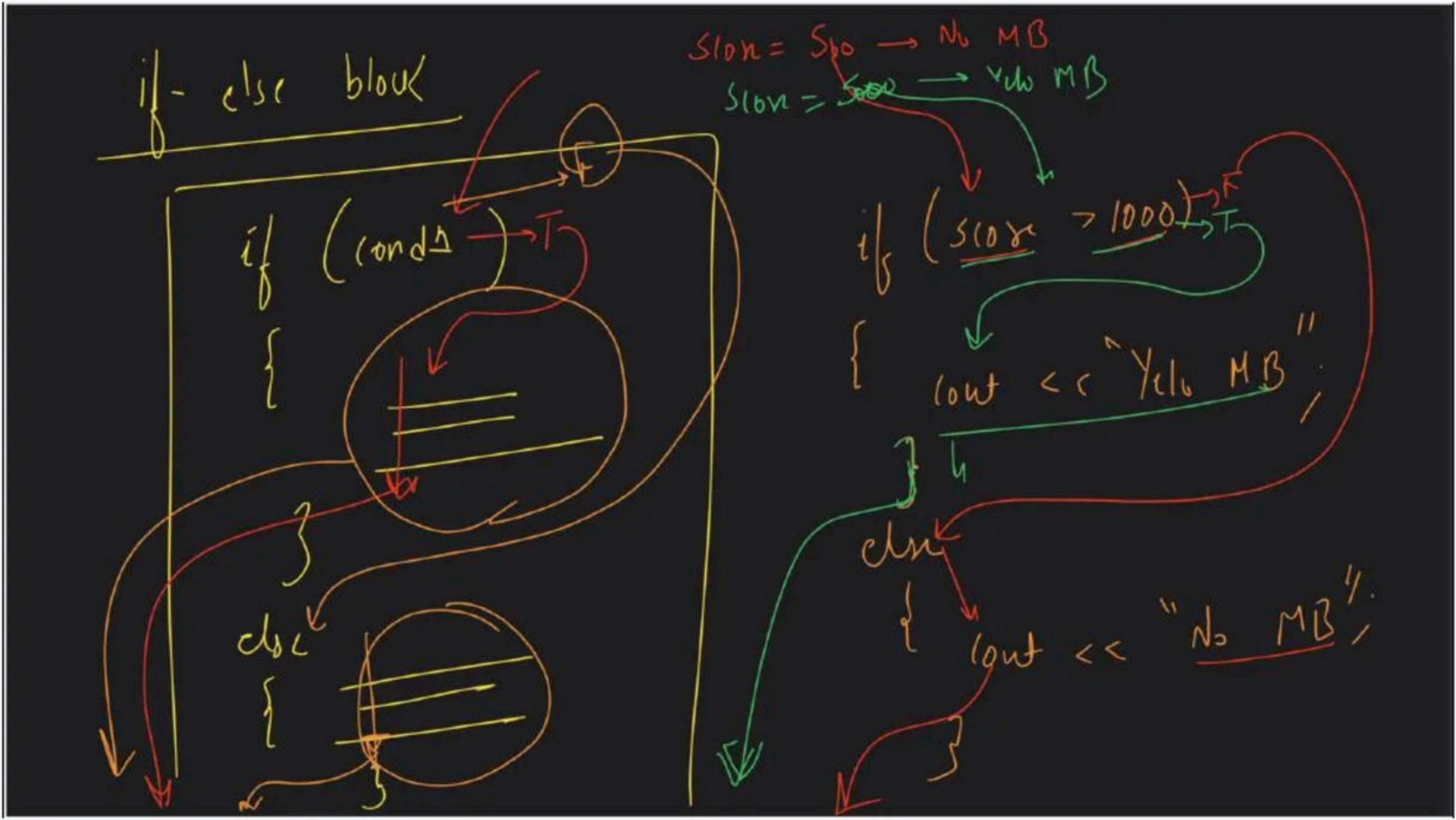

if - else - if ladder
if (cond1)
F
T
{
elseif (cond2)
F
T
{
else if (cond3)
F
T
{
Optional
else
{



short - circuiting
if
(c1 && c2 && c3 && c4)
{
cout << "Love"
}
T && T
F && T → F
F && F → F
T && F → F


Restaurant → 1 Vacancy → CHEF
Bhot badia
Bitwin

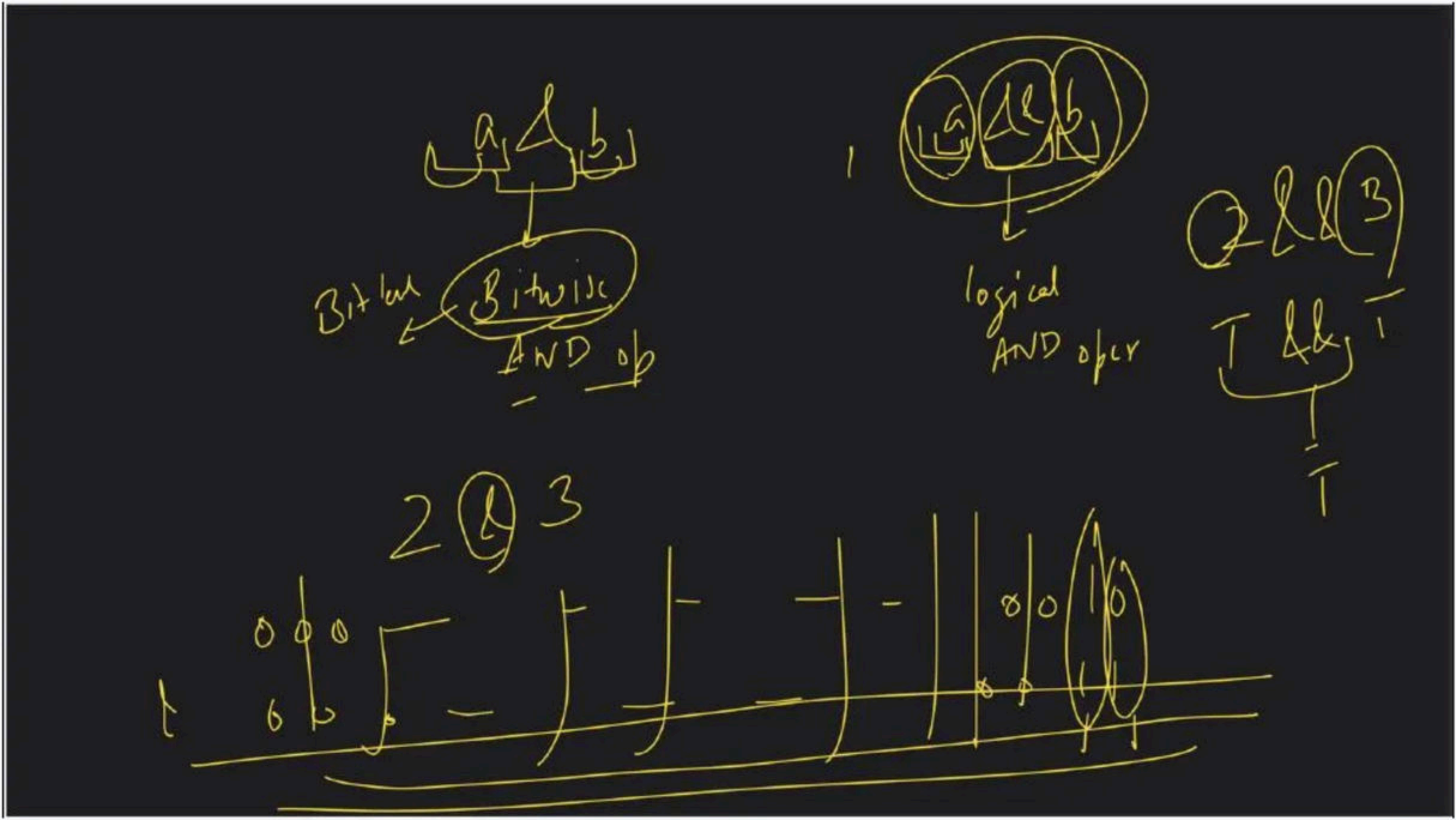

if (percentage >90) → F
{ cout << "A";
}
else if (percentage >80 && percentage <90)
{ cout << "B";
}
else if (per >70 && per <80) → T
{ cout << "C";
}
else
{ cout << "fail";
}
%
→ >90 → A
→ >80 & <90 → B
→ >70 & <80 → C
fail
percentage = 75

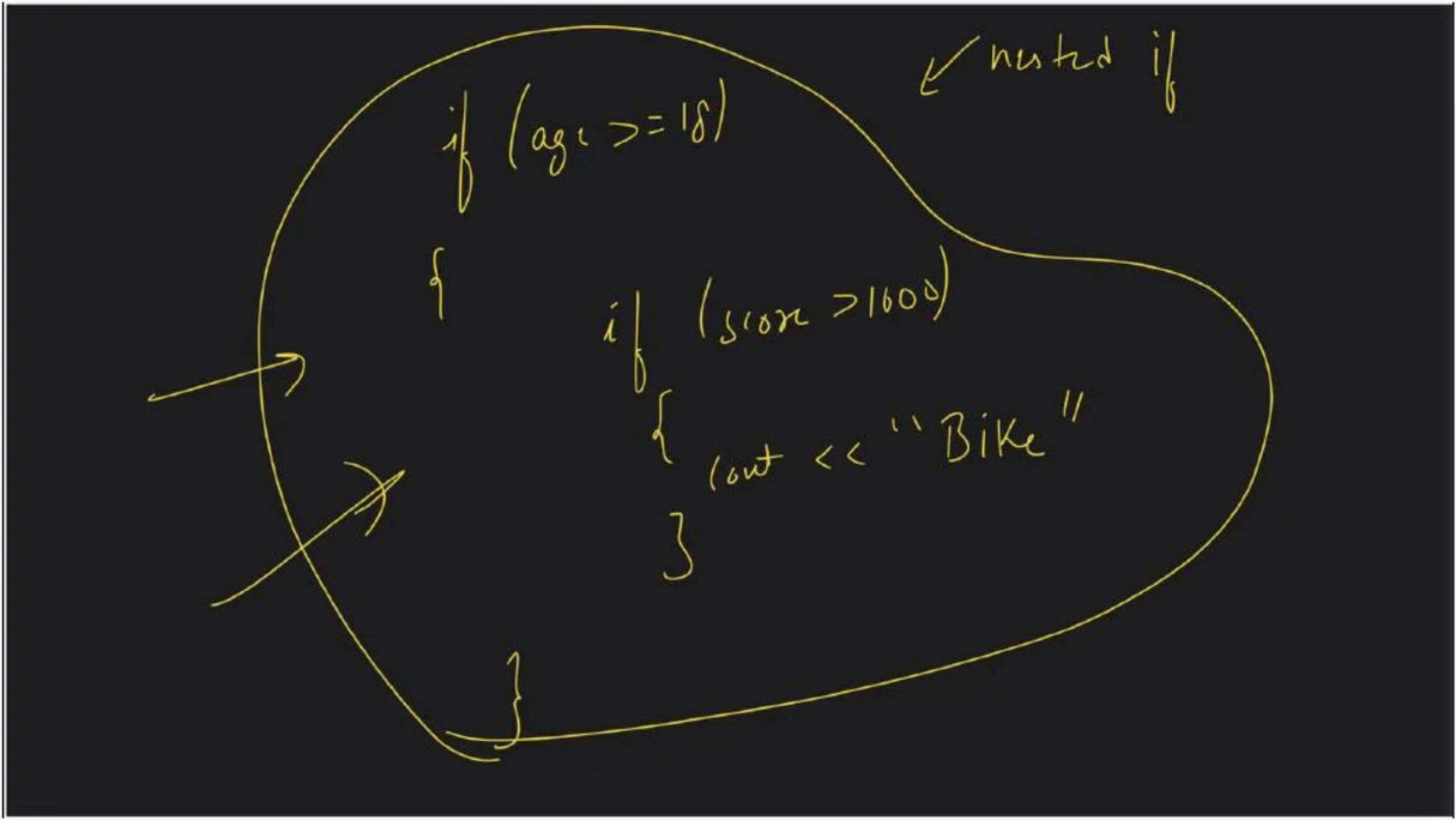

if - else block
if (cond) → T
{
}
else
{
}
size = 500 → No MB
size = 5000 → Yes MB
if (size > 1000) → T / F
{
cout << "Yes MB";
}
else
{
cout << "No MB";
}

```
if (age >= 18)
{
cout << "Yelo Lic";
}
```

```
if (score > 1000)
{
cout << "Yelo MacBook";
}
```

```
if (age >= 18 && score > 1000)
{
cout << "Bike";
}
```


Bitwise
AND op
Bitwise
logical
AND oper
T && T
T
T
2 & 3
0 0 0
0 1 0

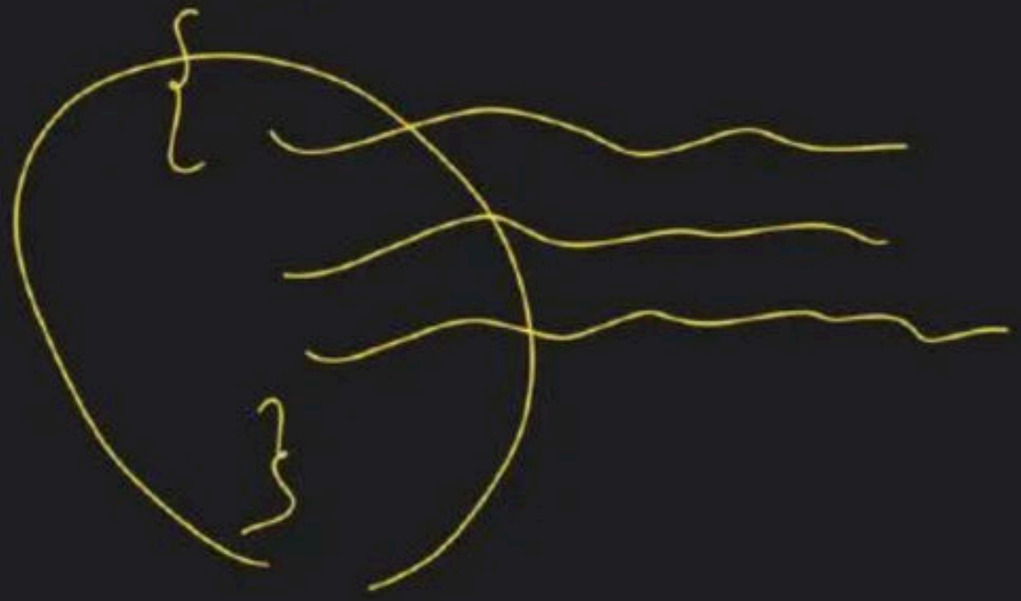

nested if
if (age >= 18)
{
if (score > 1600)
{
cout << "Bike"
}
}

Basic - if Block

multiple / join -> operation

Syntax ->

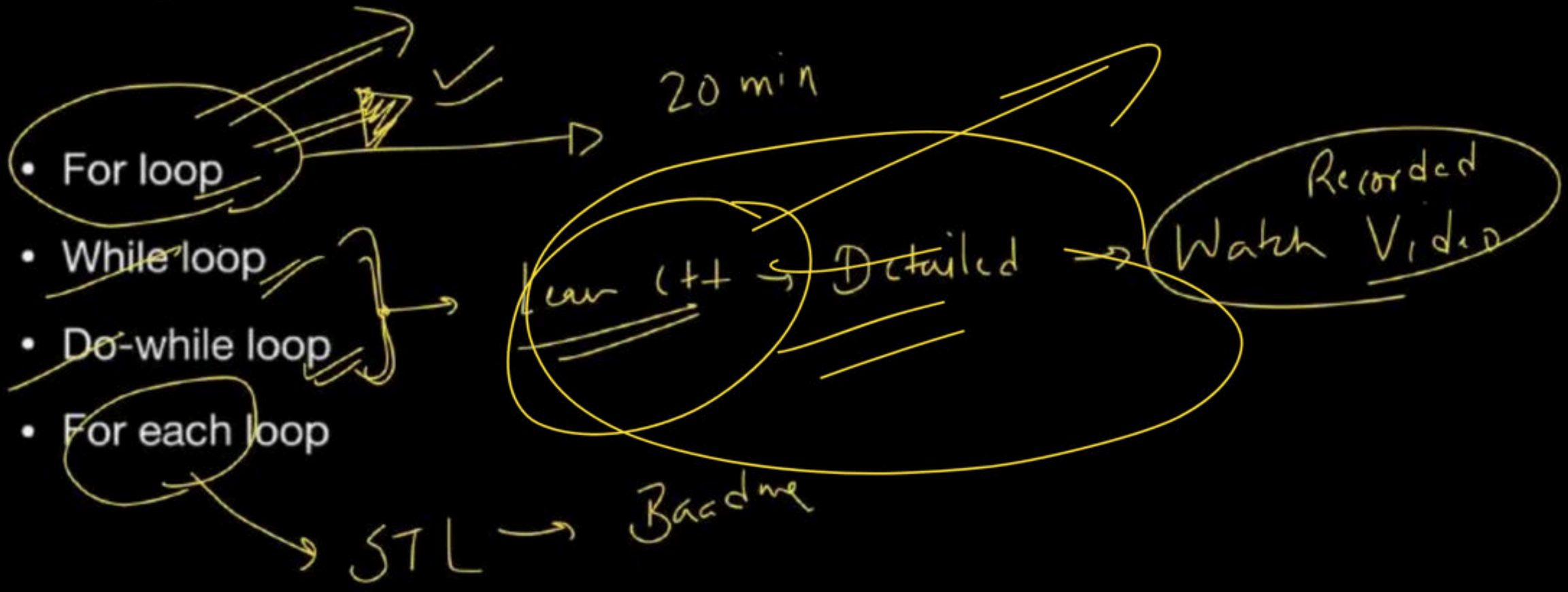
if (condition)

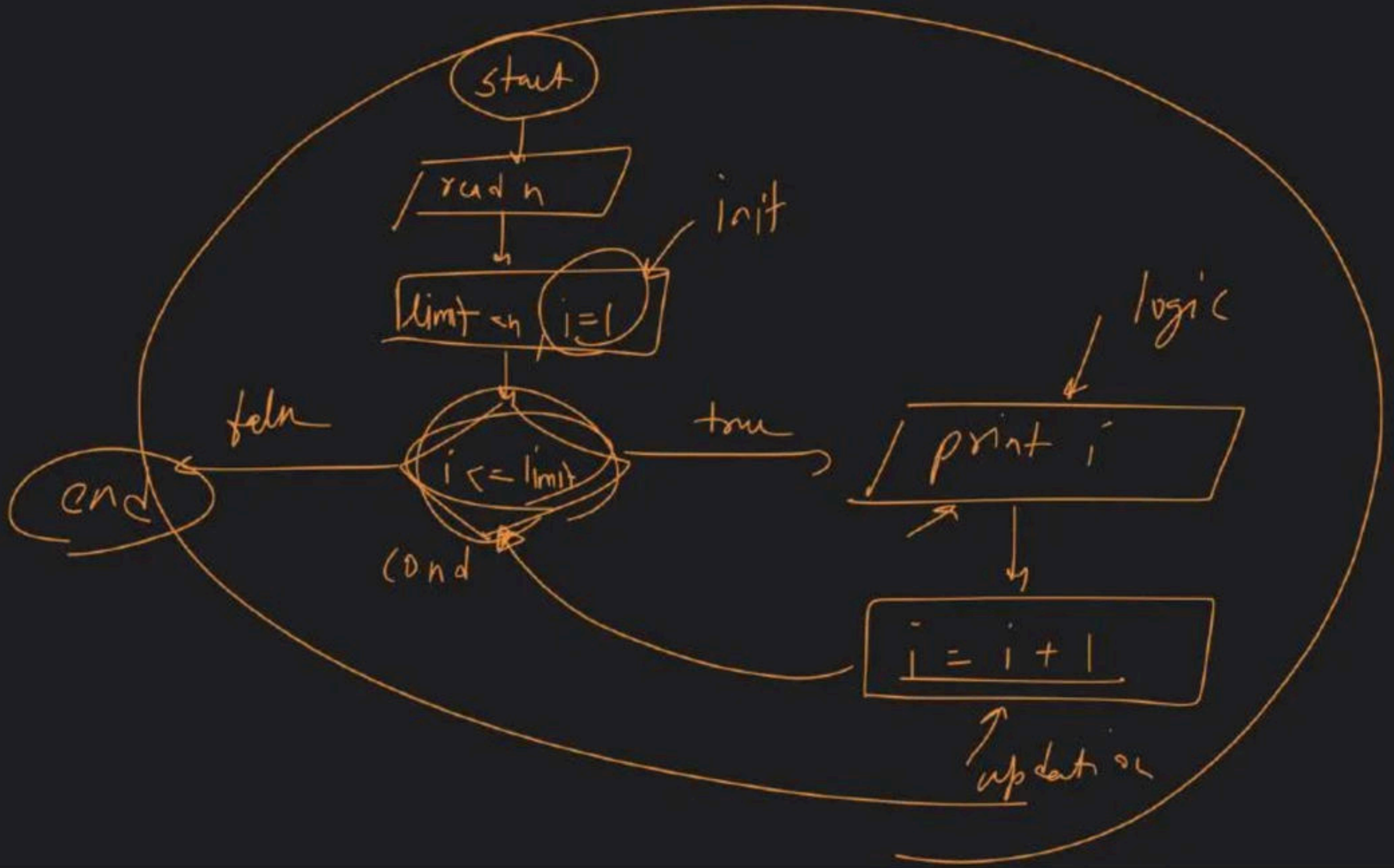


Variable Scoping

Loops:

- For loop
- While loop
- Do-while loop
- For each loop




start
read n
init
limit = n
i = 1
logic
false
true
print i
end
i <= limit
cond
i = i + 1
update

→ For Loop →

Syntax

n=5

limit=5

```
for ( int i=1 ; i<=limit ; i=i+1 )
{
    cout << "Hi" << endl;
}
```

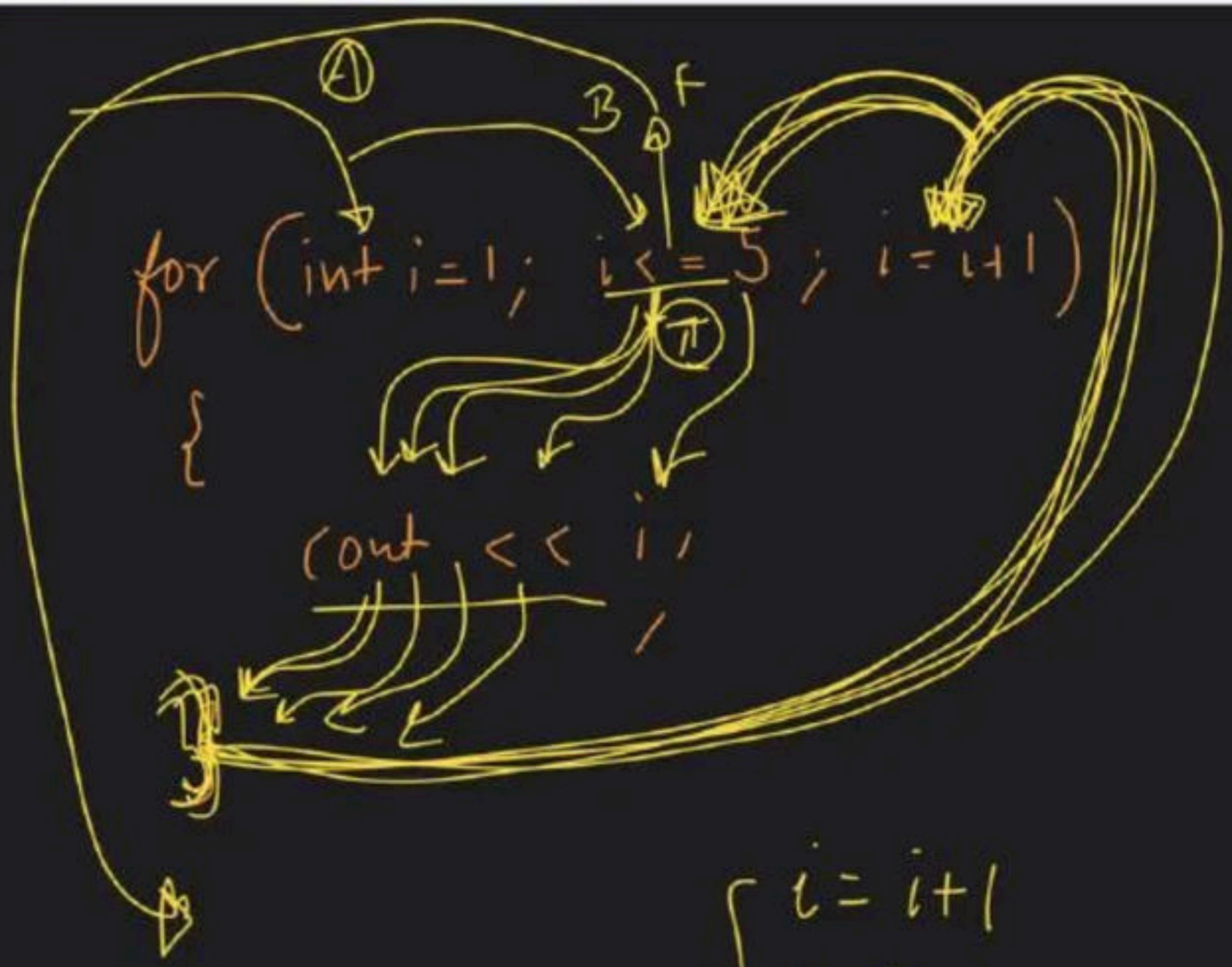
int i=1 : initialisation

i<=limit : Condition(s)

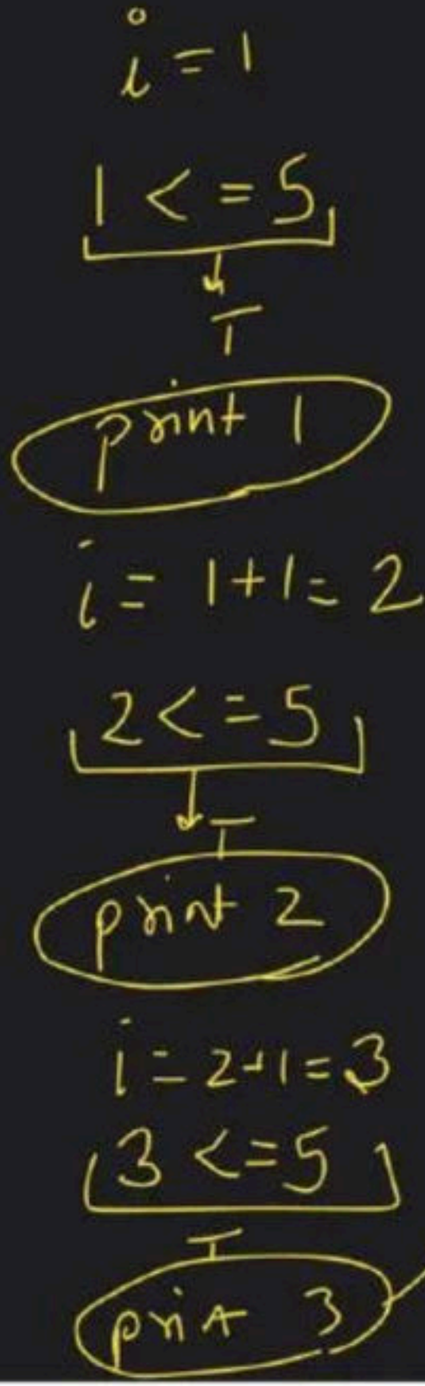
i=i+1 : updation

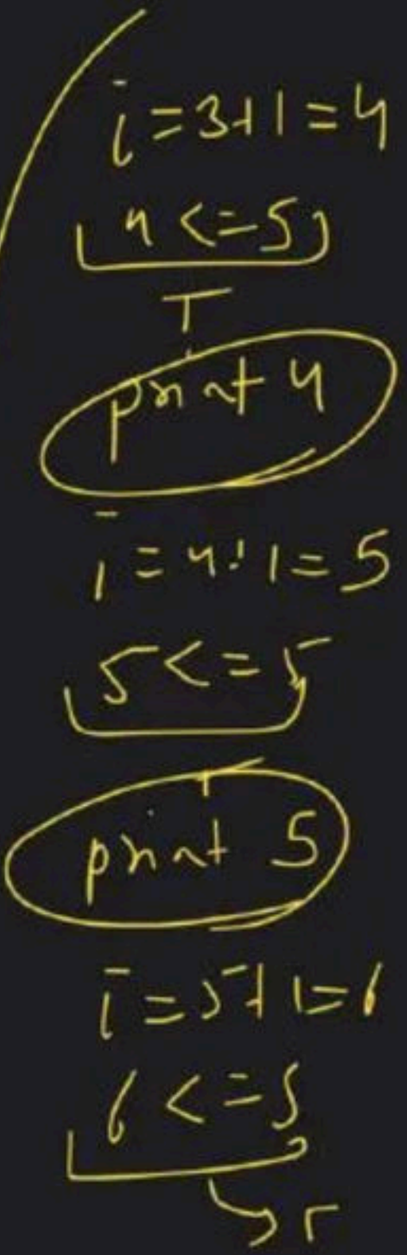
++i

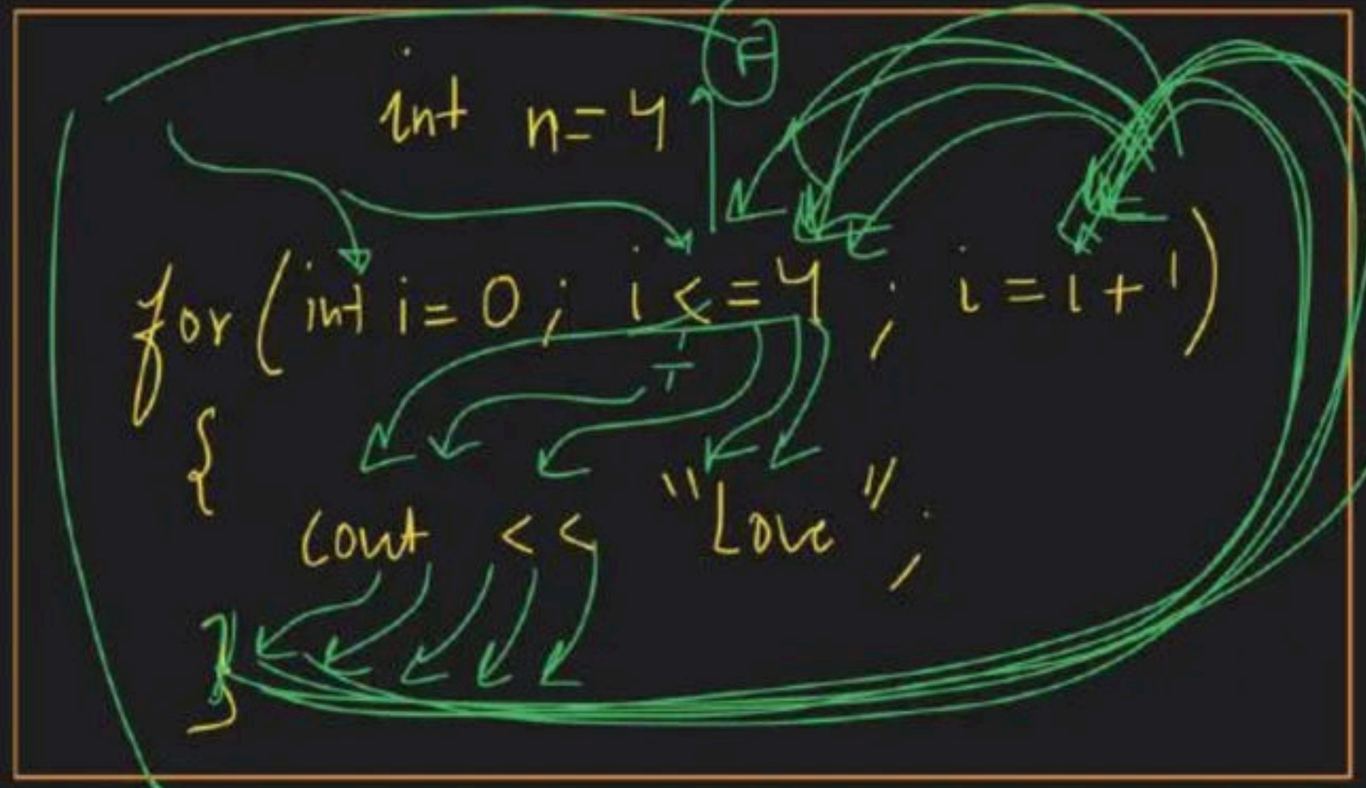
i=i+1


A
B
F
for (int i=1; i<=5; i=i+1)
T
{
cout << i;
}

- $i = i+1$
- or
- $++i$
- or
- $i++$


i=1
1<=5
T
print 1
i=1+1=2
2<=5
T
print 2
i=2+1=3
3<=5
T
print 3


i=3+1=4
4<=5
T
print 4
i=4+1=5
5<=5
T
print 5
i=5+1=6
6<=5
F



n=4

i=0

0<=4→T

Love

i=0+1=1

1<=4→T

Love

i=1+1=2

2<=4→T

Love

i=2+1=3

3<=4→T

Love

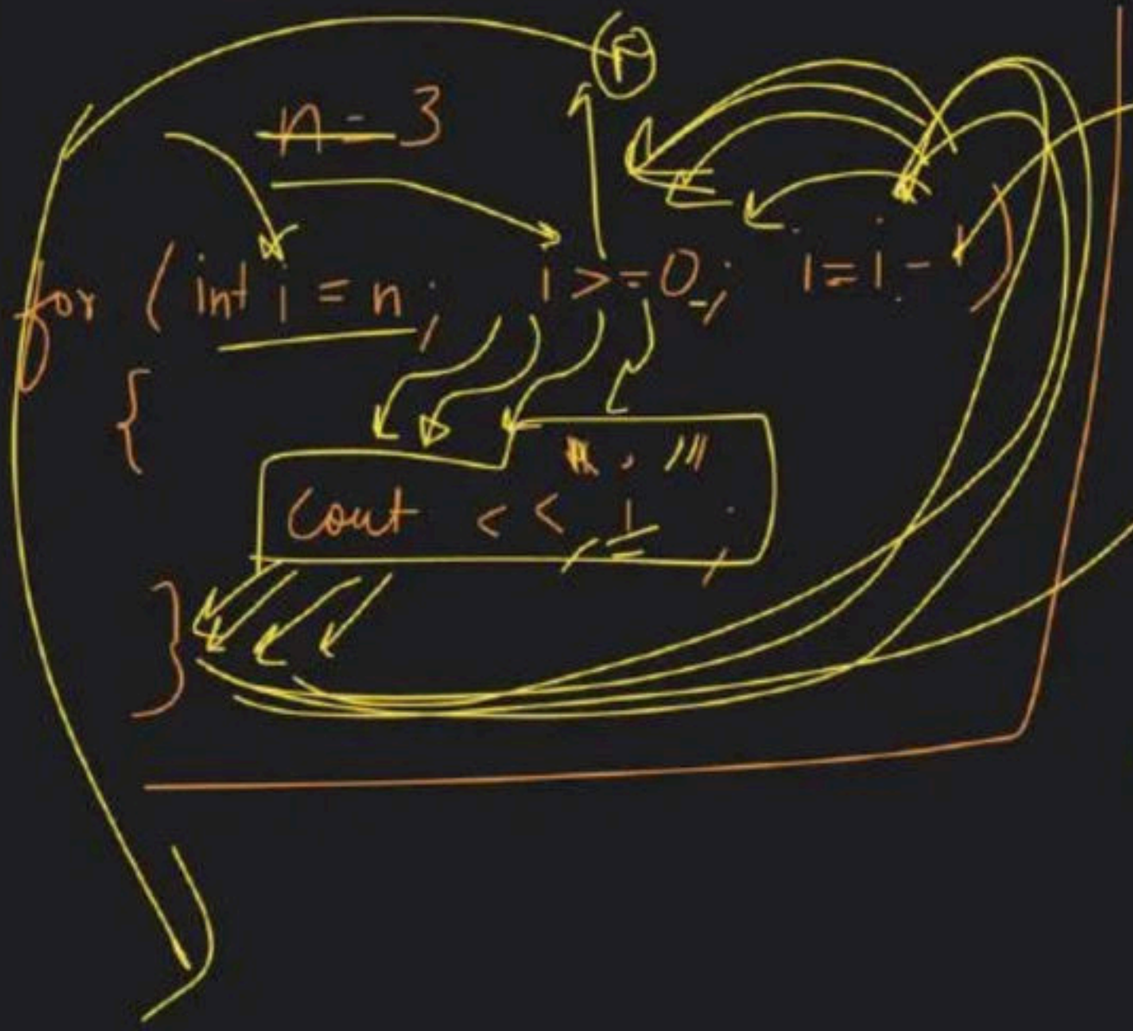
i=3+1=4

4<=4→T

Love

i=4+1=5

5<=4→f



n=3

i=3

3>=0→T

i

i=3-1=2

2>=0→T

i

i=2-1=1

1>=0→T

i

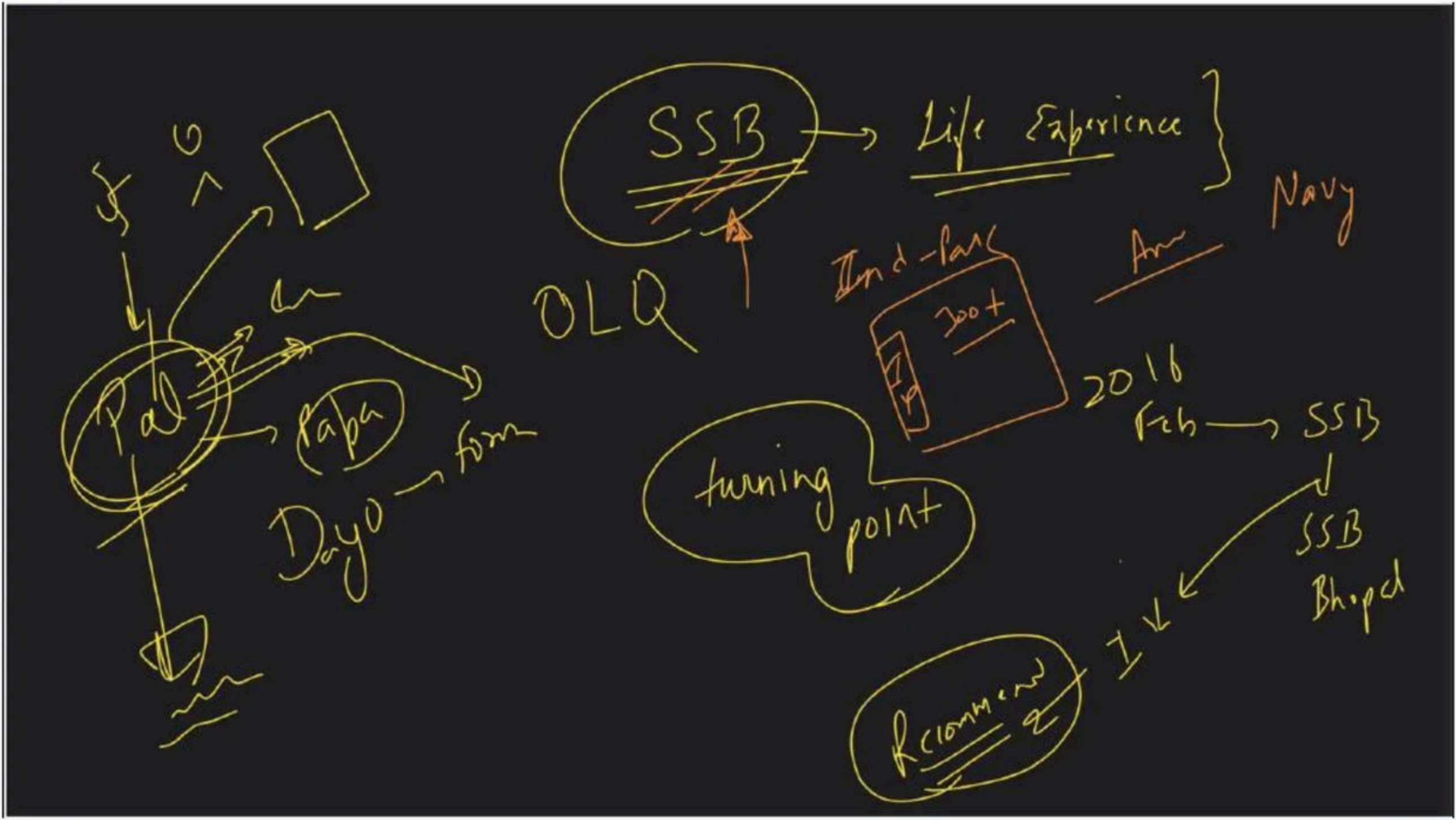
i=1-1=0

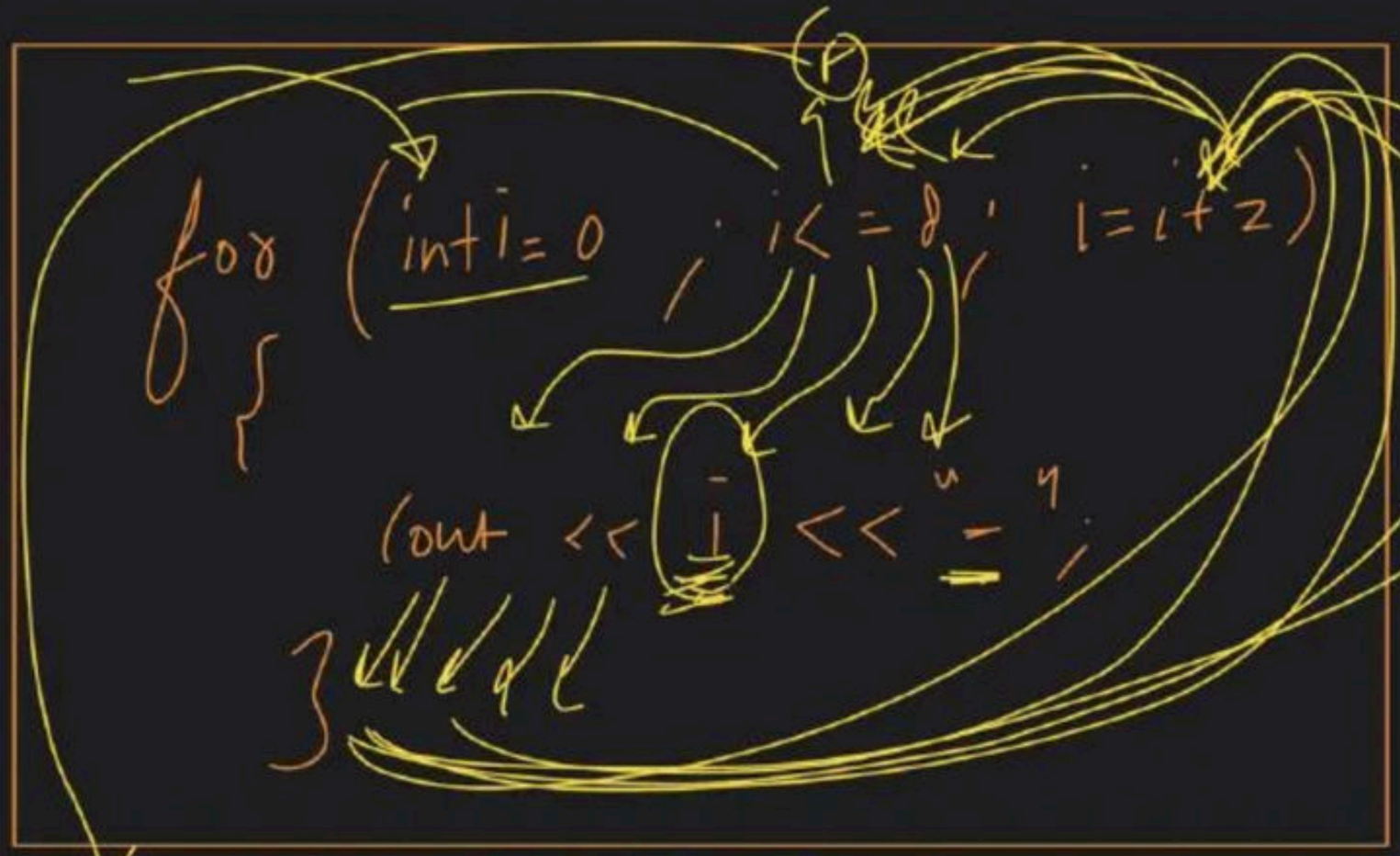
0>=0→T

i

i=0-1=-1

-1>=0→F


SSB
Life Experience
OLQ
Navy
Ind-Pak
Army
2016
Feb
SSB
SSB
Rejected
Recommend
turning
point
Papa
Dayo
form



i=0

0<=8 → T

i=0+2=2

2<=8 → T

i=2+2=4

4<=8 → T

i=4+2=6

6<=8 → T

i=6+2

=8

8<=8

→ T

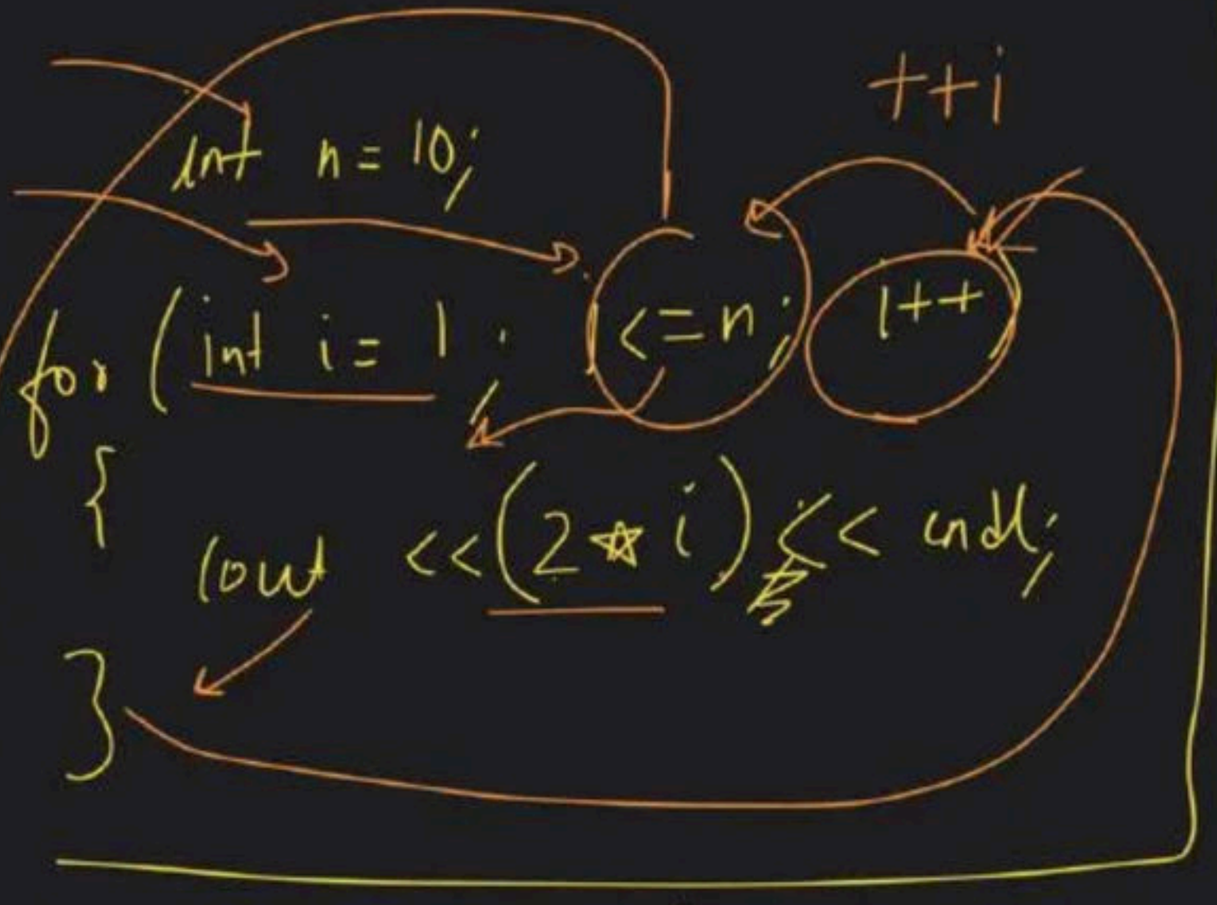
i=8+2

=10

10<=8

→ F

o/p → 0-2-4-6-8-



++i

n=10

i=1

1<=10 → T

i=2

2<=10 → T

i=3

i=4

i<5

i=6

i=7

i=9

i=i+1

or

++i

or

i++

i=10

10<=10 → T

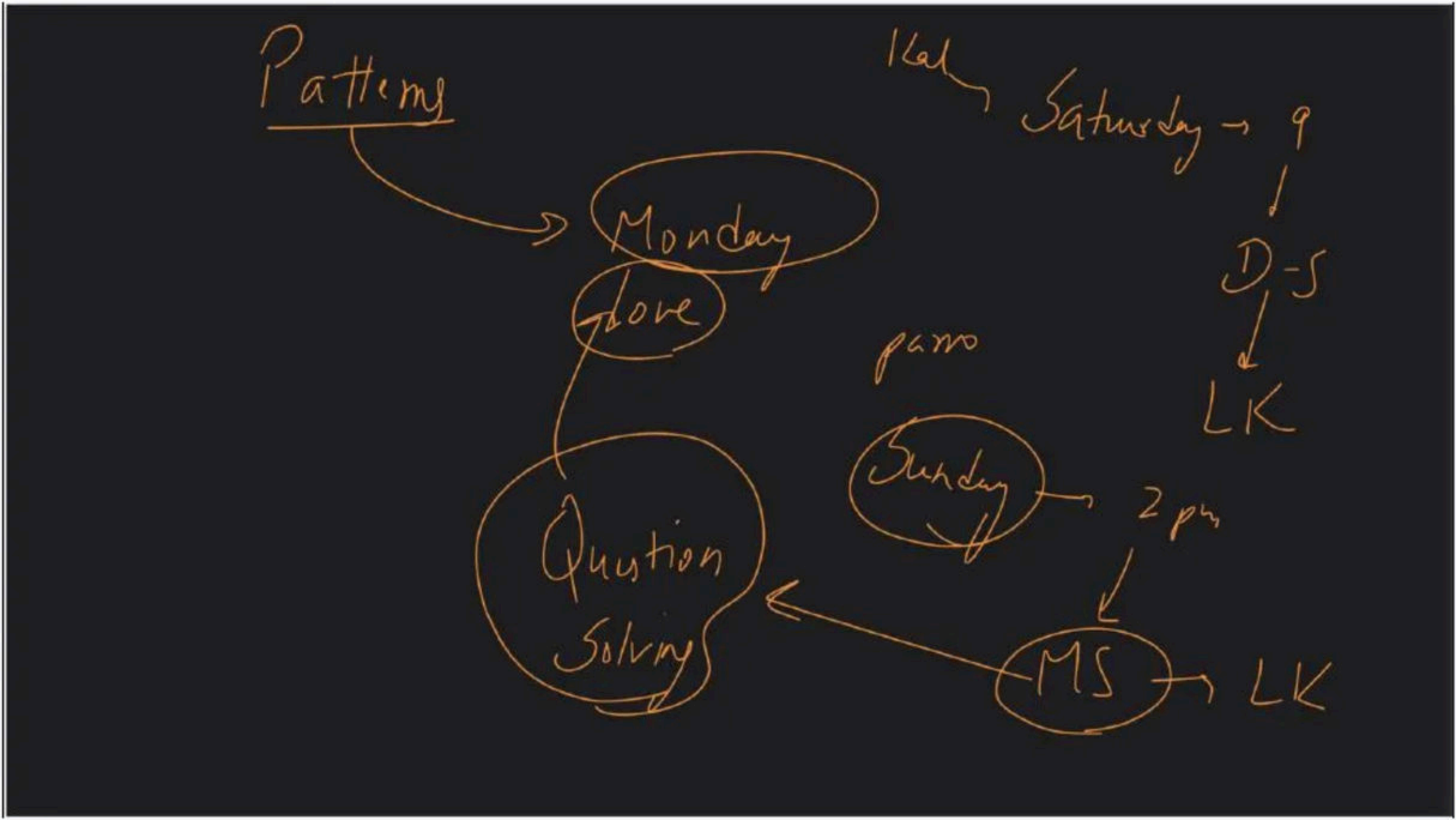
i=11

11<=10 → F

1 min

0/p → 2 4 6 8 10

12 14 16 18 20


Patterns
Monday
love
Question Solving
Sunday
2 pm
MS
LK
Saturday → 9
D-S
LK