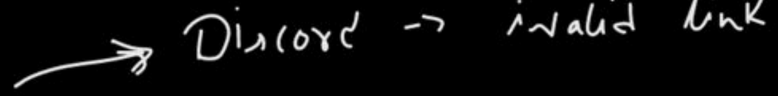
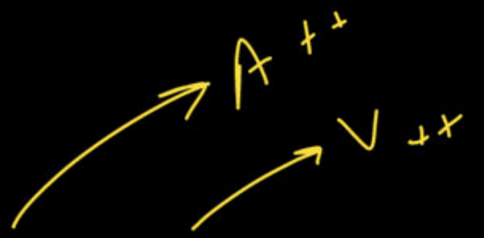
Let's solve some Patterns

Special class

Love Babbar • Jan 29, 2023

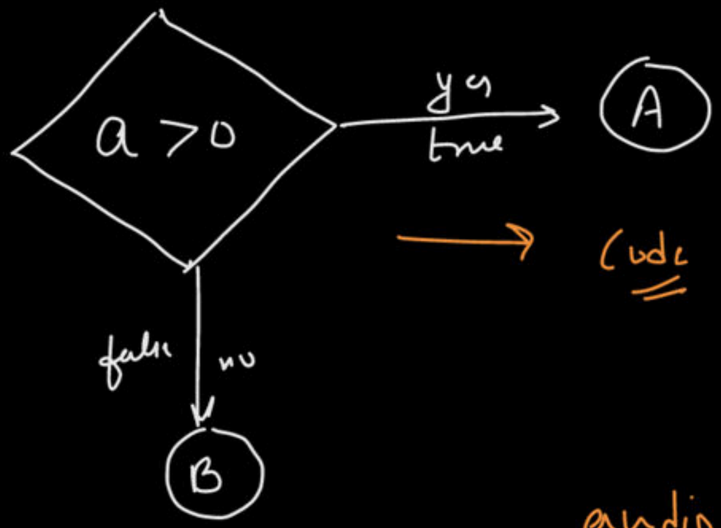




Conditionals, Loops & Patterns

Instructor: Love Babbar

Conditionals



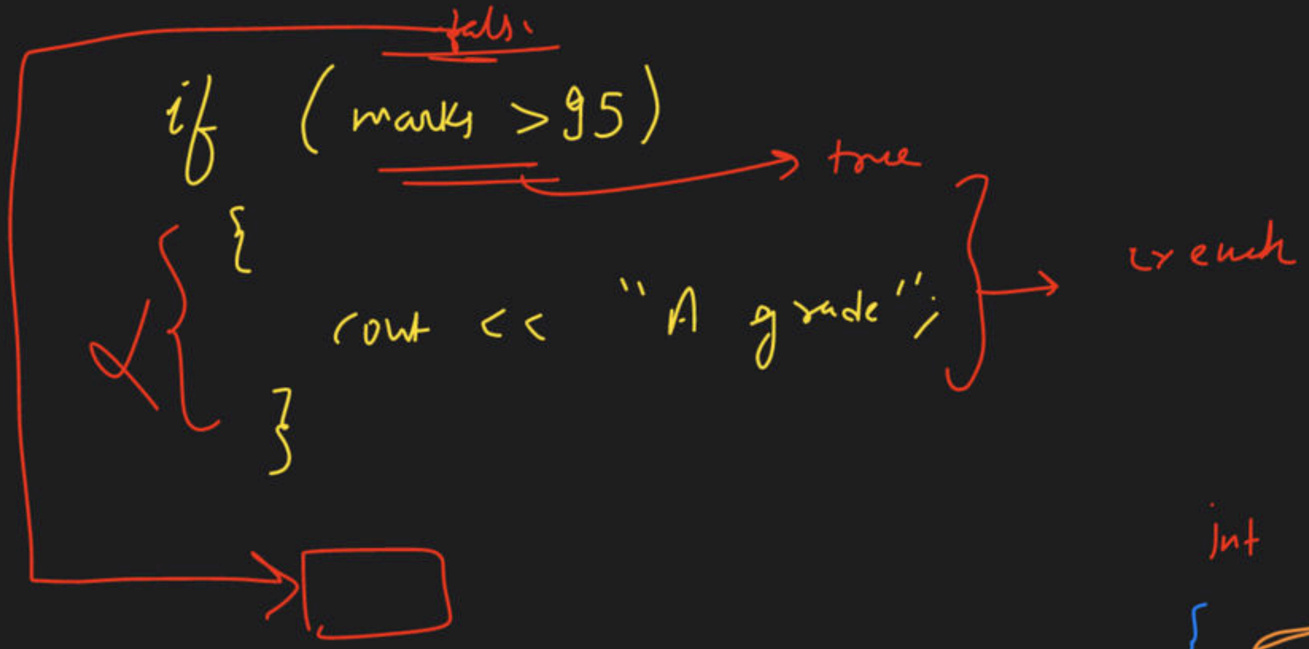
if statement

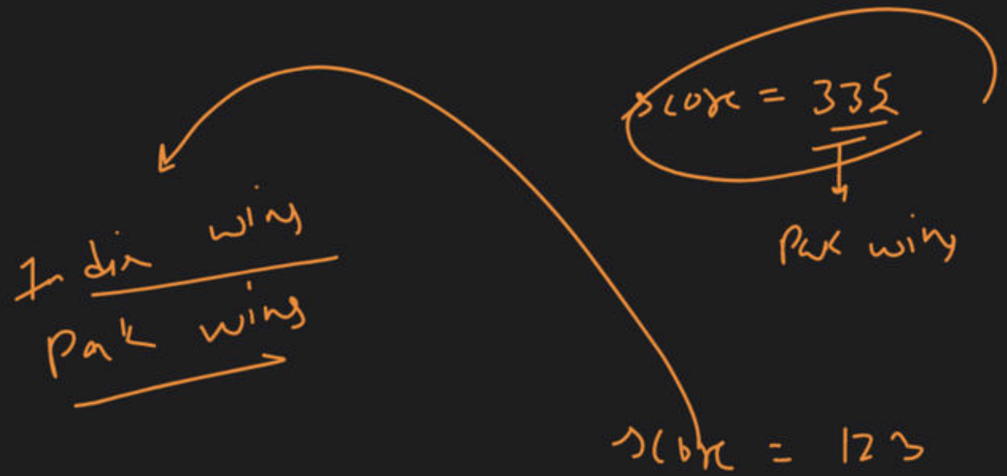
if - else

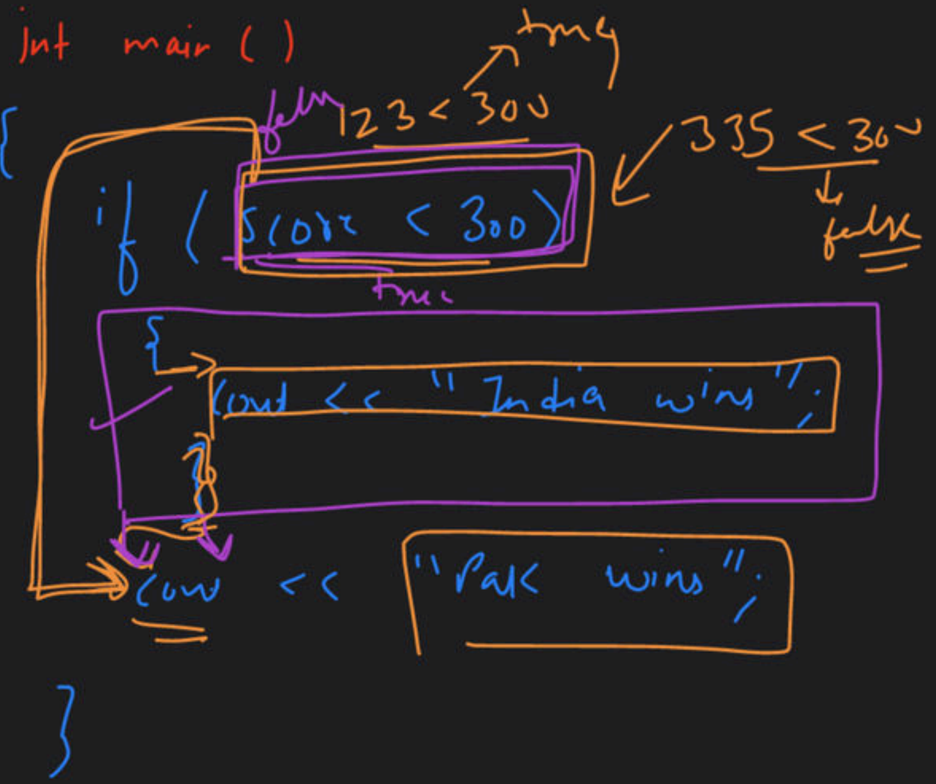
```
if ( condn )
{
    execute this
}
```

audio issue → Reload & Rejoin

```
if ( age > 17 )
{
    you are eligible to vote
}
```


false
if (marks > 95)
true
{
cout << "A grade";
}
Execute


score = 335
Pak wins
India wins
Pak wins
score = 123


int main ()
{
false
123 < 300
true
335 < 300
false
if (score < 300)
true
{
cout << "India wins";
}
cout << "Pak wins";
}

① if() {
}

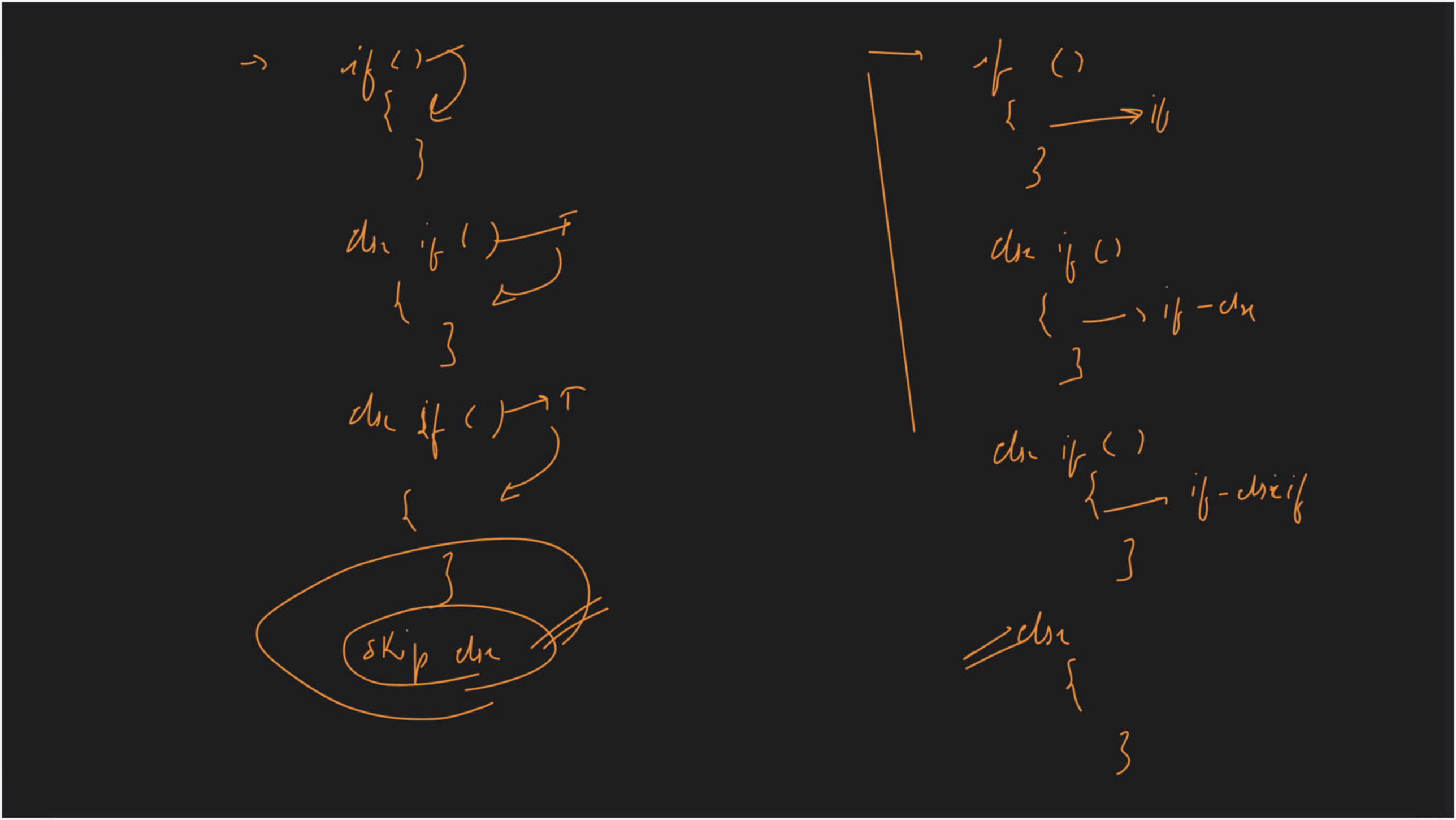
② if() {
if() { } if() { }
}

③ if - else

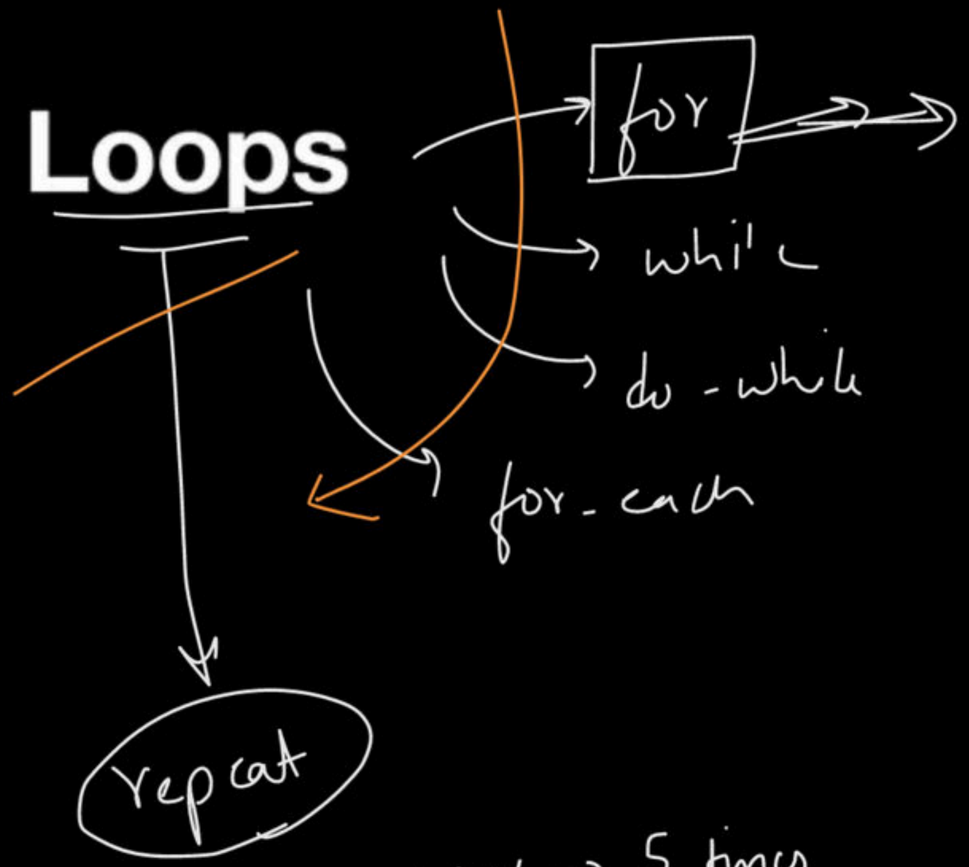
if () {
}
else {
}

>= 90 → A
>= 80 → B
>= 60 → C
>= 40 → D
< 40 → F

multiple conditions


if ( )
{
}
else if ( )
{
}
else if ( )
{
}
skip else
if ( )
{
}
else if ( )
{
}
else if ( )
{
}
else
{
}

Loops

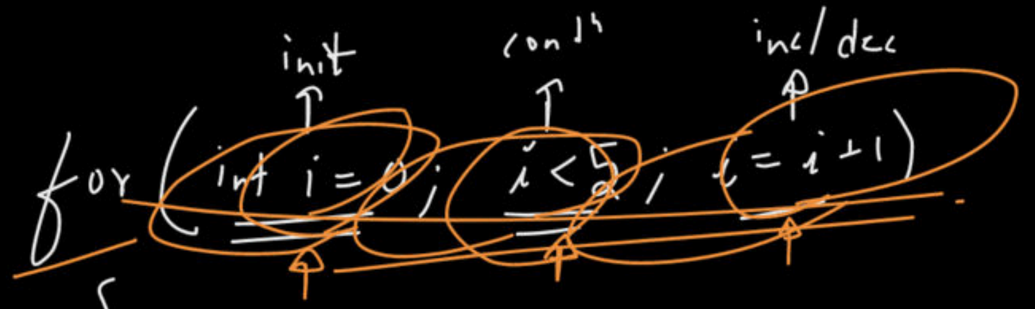


Yep cat

name -> 5 times

loop

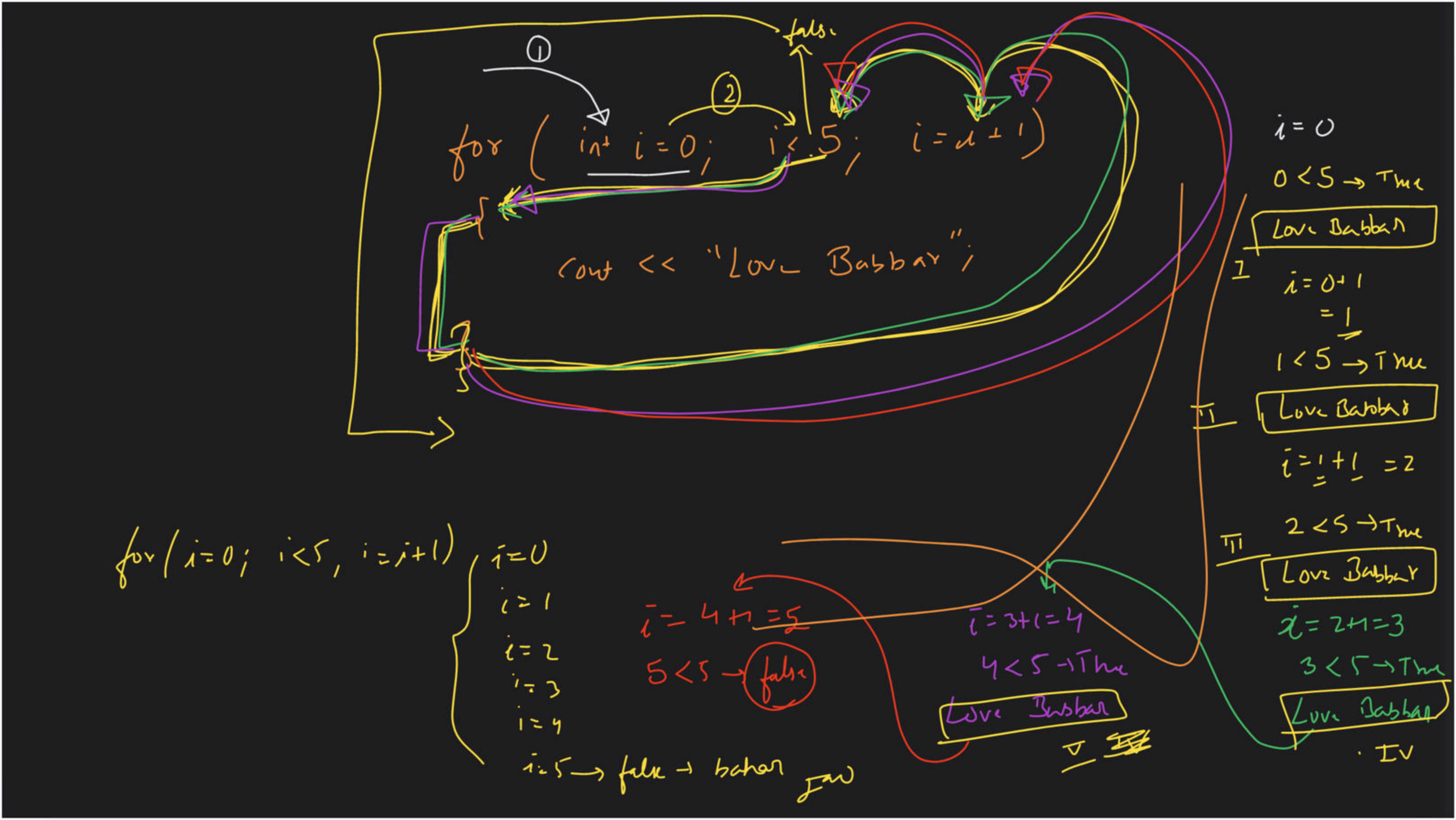
counting -> 1 -> 5

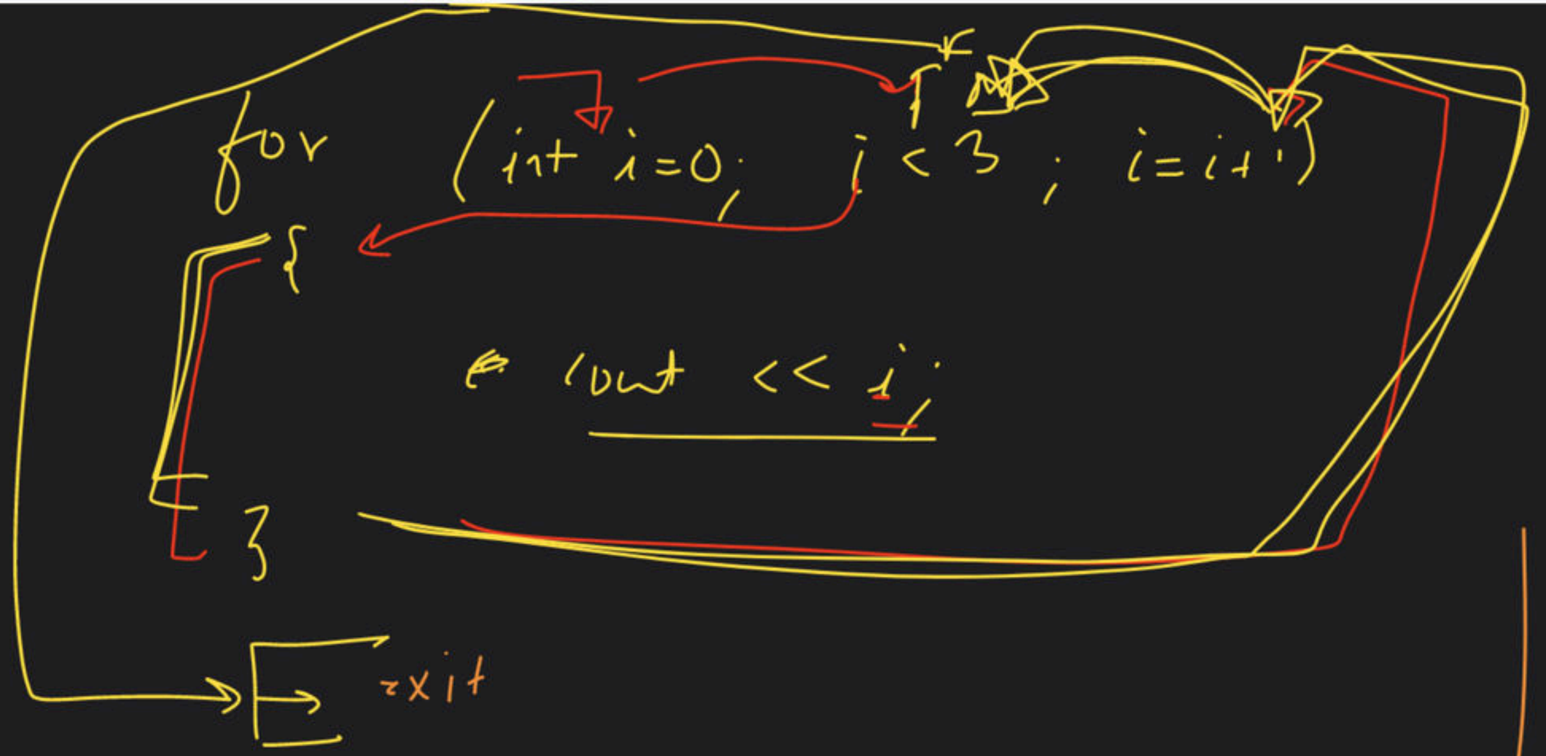


{

code

}


false
1
2
for (int i = 0 ; i < 5 ; i = i + 1)
{
cout << "Love Babbar" ;
}
i = 0
0 < 5 → True
Love Babbar
I
i = 0 + 1
= 1
1 < 5 → True
Love Babbar
II
i = 1 + 1 = 2
2 < 5 → True
III
Love Babbar
i = 2 + 1 = 3
3 < 5 → True
Love Babbar
IV
i = 3 + 1 = 4
4 < 5 → True
Love Babbar
V
i = 4 + 1 = 5
5 < 5 → false
for (i = 0 ; i < 5 , i = i + 1)
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5 → false → bahar
IV



i = 0

0 < 3 → T

print 0

i = i+1 = 0+1 = 1

1 < 3 → T

print 1

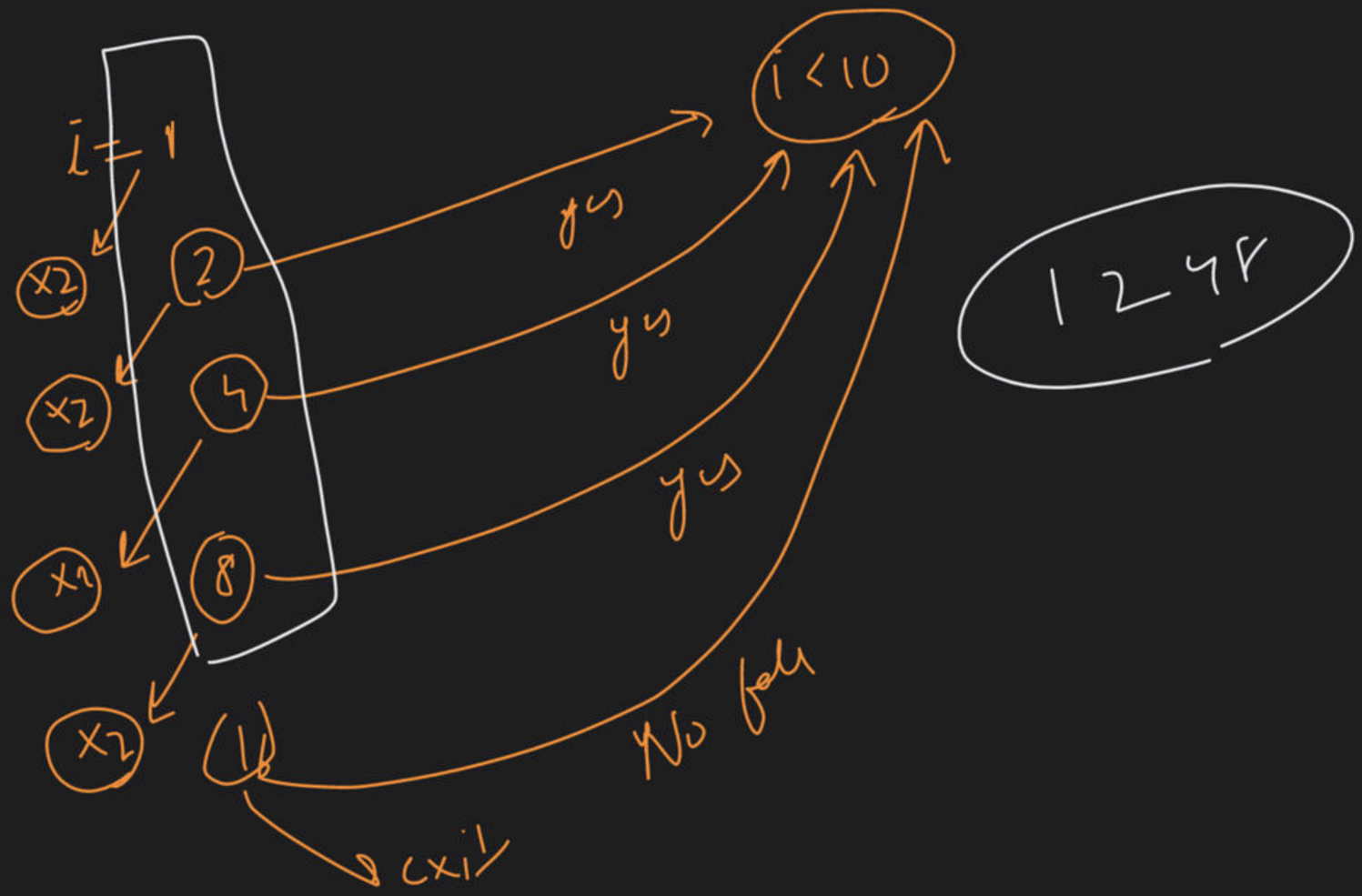
i = i+1 = 1+1 = 2

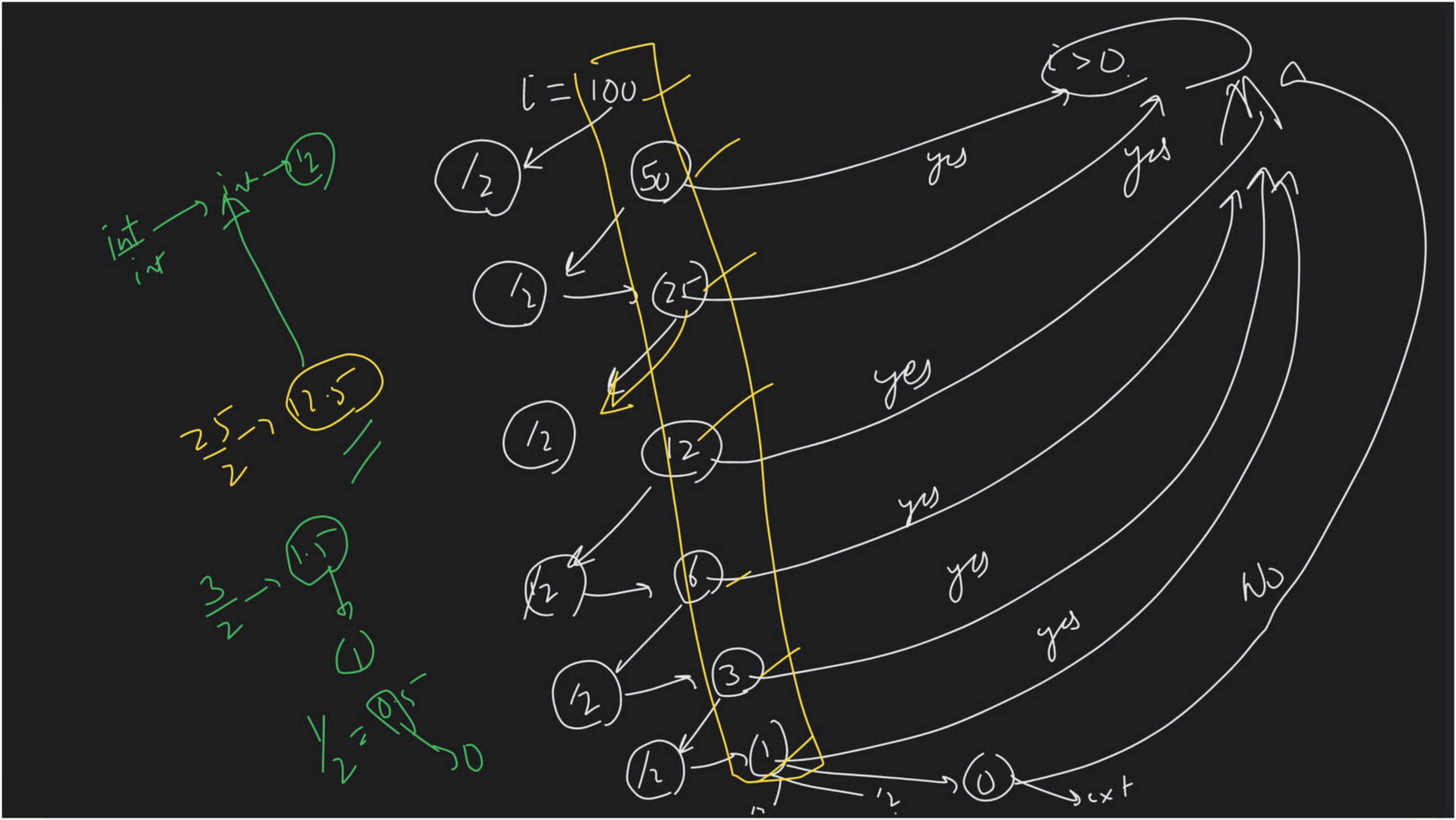
2 < 3 → T

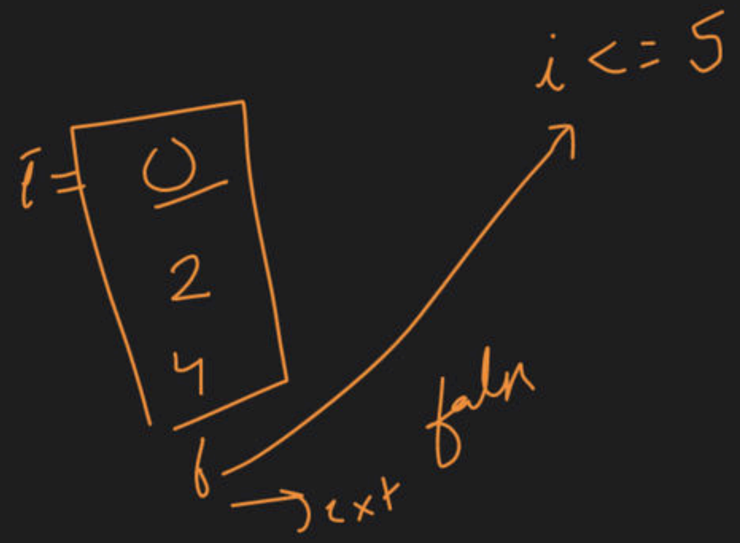
print 2

i = i+1 = 2+1 = 3

3 < 3 → false

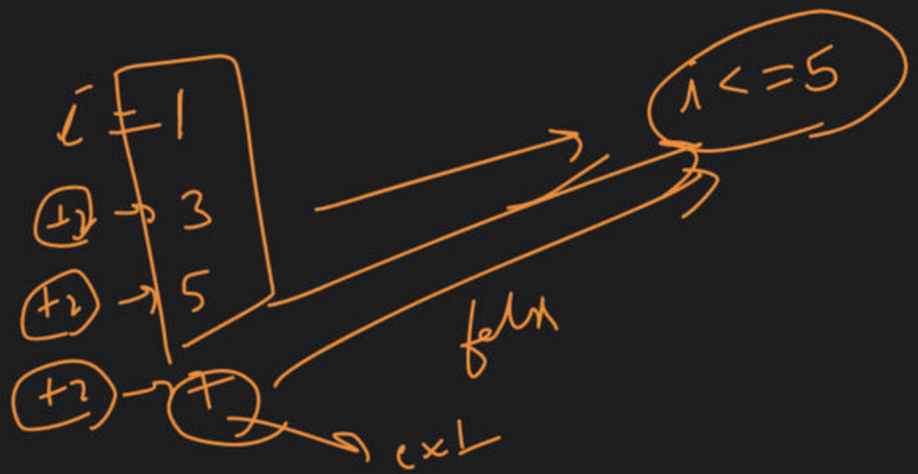

i = 1
x2
2
x2
4
x1
8
x2
1
i<10
yes
yes
yes
No fall
exit
1 2 4 8

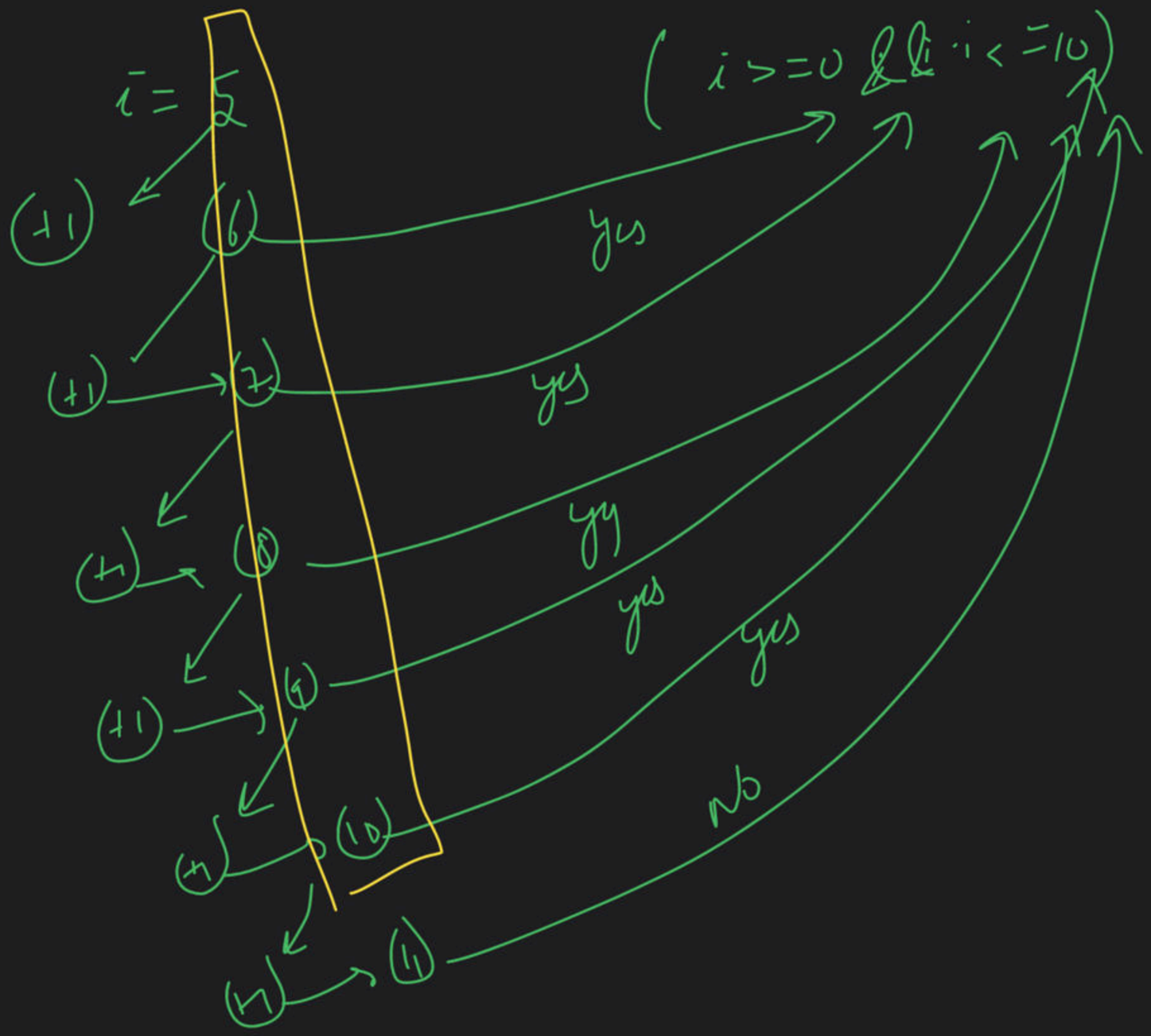

i = 100
i > 0
yes
yes
yes
yes
yes
yes
NO
int
int
12
12.5
2.5
2
1.5
3
2
1
0.5
0
cnt



i = 1 → i<=10

i -> 1 ×2 → 2
4
2 ×2
3 ×2 6
4 ×2 8
5 ×2 10
6 ×2 12
7 ×2 14
8 ×2 16
9 ×2 18
10 ×2 20

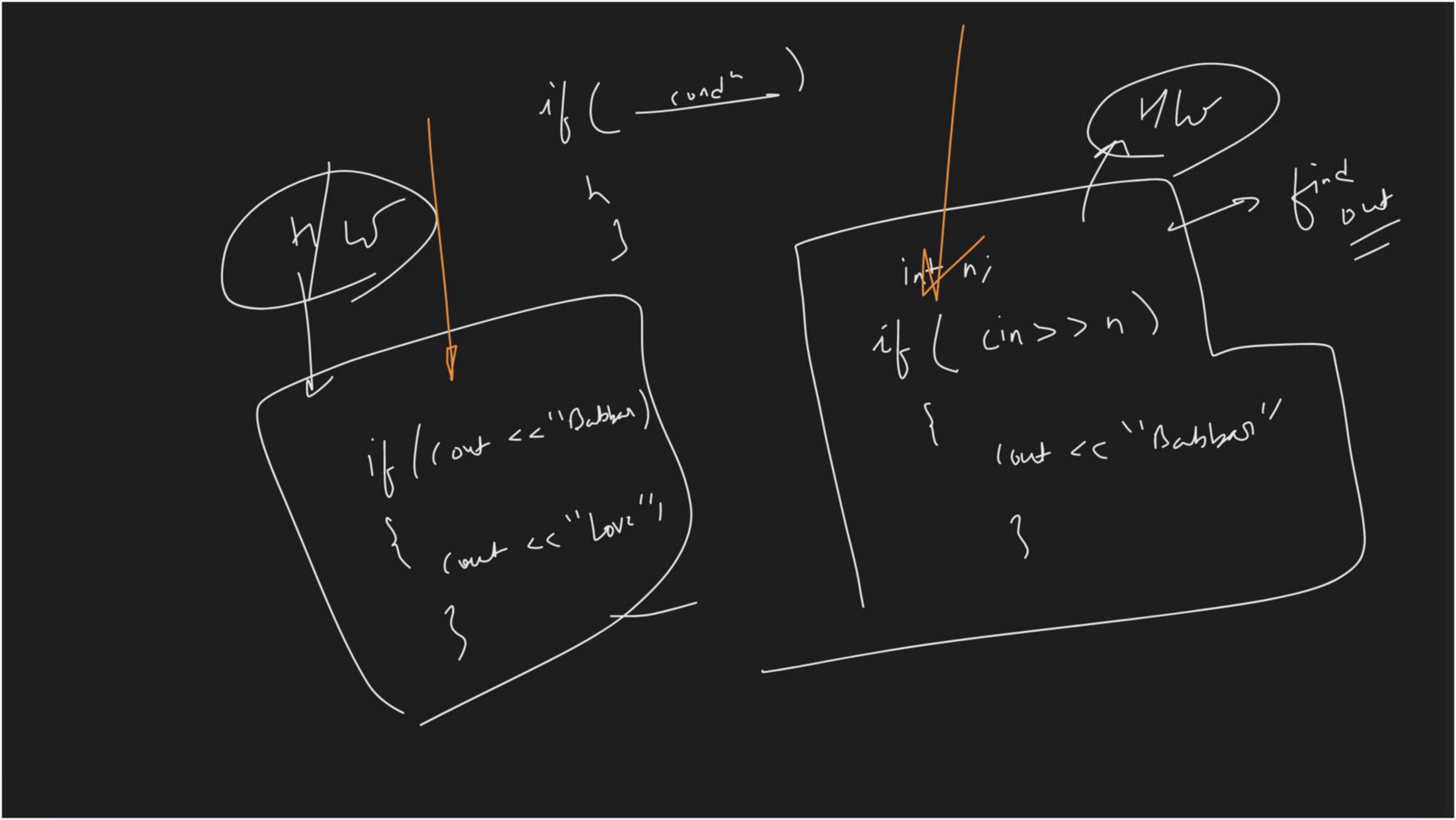

i = 1
+1
3
+2
5
+2
1<=5
false
exit

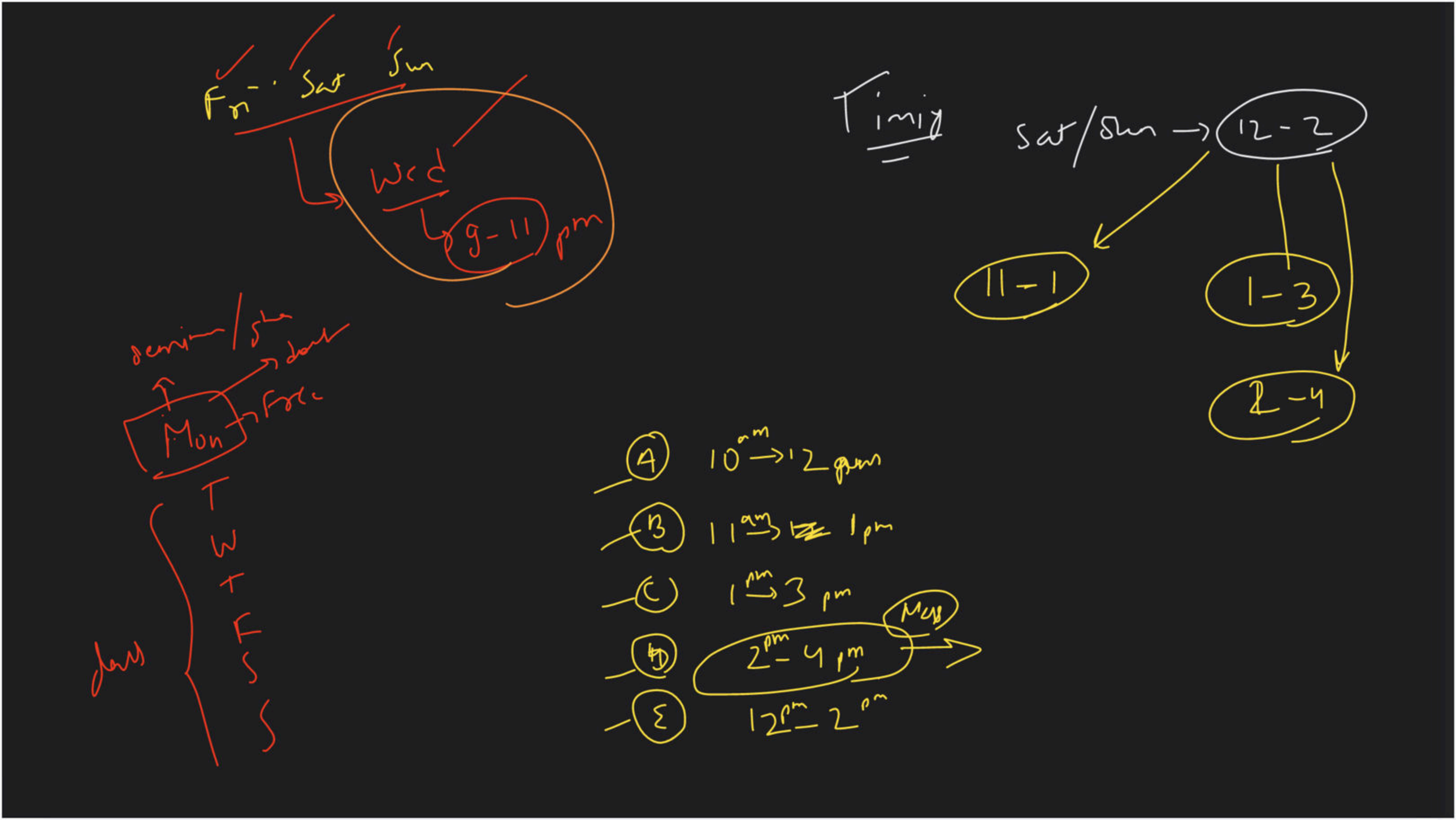

i = 5
(i >= 0 && i <= 10)
+1
6
yes
+1
7
yes
+1
8
yes
+1
9
yes
yes
+1
10
yes
+1
11
no
+1
11

Mandatory / Optional

```
for ( init ; cond ; updation )
{

}
```


if (cond^n)
{
}
My Lw
if ((out << "Babbar")
{ (out << "love";
}
My Lw
find out
int n;
if (cin >> n)
{
(out << "Babbar";
}


Fri Sat Sun
Wed
9-11 pm
Timing
Sat/Sun → 12-2
11-1
1-3
2-4
session/class
Mon
doubt
Force
class
T
W
T
F
S
S
A 10 am → 12 pm
B 11 am → 1 pm
C 1 pm → 3 pm
D 2 pm - 4 pm
E 12 pm - 2 pm
Mon

```
int bronum;
cin >> bronum;
if ( bronum == 0)
  {
    cout << " Baat Banyagi";
  }
else
  {
    cout << " Baat Nahi";
  }
```

30 min
↳ if-else

bronum

IU Scc

bronum -> variable create

take input

bronum == 0

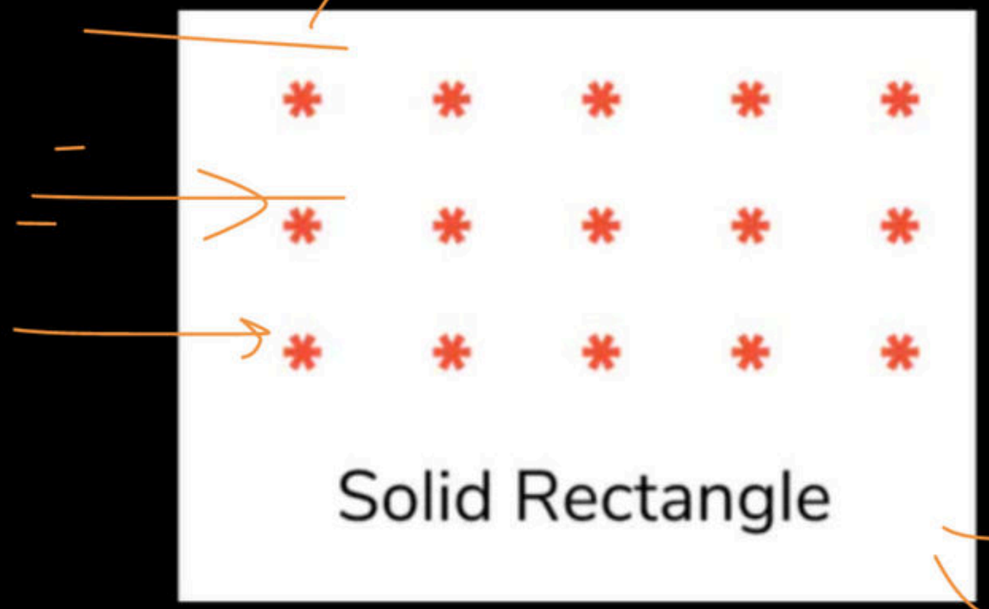
T

print "Baat banyagi"

agar nahi h

cout << "Baat nhi banyi"

Solid Rectangle



2 min Break

Why ?

5 month

Logic Building

Logic Building

Loops → strong

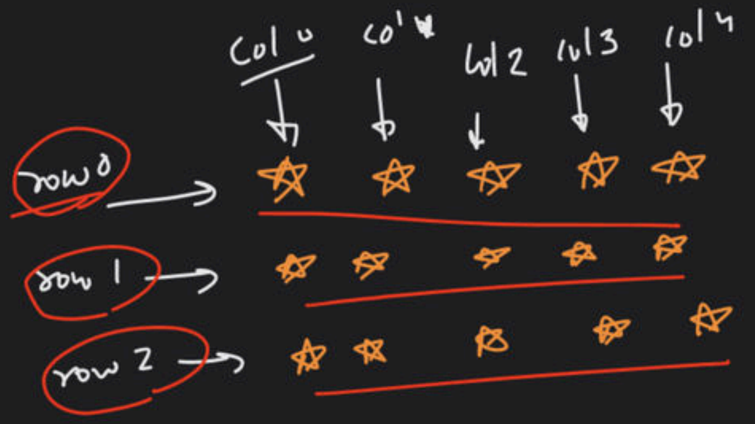
① Row - observation

total rows → 3

② Col - observation

col 0 → 3 ★
col 1 → 3 ★
col 2 → 3 ★
col 3 → 3 ★
col 4 → 3 ★

har column
me 3 ★ print karna hai



Solid Rectangle

Outer → row → row count

Outer

```
for ( int row = 0 ; row < 3 ; row = row + 1)
{
    for (int col=0; col < 5 ; col = col + 1)
    {
        cout << "*";
    }
    cout << endl;
}
```

Outer

for ( )

Inner

for ( )

{

}

}

row = 0
row < 3

0
1
2

col 0 col 1 col 2 col 3 col 4

row 0
row 1
row 2

row 0 → 5 *
row 1 → 5 *
row 2 → 5 *

Code :-

nested loop

2 loops

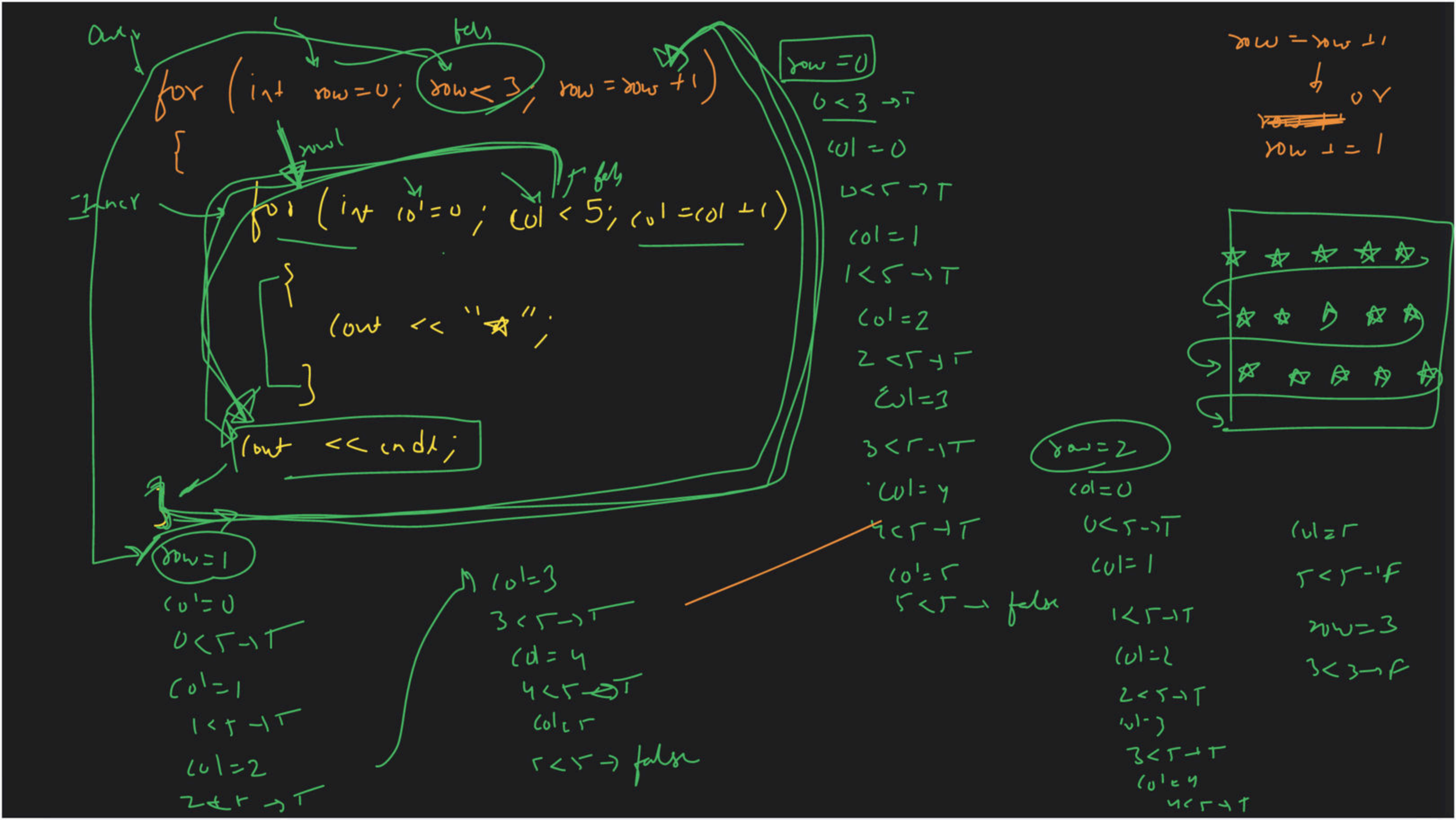
multiple loops

row 1

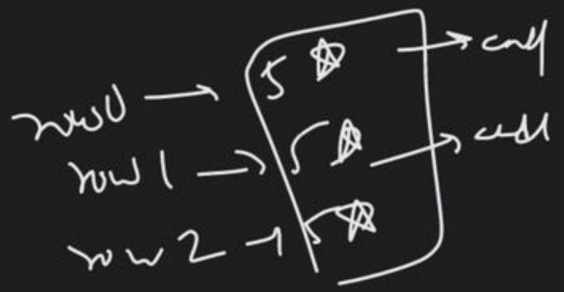
Outer

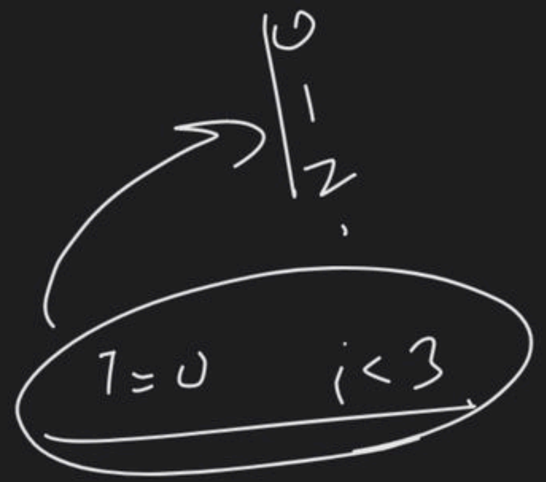
Inner

cols


Outer
for (int row=0; row<3; row=row+1)
{
row=0
0<3 → T
col=0
0<5 → T
col=1
1<5 → T
col=2
2<5 → T
col=3
3<5 → T
col=4
4<5 → T
col=5
5<5 → false
Inner
for (int col=0; col<5; col=col+1)
{
cout << "*";
}
cout << endl;
row=1
col=0
0<5 → T
col=1
1<5 → T
col=2
2<5 → T
col=3
3<5 → T
col=4
4<5 → T
col=5
5<5 → false
row=2
col=0
0<5 → T
col=1
1<5 → T
col=2
2<5 → T
col=3
3<5 → T
col=4
4<5 → T
col=5
5<5 → false
row=3
3<3 → false
row=row+1
0+1=1
1+1=2
2+1=3
col=5
5<5 → F
row=3
3<3 → F

```
for ( int row = 0 ; row < 3 ; row = row + 1 )
{
    for ( int col = 0 ; col < 5 ; col = col + 1 )
    {
        cout << "*" ;
    }
    cout << endl;
}
```



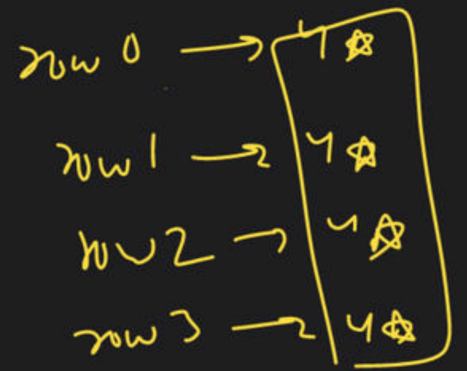


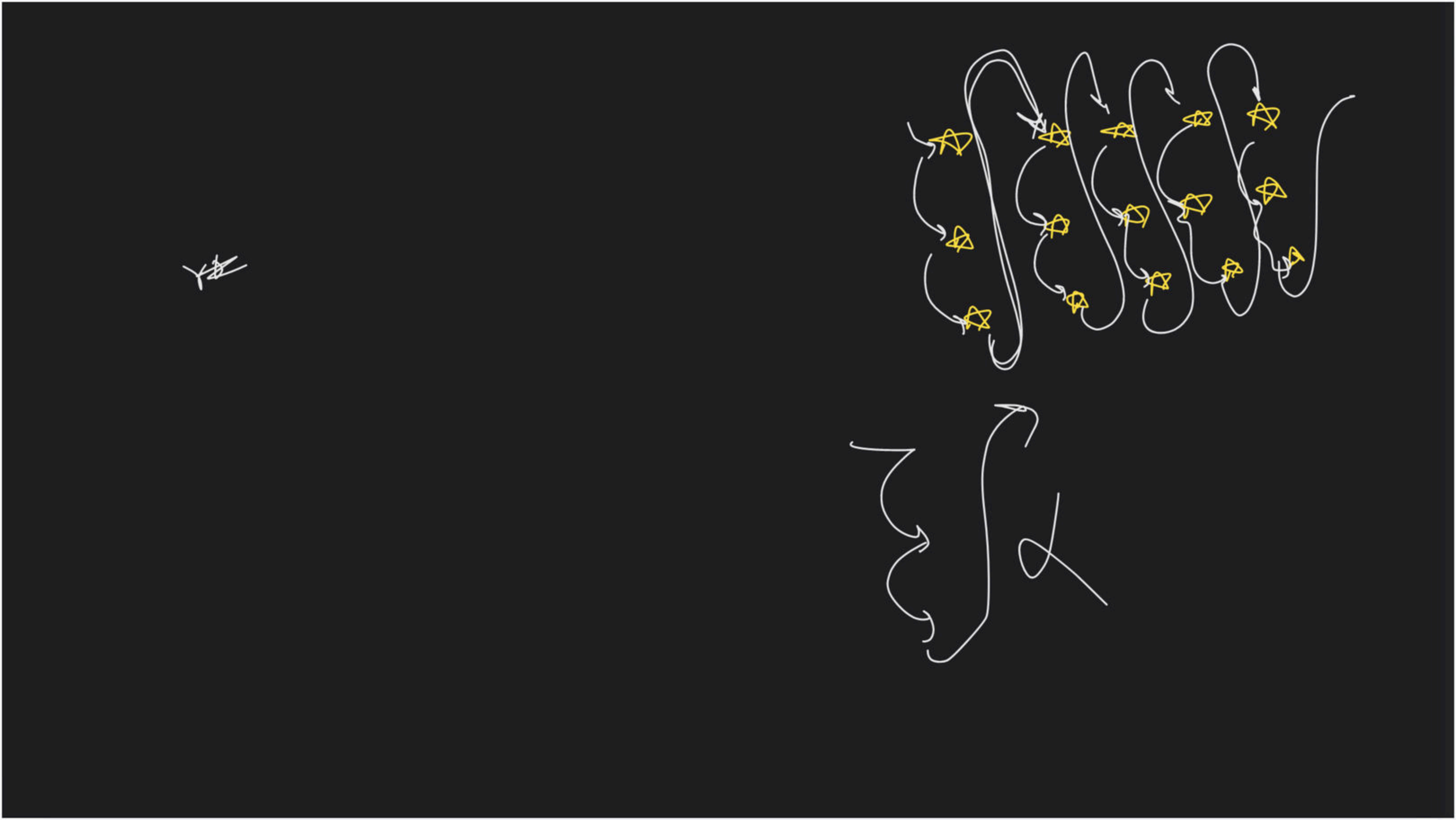
①

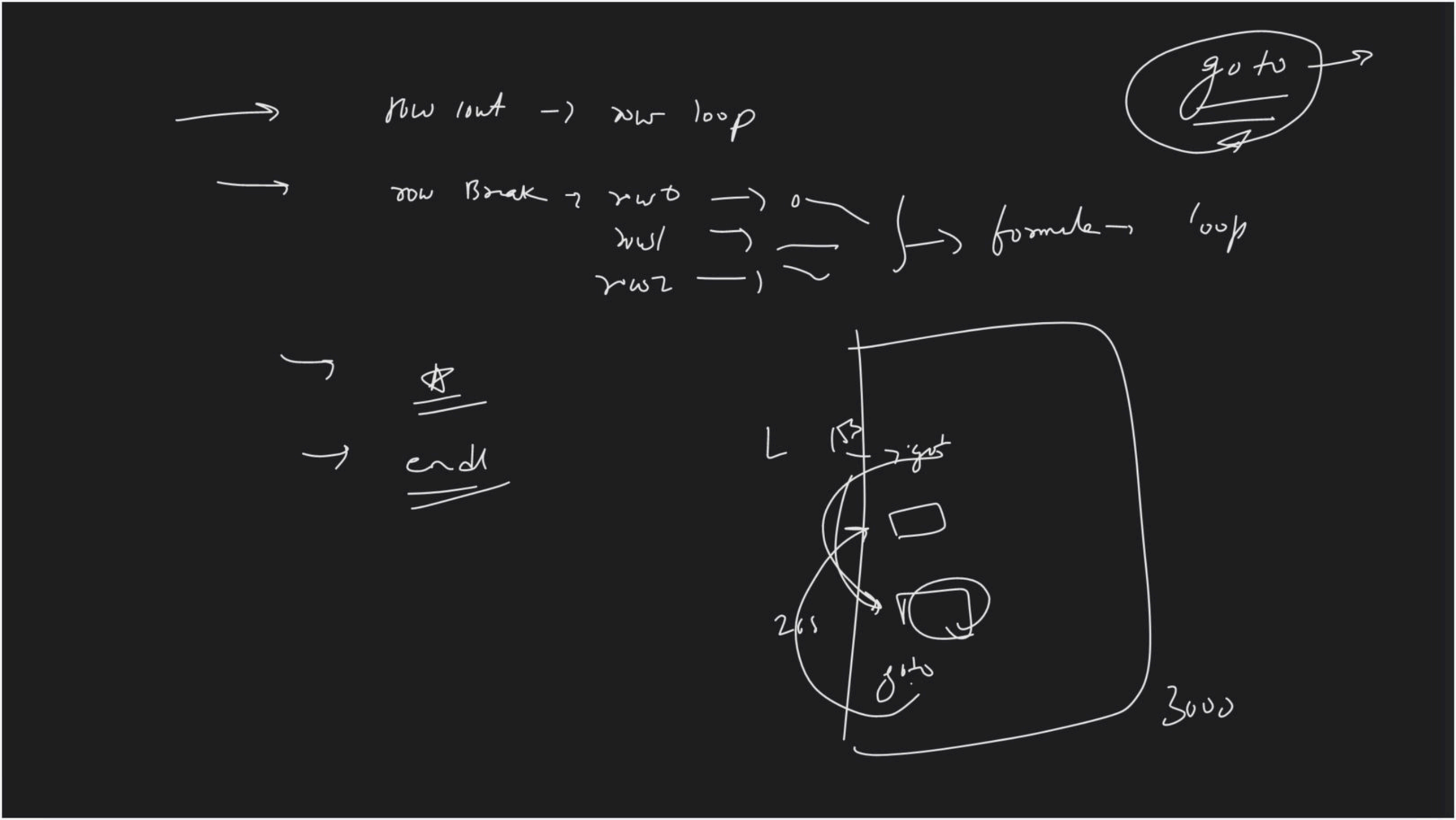
```
for(int row=0; row<4; row=row+1)
{
    for(int col=0; col<4; col=col+1)
    {
        cout << "*";
    }
    cout << endl;
}
```

0 → 4

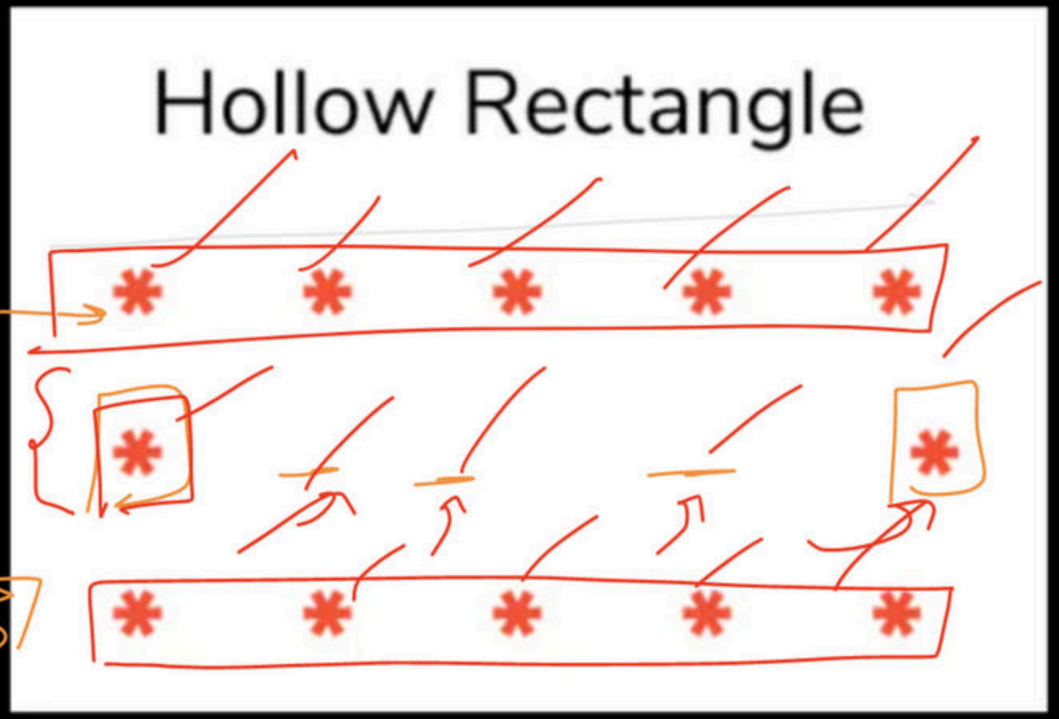
Square Pattern ⇒ 1 → 5






goto
endl
3000

Hollow Rectangle

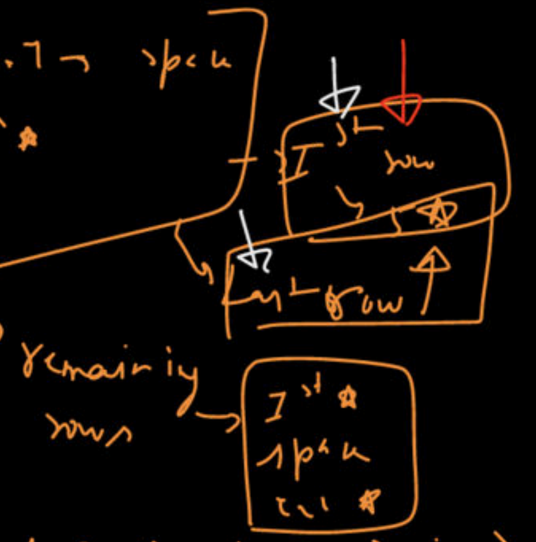



```
for (int i=0; i<3; i=i+1)
{
    for(int j=0; j<5; j=j+1)
    {
        if (first Row || last row)
            print *
    }
}
```

row 0 → 5 *

row 1 → 1st * / 5th *

row 2 → 5 *



row 0 → 5 *

row 2 → 5 *

remaining middle rows → 1st * → space → end *

row 0 → 5 ★

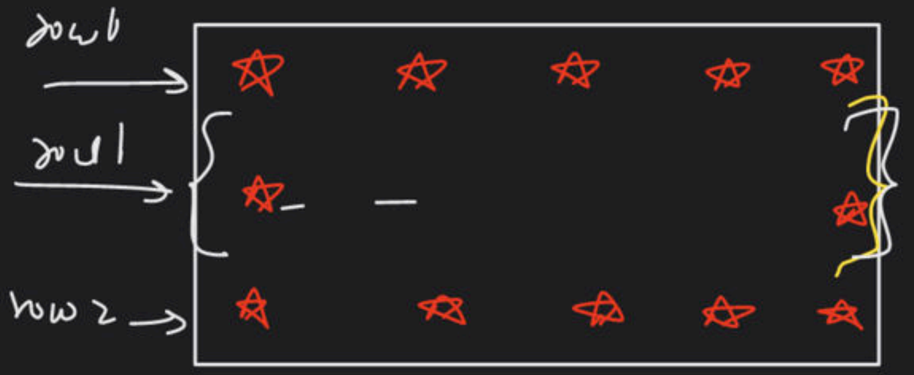
row 1 → 1 ★ space 1 ★ → 5

row 2 → 5 ★

total 3 rows

Outer loop

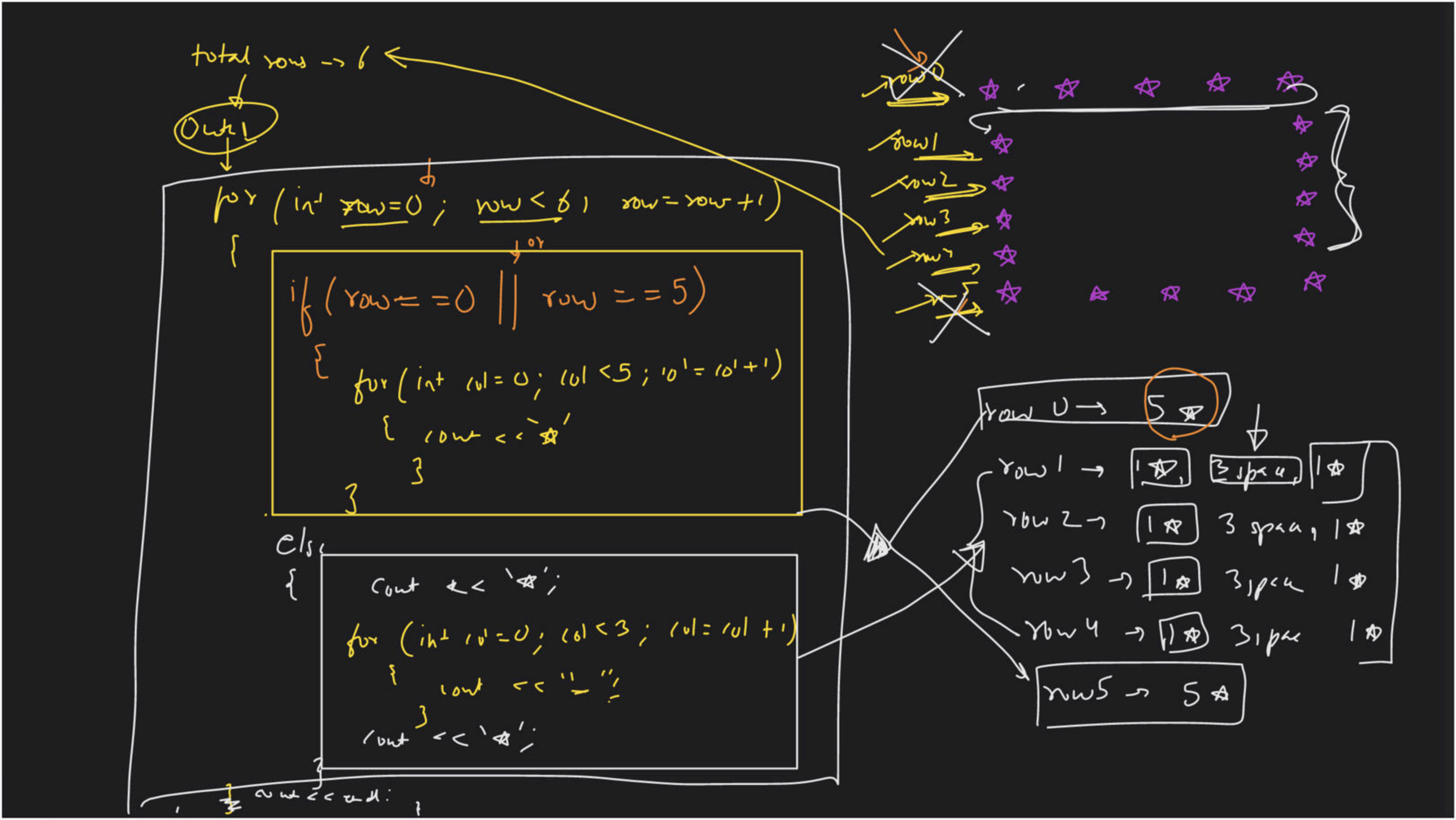
```
for (row=0; row<3; row=row+1)
{
    if (row==0 || row==2)
    {
        for(int i=0; i<5; i=i+1)
        { cout << *;
        }
    }
    else {
        cout << *
        for(int i=0; i<3; i=i+1)
        { cout << " ";
        }
        cout << *;
    }
}
```

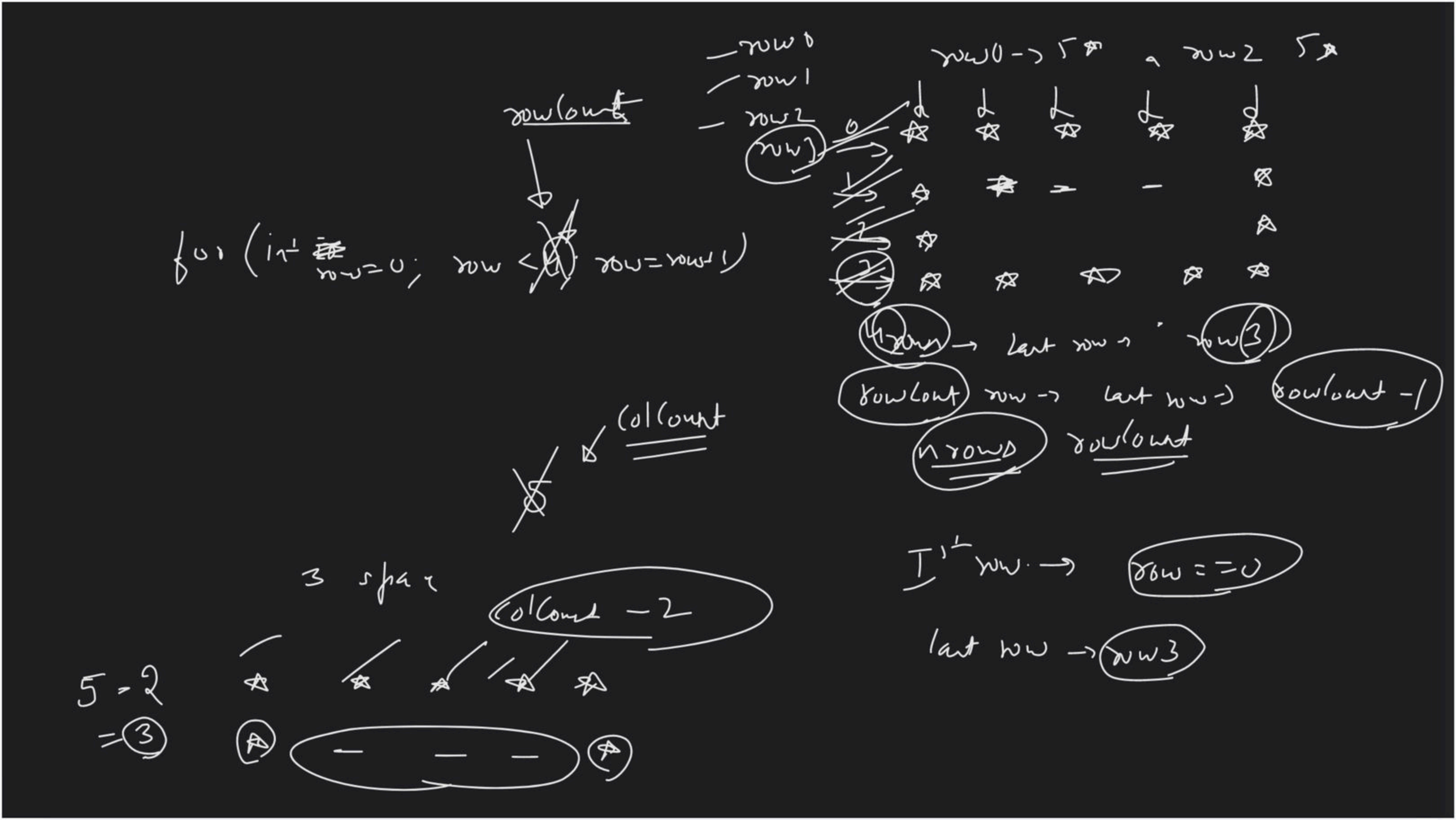


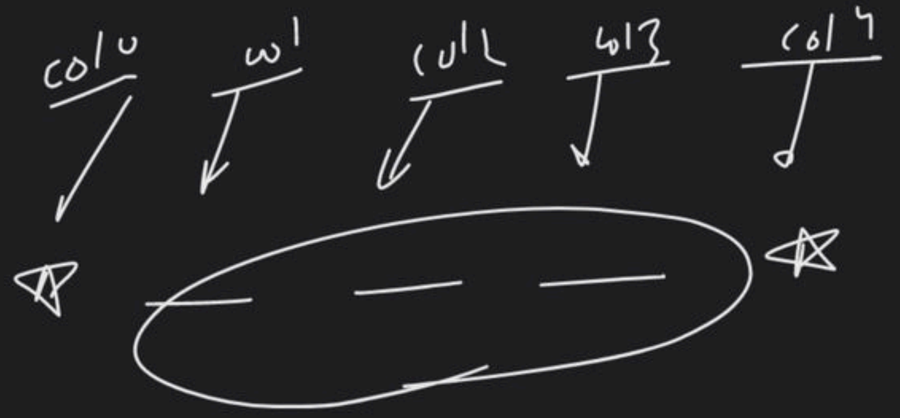
row 0 → 5 ★

row 1 → 1 ★, 3 spaces, 1 ★

row 2 → 5 ★


total rows -> 6
Output
for (int row=0; row<6; row=row+1)
{
if (row==0 || row==5)
{
for (int col=0; col<5; col=col+1)
{ cout << '*'
}
}
else
{
cout << '*';
for (int col=0; col<3; col=col+1)
{ cout << ' ';
}
cout << '*';
}
cout << endl;
}
row 0
row 1
row 2
row 3
row 4
row 5
row 0 -> 5 *
row 1 -> 1 * 3 spaces 1 *
row 2 -> 1 * 3 spaces 1 *
row 3 -> 1 * 3 spaces 1 *
row 4 -> 1 * 3 spaces 1 *
row 5 -> 5 *

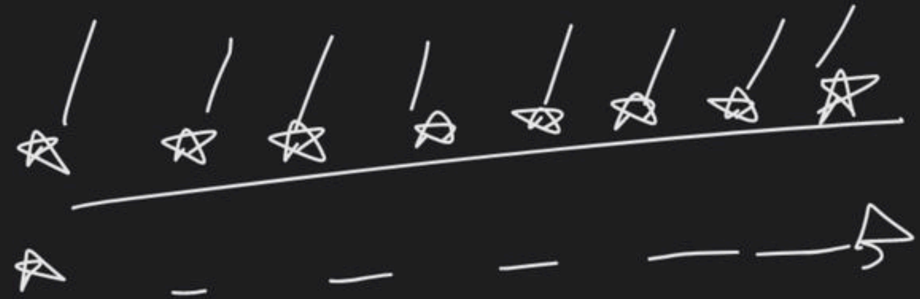

rowCount
for (int row=0; row < rowCount; row=row+1)
row0
row1
row2
row3
row0 -> 5 ⭐
row2 5 ⭐
1 row → last row → row(3)
rowCount row -> last row -> rowCount - 1
n rows rowCount
colCount
3 space
colCount - 2
5 - 2
= 3
I st row → row == 0
last row → row3



5 col → 3 space

8 − 2 → 6 space

4 − 2 → 2 space



no of col − 2 → spaces

colCount − 2

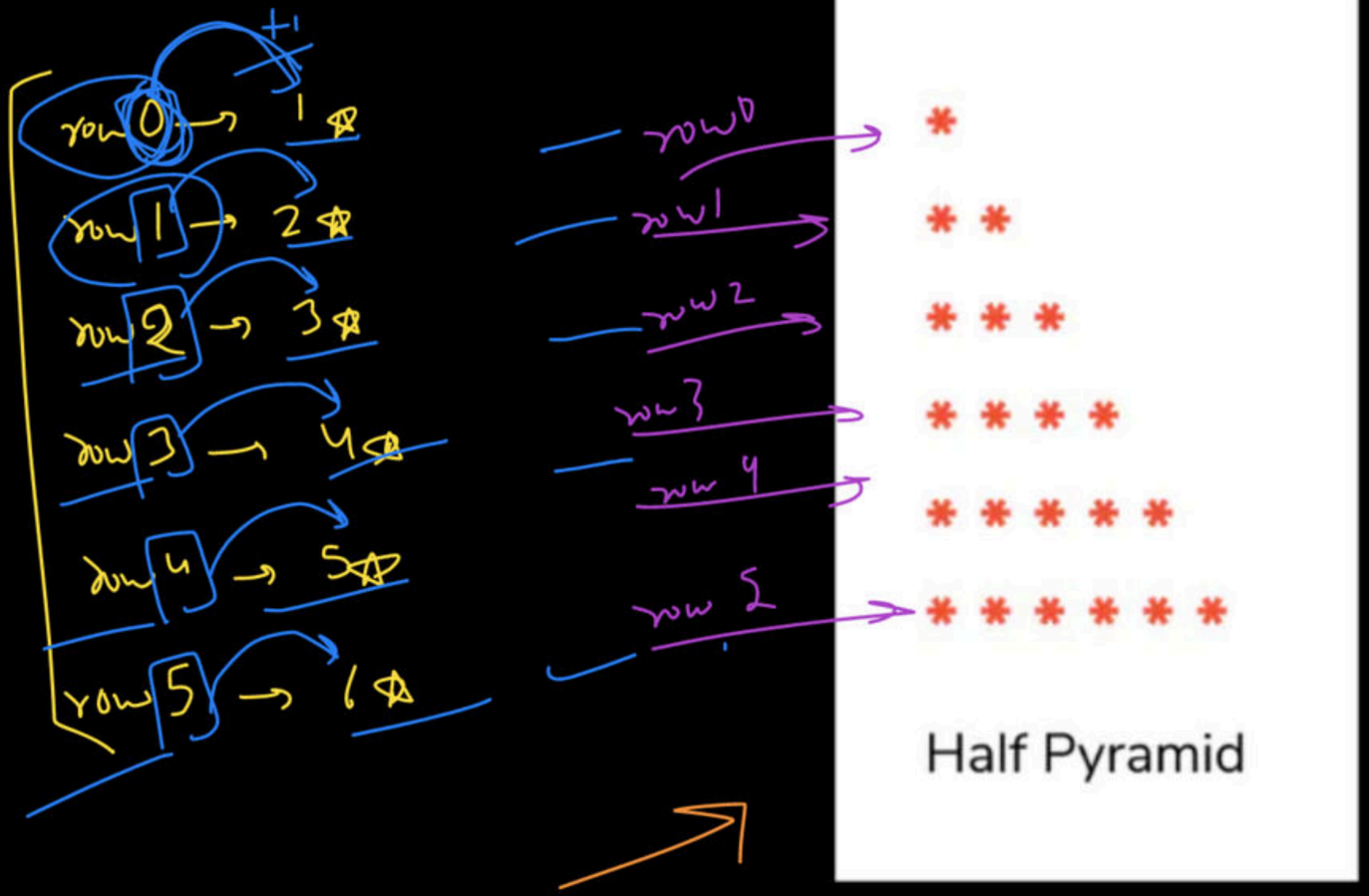
3 − 2 : 1

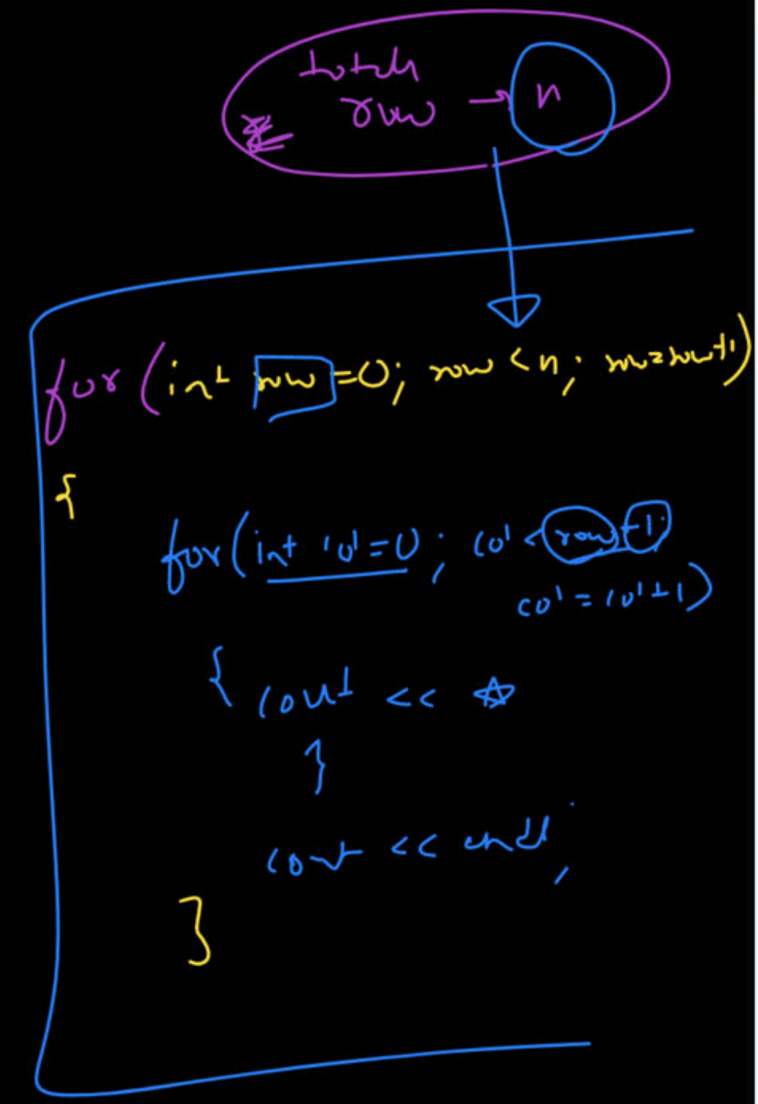
```
for (int i=0; i<3; i=(i+1))
{
    cout << "_ ";
}
```

5 ☆

```
for (int i=0; i<5; j=i+1)
{
    cout << "☆";
}
```

Half Pyramid



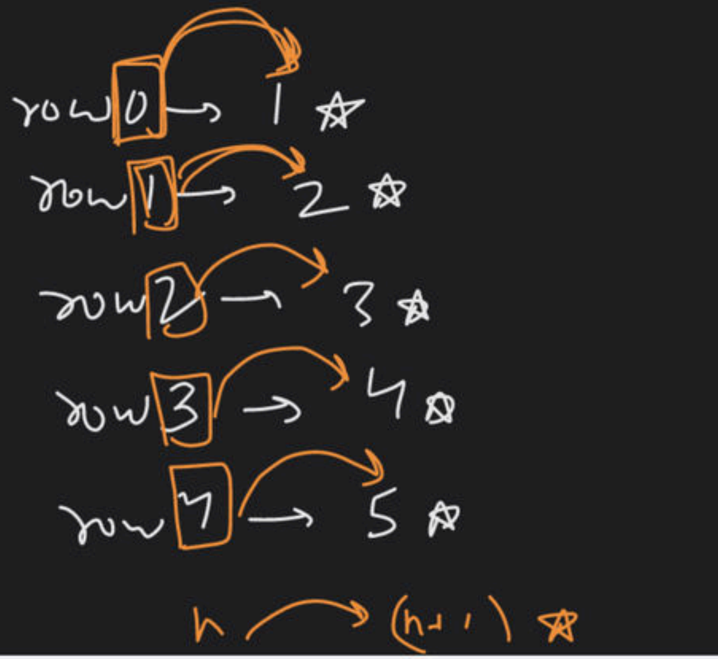


total rows → 5

```
for ( int row=0; row < 5; row=row+1)
{
    for (int col=0; col < row+1; col=col+1)
    {
        cout << '*';
    }
    cout << endl;
}
```

n rows

not available
↓
Reload
or
Rejoin



Inverted Half Pyramid

```
for ( int row = 0 ; row < n ; row = row + 1 )
  {
```

n - row

row 0 → 6 ★ → 6 - 0

row 1 → 5 ★ → 6 - 1

row 2 → 4 ★ → 6 - 2

row 3 → 3 ★ → 6 - 3

row 4 → 2 ★ → 6 - 4

row 5 → 1 ★ → 6 - 5

n = 6

row 0

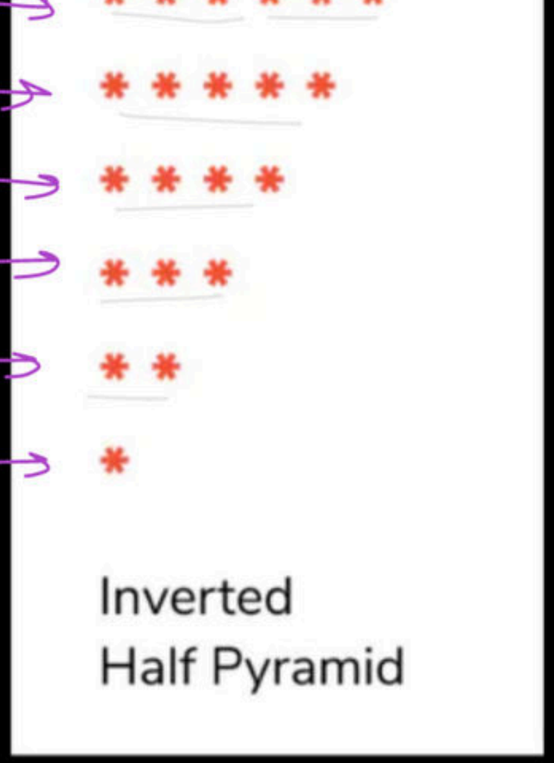
row 1

row 2

row 3

row 4

row 5




```
for (int col = 0; col < n-row ; col = col + 1)
  {
    cout << '*';
  }
cout << endl;
```

Steps:- (1) row count → row loop

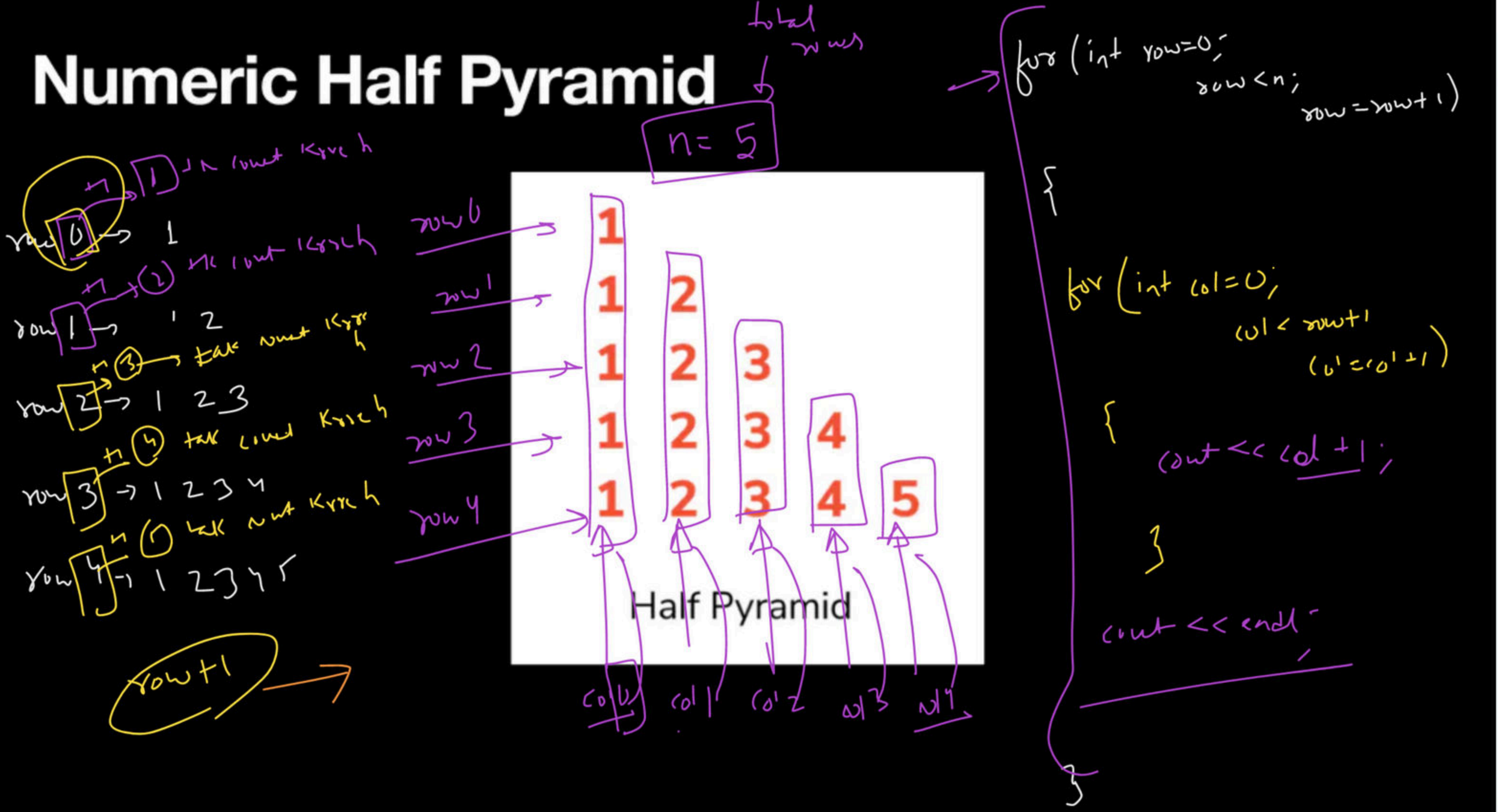
Step 2 → row logic break → formula → inner loop

★

_ _ _

```
}
```

Numeric Half Pyramid



row = 0

0 < 3 → T

↳ Inner loop starts now

col = 0

0 < row+1, 0 < 0+1 → T

print col+1 → print 1

col = 1

1 < 1 → F

↳ break and inner loop ends

row = 1

1 < 3 → T

↳ Inner loop start

col = 0

0 < row+1, 0 < 1+1 → T

print col+1 → 0+1 → 1

n = 3

col = 1

1 < 2 → T

print 1+1 = 2

col = 2

2 < 2 → F

↳ Inner ends here

row = 2

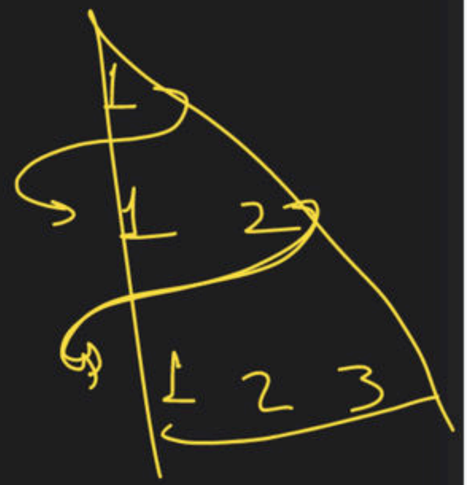
2 < 3 → T

↳ Inner loop

col = 0

0 < row+1 → 0 < 2+1 → T

print col+1 = 0+1 → 1



col = 1

1 < 3 → T

print col+1 = 1+1 = 2

col = 2

2 < 3 → T

print col+1 = 2+1 = 3

col = 3

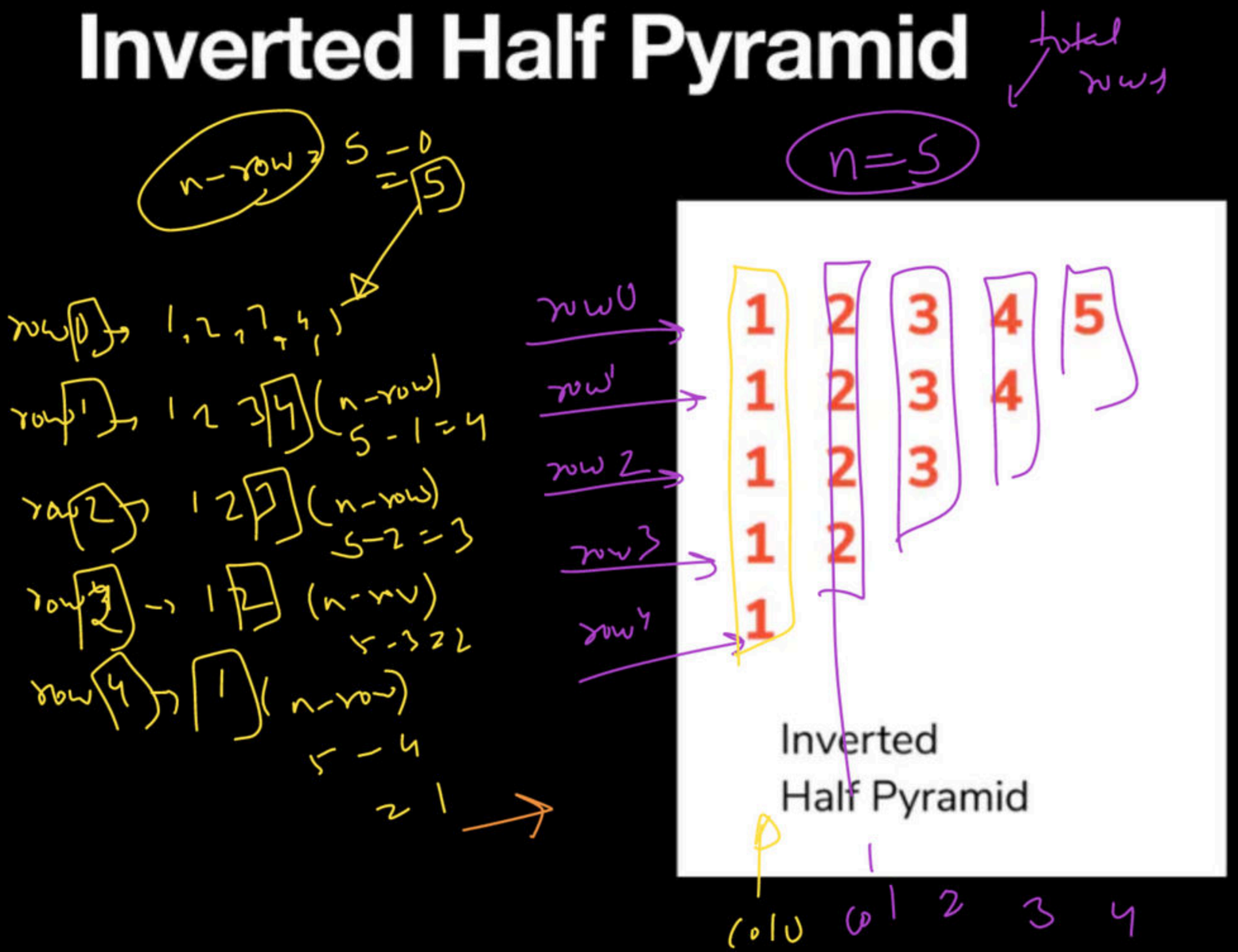
3 < 3 → F

↳ Inner ends here

↳ row++

3 < 3 → F / Outer loop ends

Inverted Half Pyramid




```
for (int row=0; row<n; row=row+1)
{
    for(int col=0; col<n-row; col=col+1)
    {
        print col+1
    }
}
```

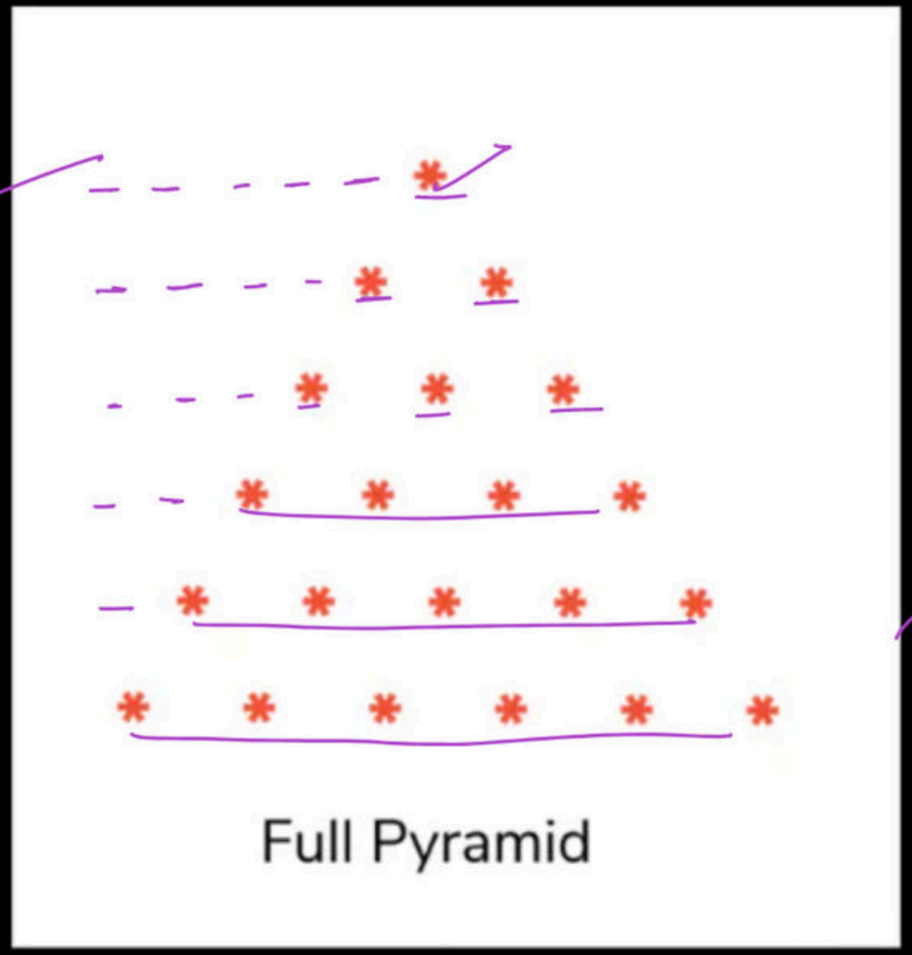
WCA

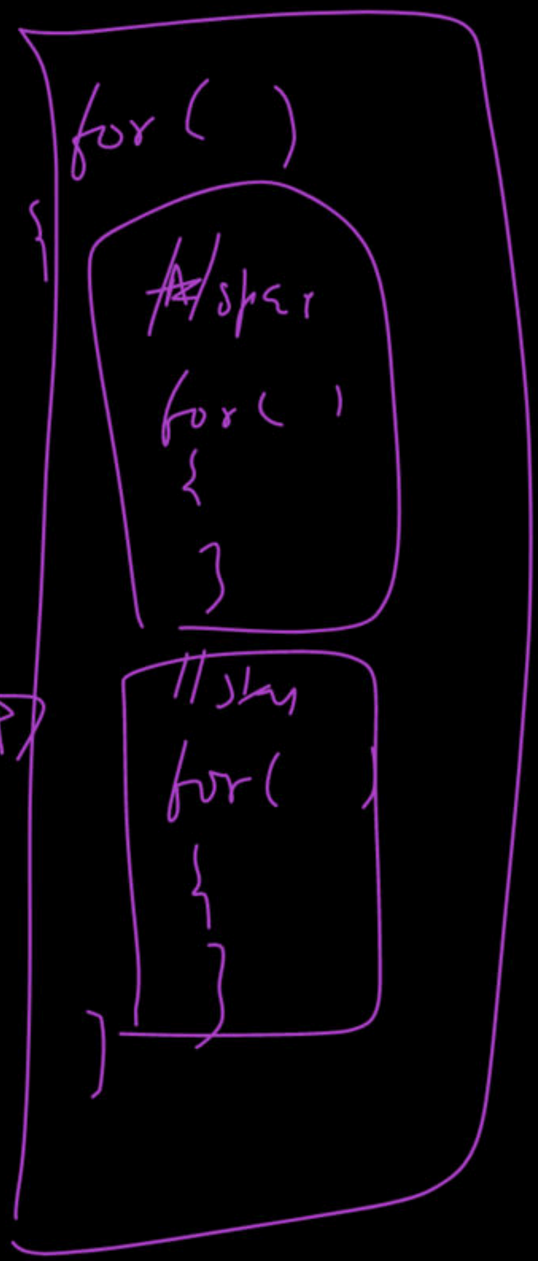
9-11 pm

Full Pyramid:

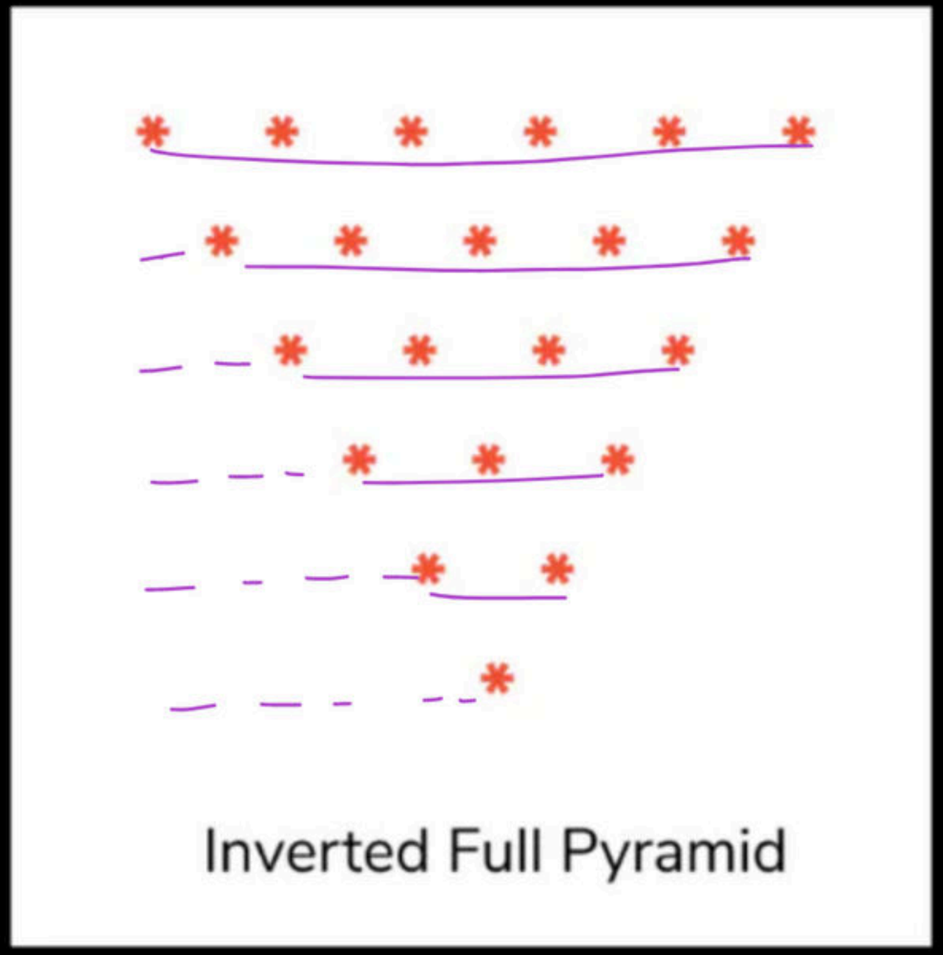
3 classes → padhai

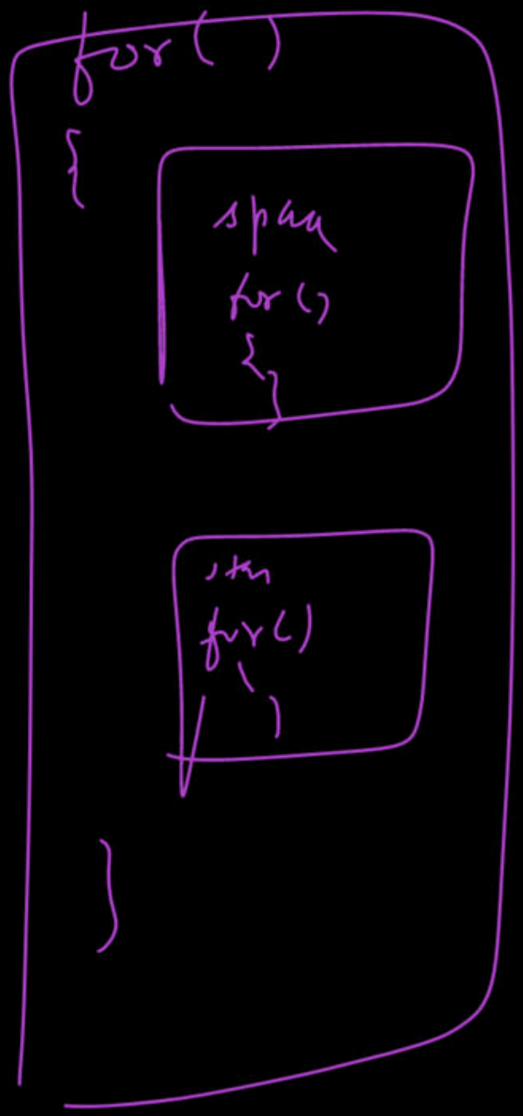
1 class → revision doubt discussion motivation gyaan feedback Practice topic



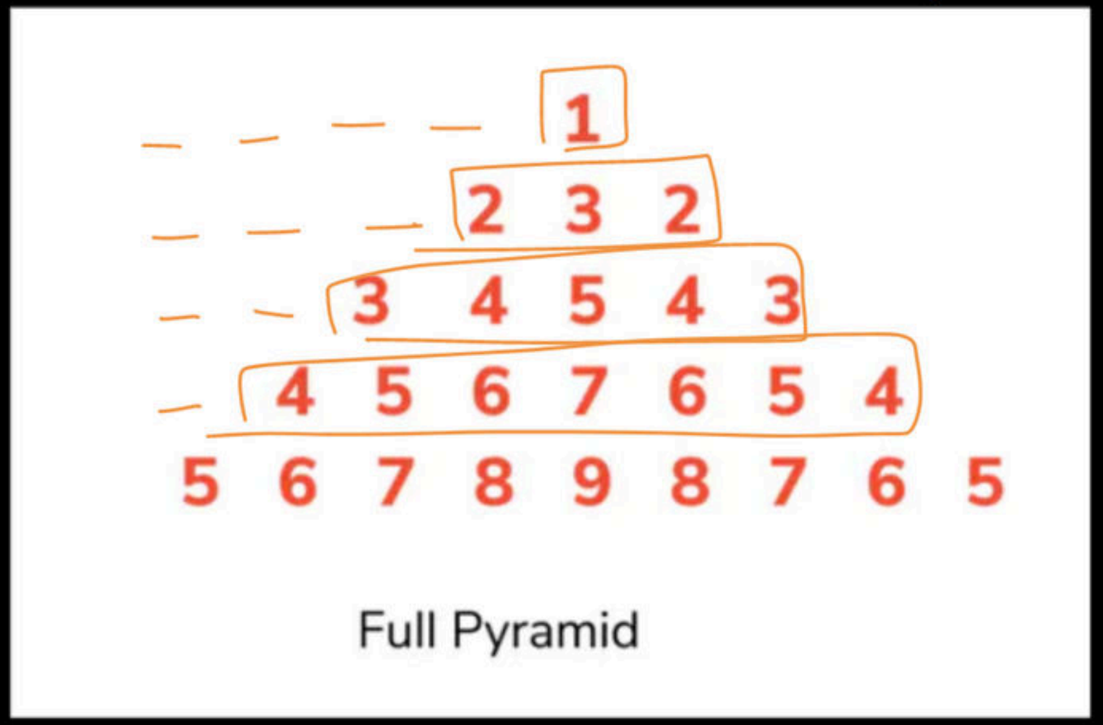


Inverted Full Pyramid:





Numeric Full Pyramid:



Full Pyramid

Numeric Hollow Full Pyramid

main