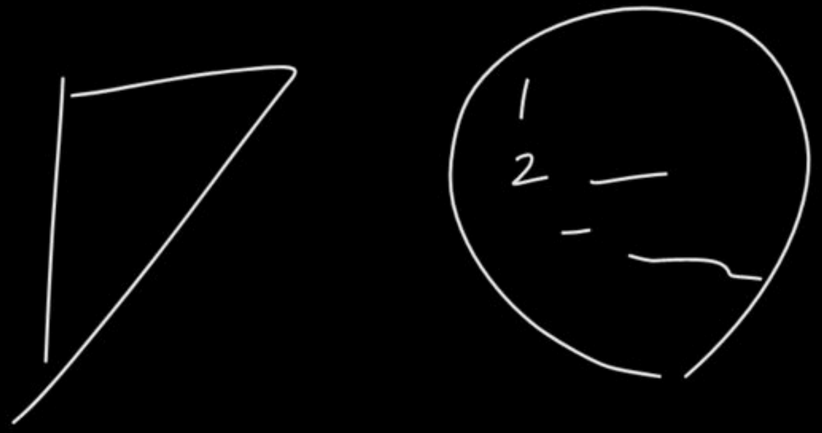
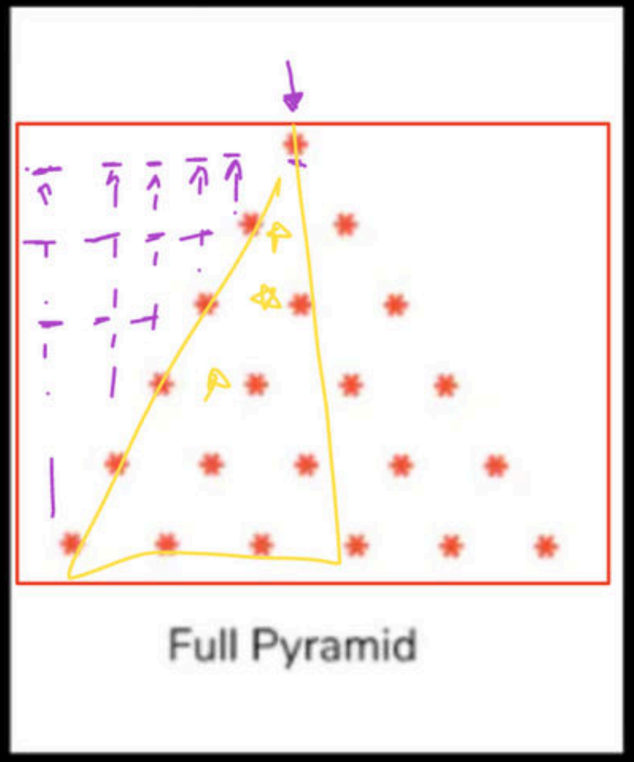
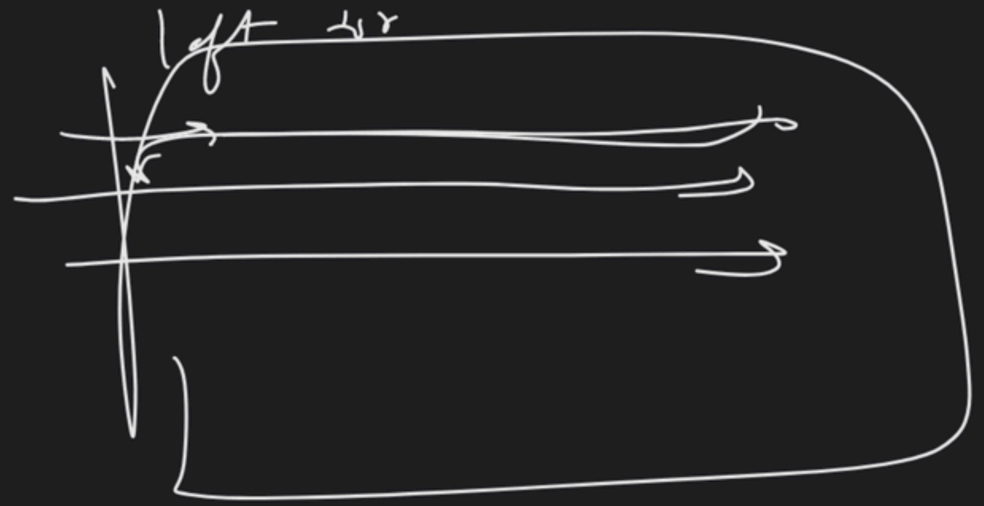


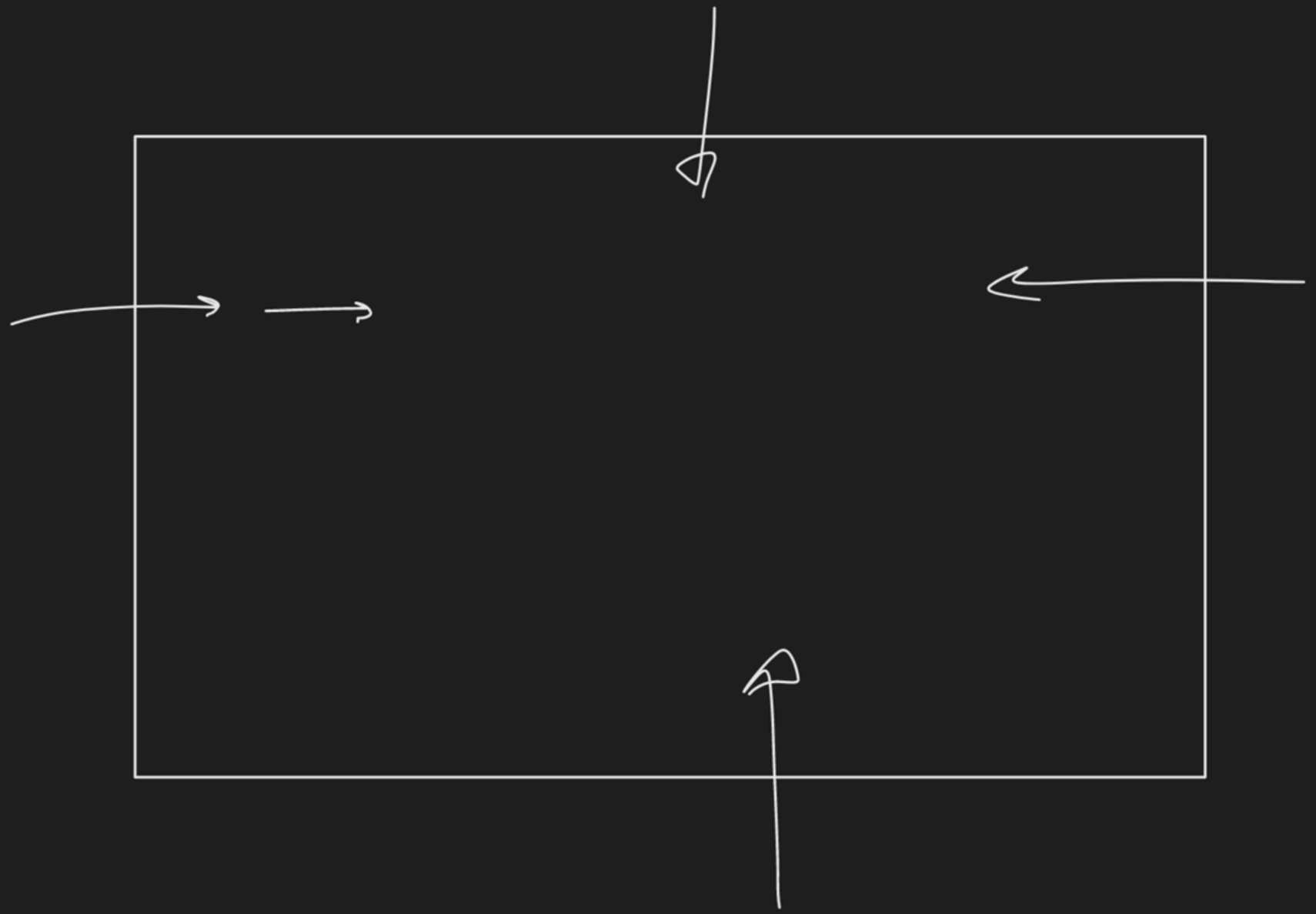
Some more Patterns

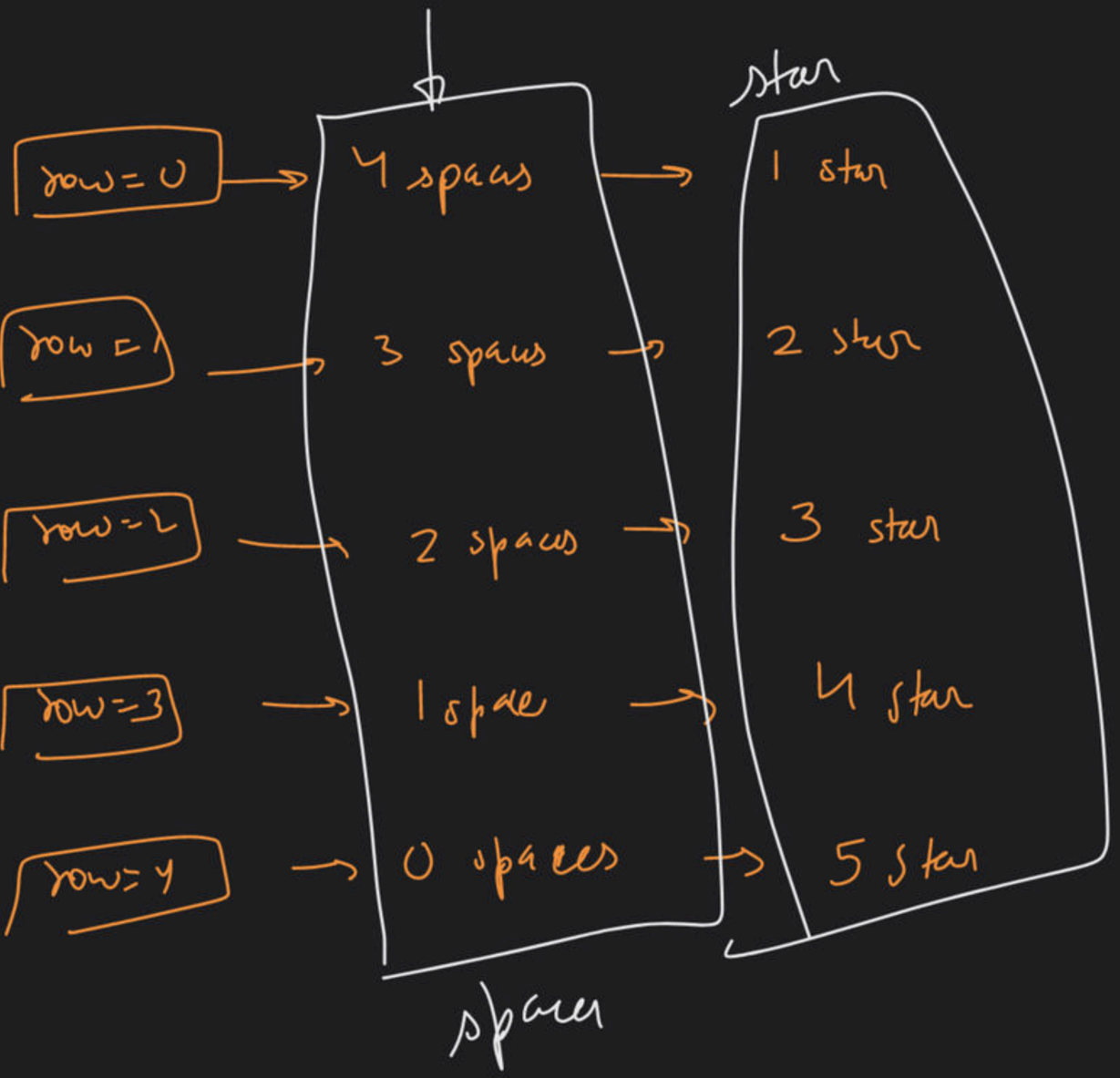
Instructor: Love Babbar

Full Pyramid

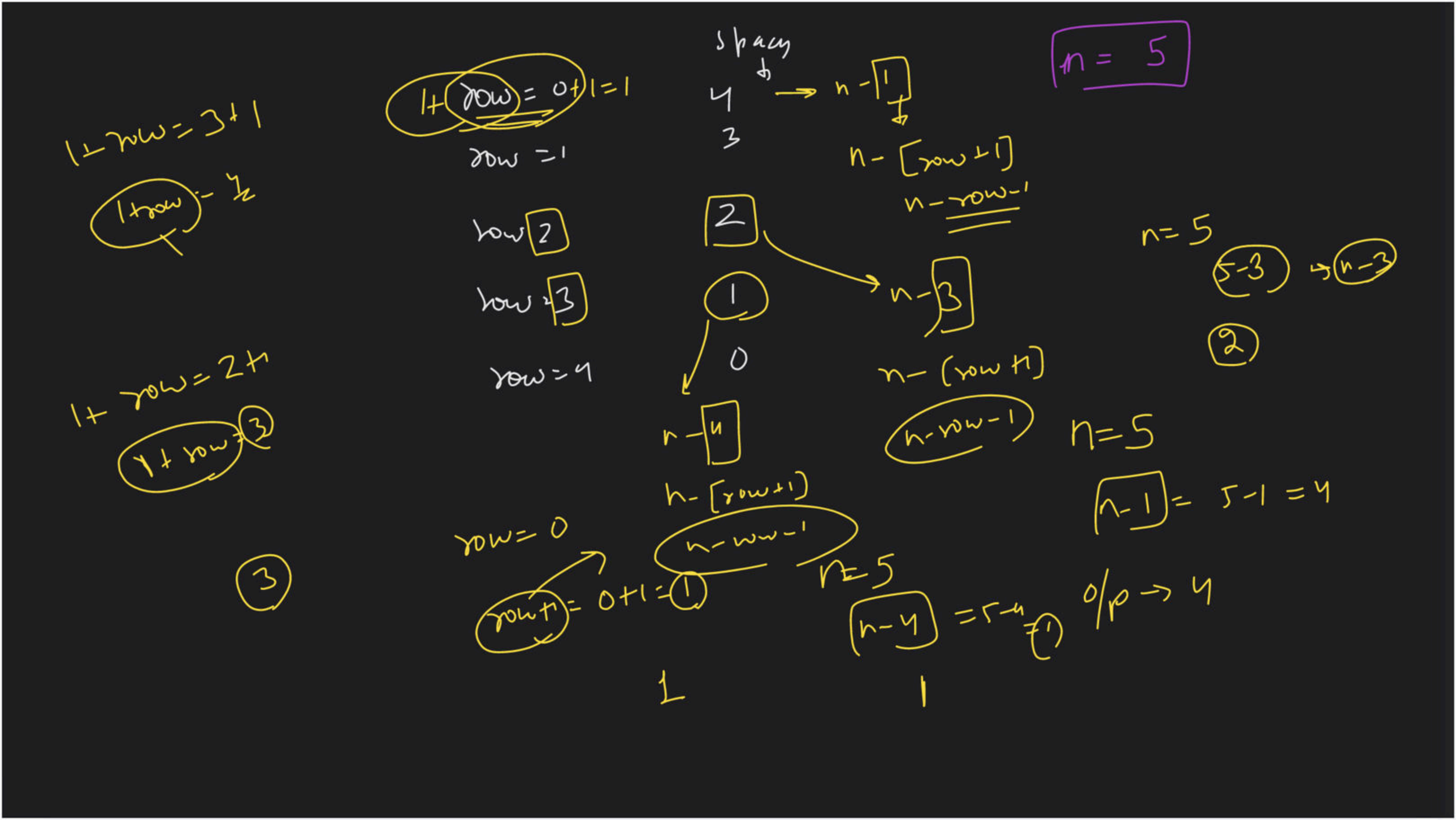


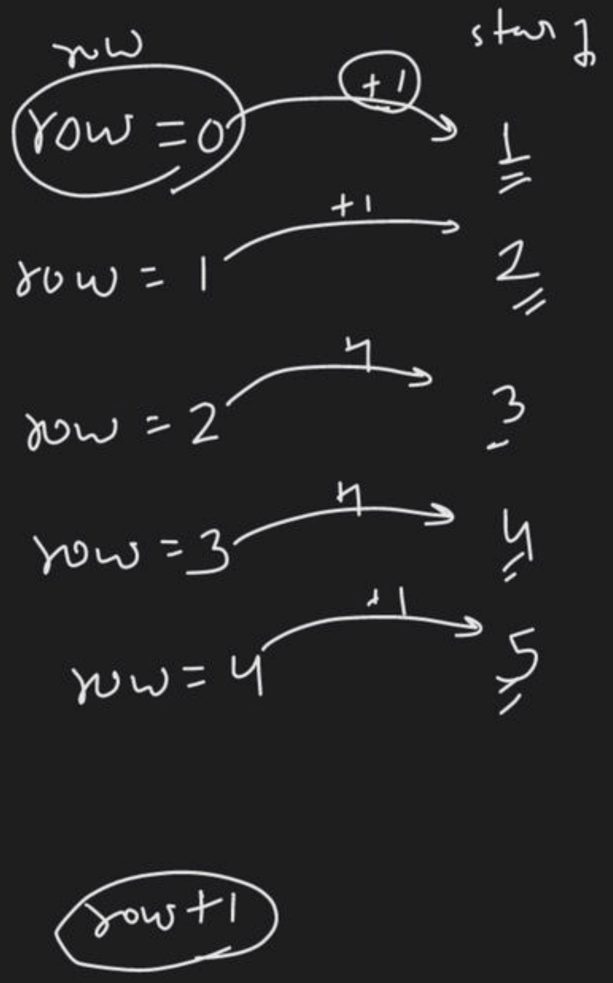





row = 0
4 spaces
1 star
row = 1
3 spaces
2 star
row = 2
2 spaces
3 star
row = 3
1 space
4 star
row = 4
0 spaces
5 star
star
spaces

r = 5


n = 5
spacy
4
3
n - 1
n - [row + 1]
n - row - 1
1 + row = 0 + 1 = 1
row = 1
1 + row = 3 + 1
1 + row = 4
row = 2
row = 3
row = 4
2
1
0
n - 3
n - (row + 1)
n - row - 1
n - 4
n - [row + 1]
n - row - 1
row = 0
row + 1 = 0 + 1 = 1
1 + row = 2 + 1
1 + row = 3
3
n = 5
5 - 3 → n - 3
2
n = 5
n - 1 = 5 - 1 = 4
n = 5
n - 4 = 5 - 4 = 1
o/p → 4
1
1


row
start
row = 0
+1
1
row = 1
+1
2
row = 2
+1
3
row = 3
+1
4
row = 4
+1
5
row+1

(3)

n = 4

n = 4 → total no of rows

```
outer loop
for (int row = 0; row < n; row = row + 1)
{
    // spaces
    for ( col = 0; col < n - row - 1 ; )
    {
    }
    // stars
    for (                          ; row + 1 )
    {
    }
}
```

inner loop

firstly → secondly

row 0 → 3 spaces → 1 star

row 1 → 2 spaces → 2 stars

row 2 → 1 space → 3 stars

row 3 → 0 spaces → 4 stars

row = 0

n - 1

n - [row + 1]

n - row - 1

n - 2

n - [row + 1]

n - row - 1

row 0 → 3 sp

row 1 → 2 sp

row = 1

row 2 → 1 sp

row 3 → 0 sp

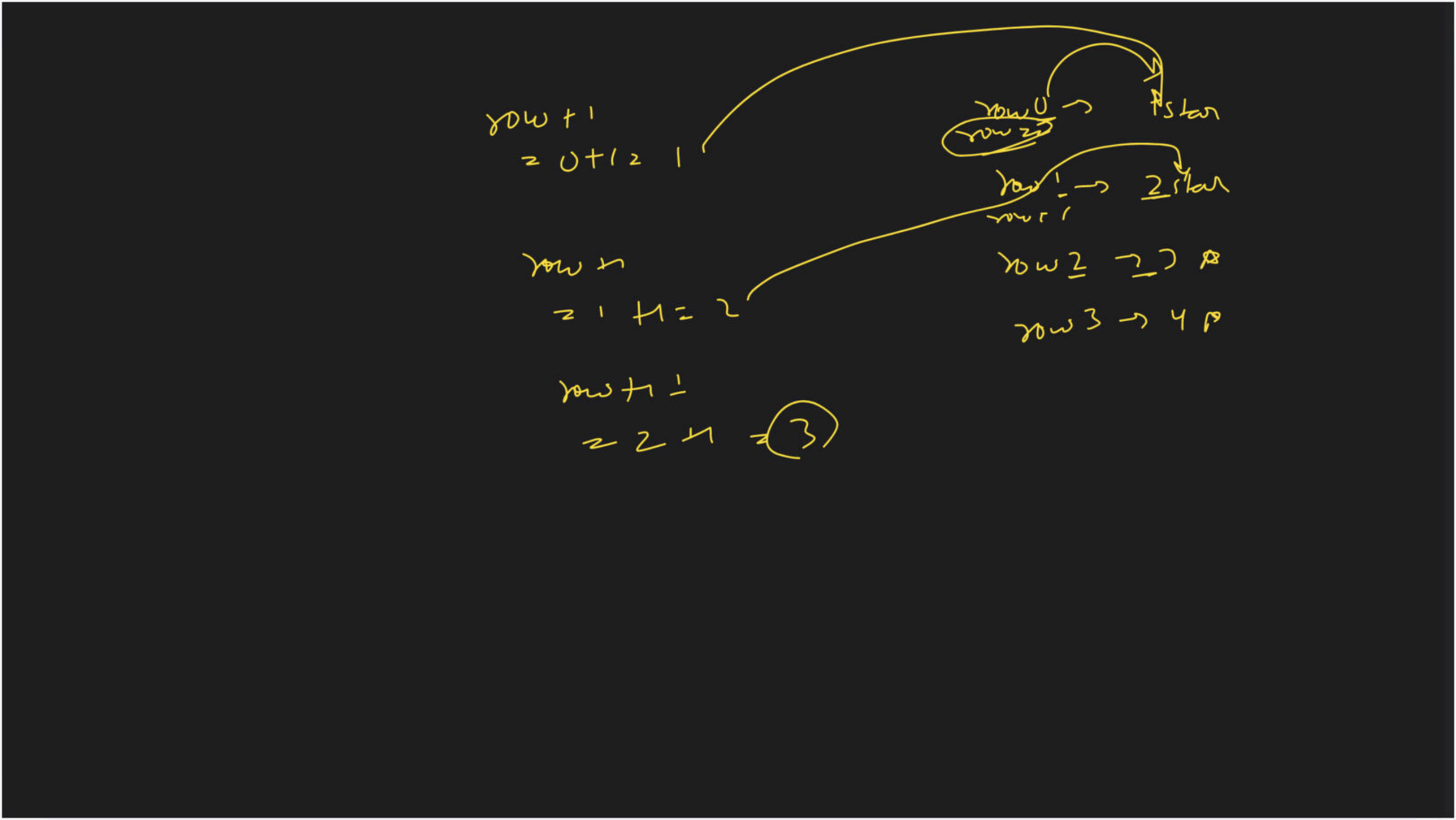
n = 4

n = 4

n - 3

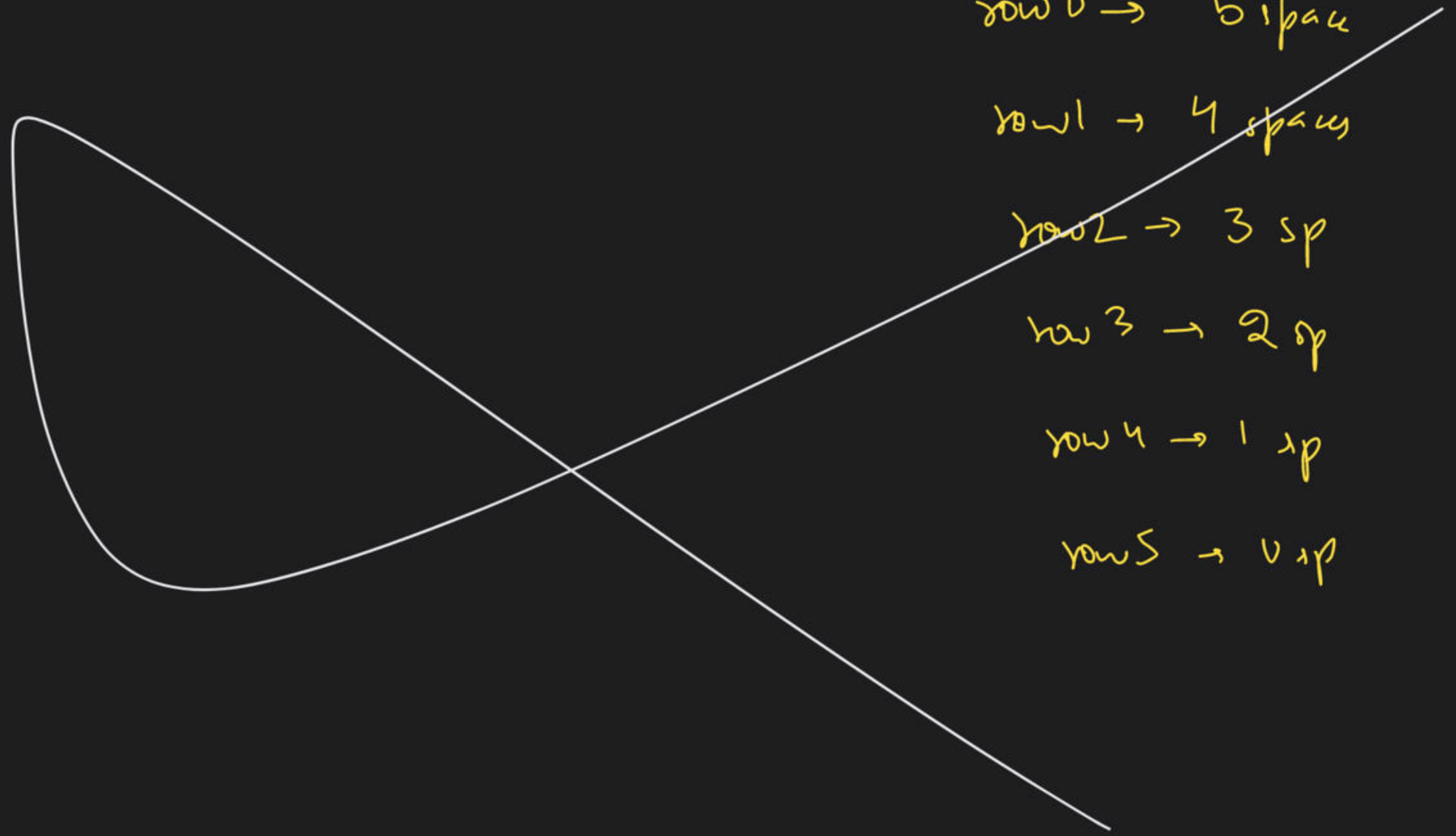
n - [row + 1]

n - row - 1


row+1
=0+1=1
row+1
=1+1=2
row+1=
=2+1=3
row0 → 1star
row1 → 2star
row2 → 3
row3 → 4

r - [row + 1]

+ x - = ⊖ n - row - 1


row 0 → 5 spaces
row 1 → 4 spaces
row 2 → 3 sp
row 3 → 2 sp
row 4 → 1 sp
row 5 → 0 sp

Outer loop

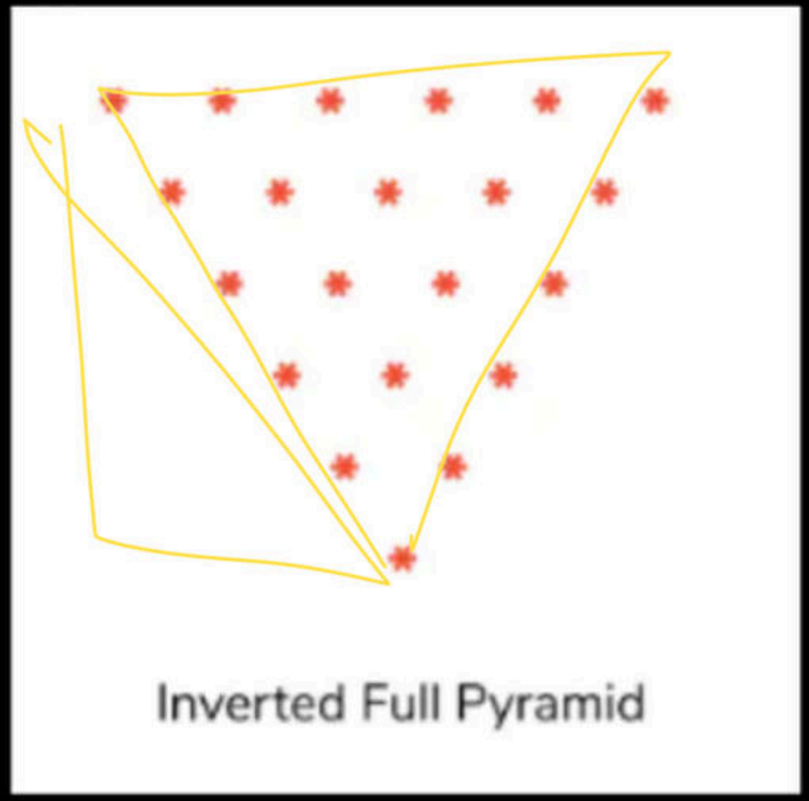
```
for ( int row = 0;   row < 6;   row = row + 1 )
{
```

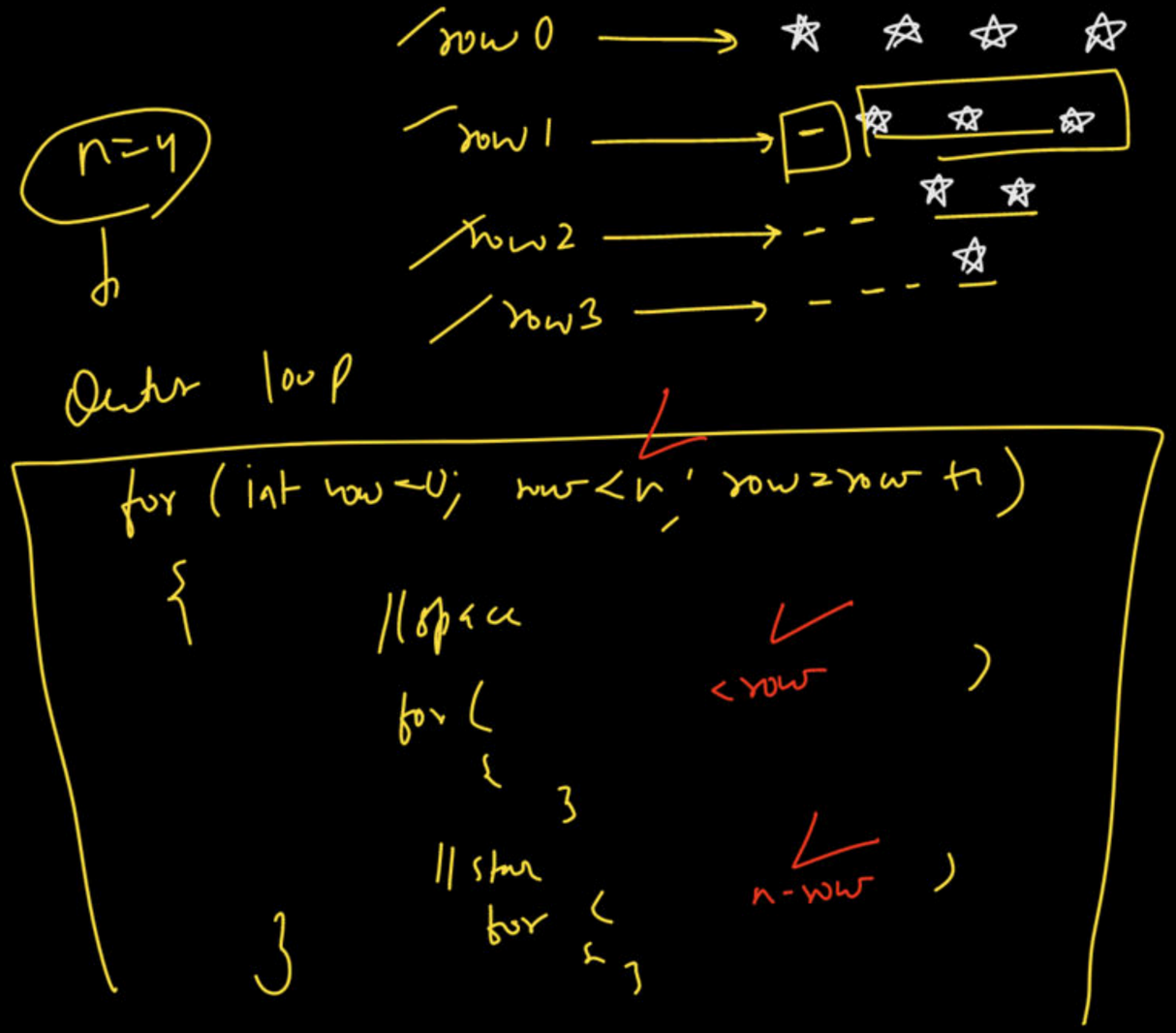
Inner loop

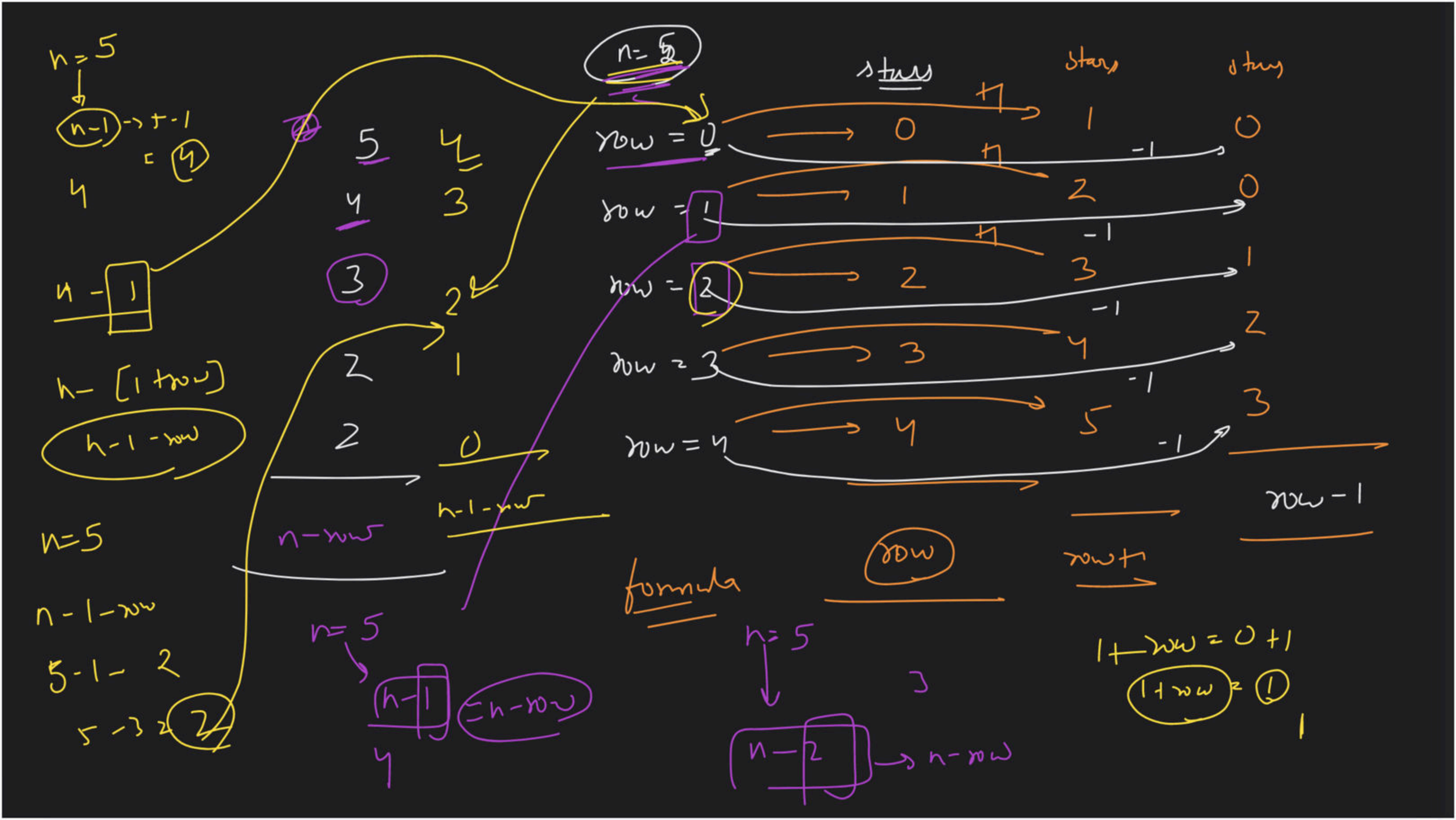
```
for (                          ) ← space
{

}

for(                           ) ← stars
{

}

cout << endl;
```

```
}
```

Inverted Full Pyramid






n = 5
n-1 -> 5-1 = 4
4
4 - 1
n - [1 + row]
n-1-row
n=5
n-1-row
5-1-2
5-3=2
5 4
4 3
3
2
2 1
2 0
n-row
n-1-row
n= 5
n-1
=n-row
4
n= 5
row = 0
row = 1
row = 2
row = 3
row = 4
stars
0 1
1 2
2 3
3 4
4 5
+1
-1
row
formula
n= 5
3
n-2
n-row
row+1
row - 1
1 + row = 0 + 1
1 + row = 1
1

row = 4

1 + row = 1 + 4

1 + row = 5

n = 5

n - 3 = 5 - 3

n - 3 = 2

2

n = 5

n - 5 = 5 - 1 -

n - 5 = 0

0

row = 0

row = 1

row = 2

row = 3

row = 4

row = 2

1 + row = 2 + 1

1 + row = 2

Start

n = 5

4 — n - 1

3

n - [1 + row] = n - 1 - row

2 — n - 3

1

n - [1 + row]

n - 1 - row

1 — n - 5

n - [1 + row]

= n - row - 1

row = 0

1 + row = 0 + 1

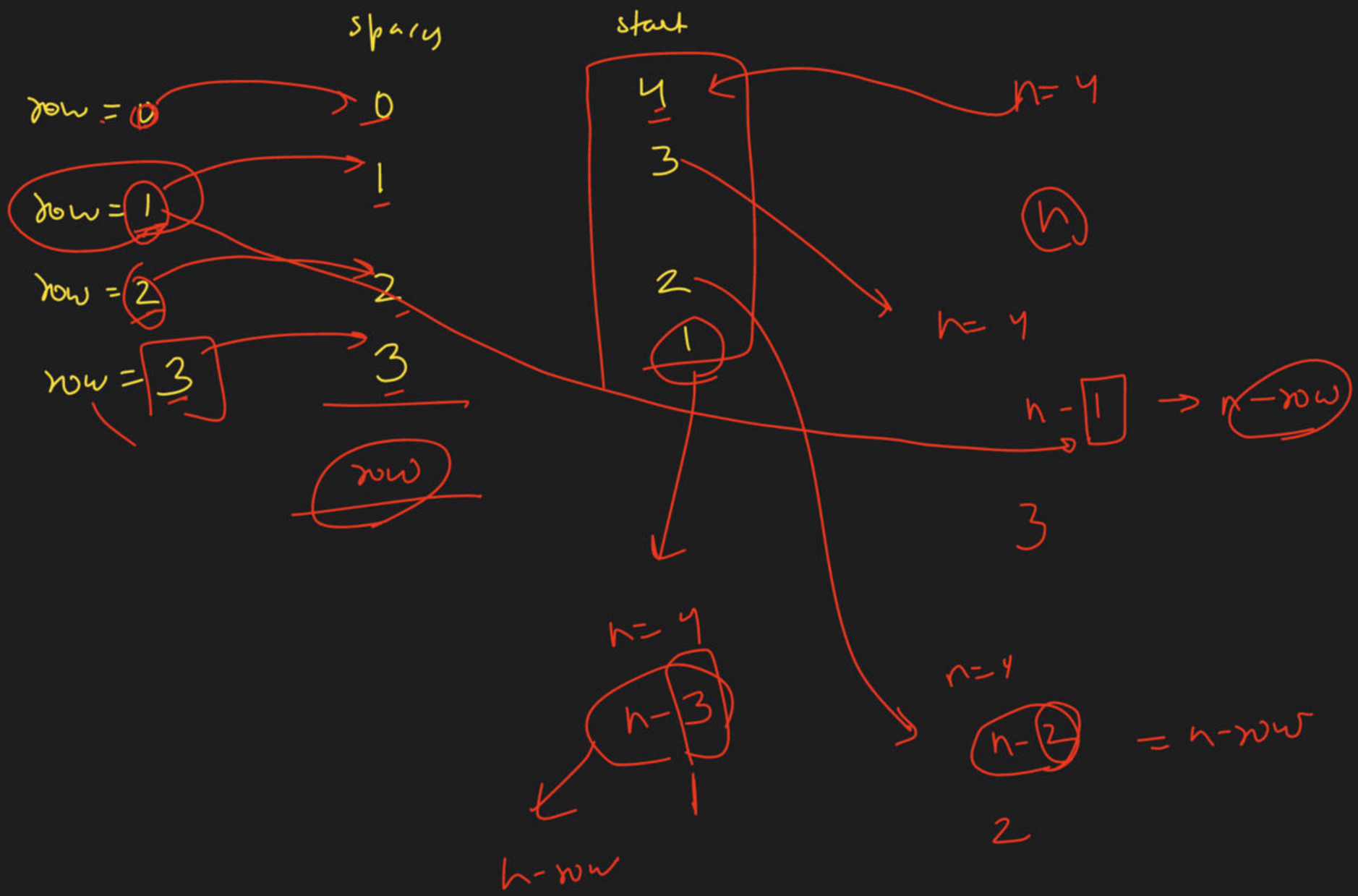
1 + row = 1

n = 5

n - 1 = 5 - 1

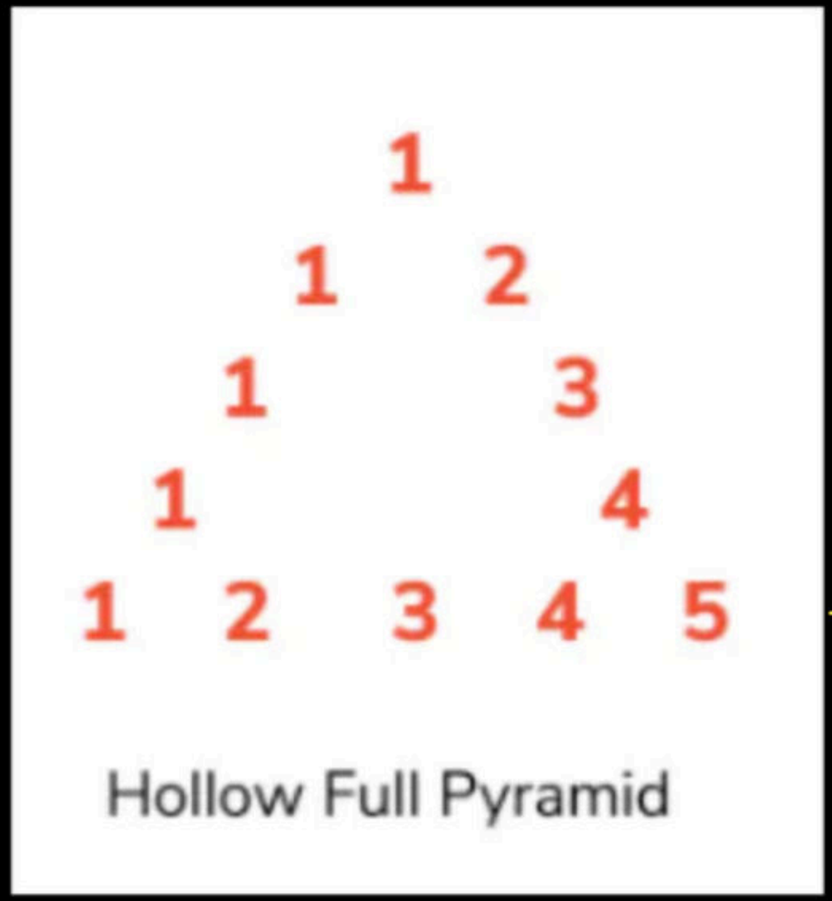
n - 1 = 4

4

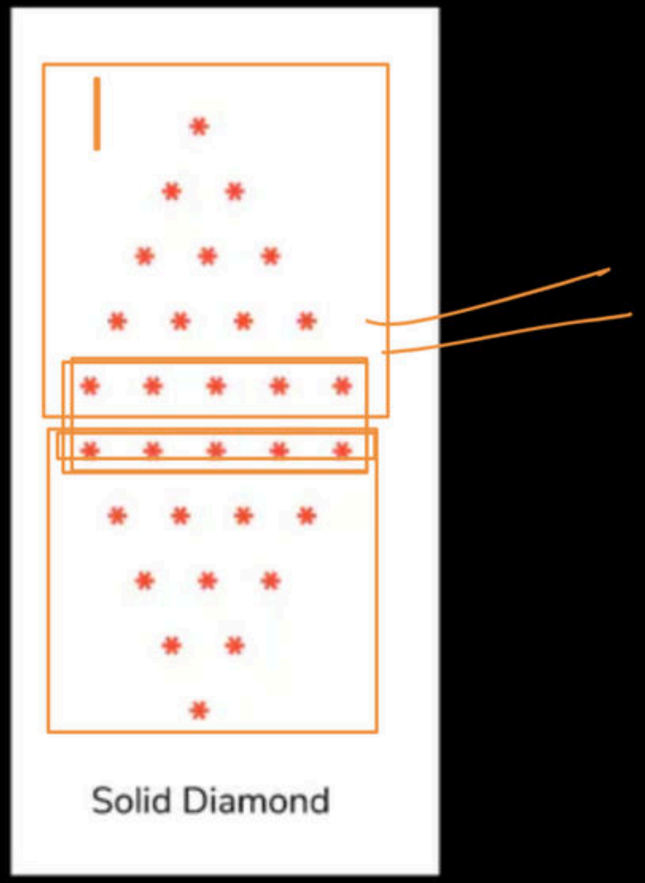

sparse
start
row = 0
0
row = 1
1
row = 2
2
row = 3
3
row
4
3
2
1
n = 4
n
n = 4
n - 1
n - row
3
n = 4
n - 3
n - row
n = 4
n - 2
= n - row
2

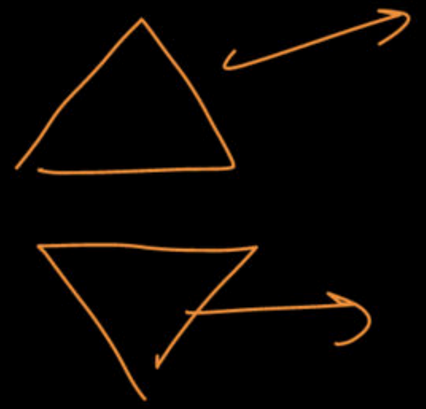
Numeric Full Pyramid

Numeric Hollow Pyramid

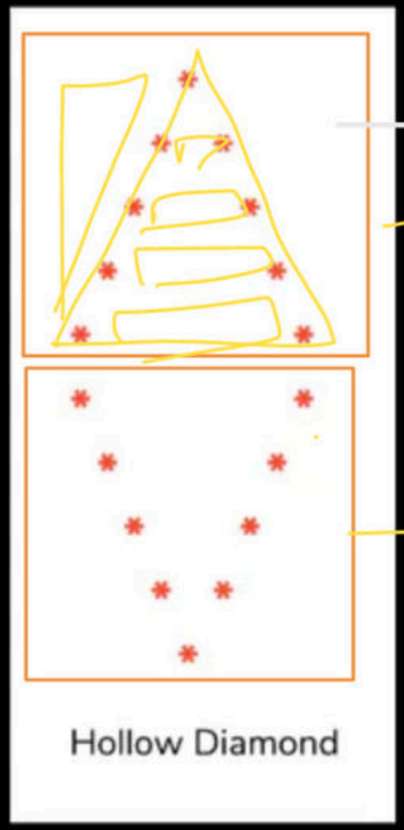


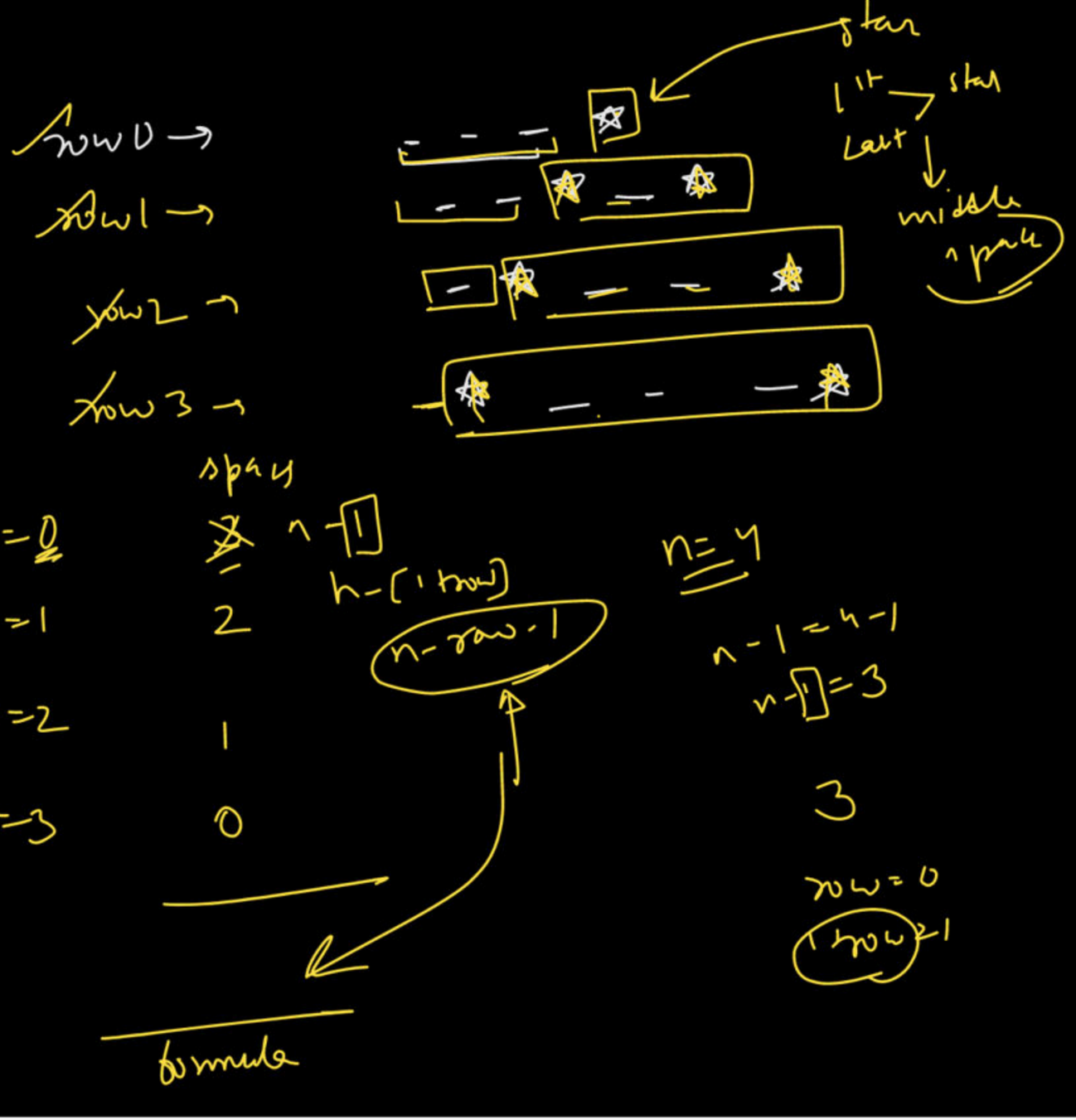
Solid Diamond





Hollow Diamond





2×0 -1 = -1

row= 0 Outer loop

2*row+1

1

3

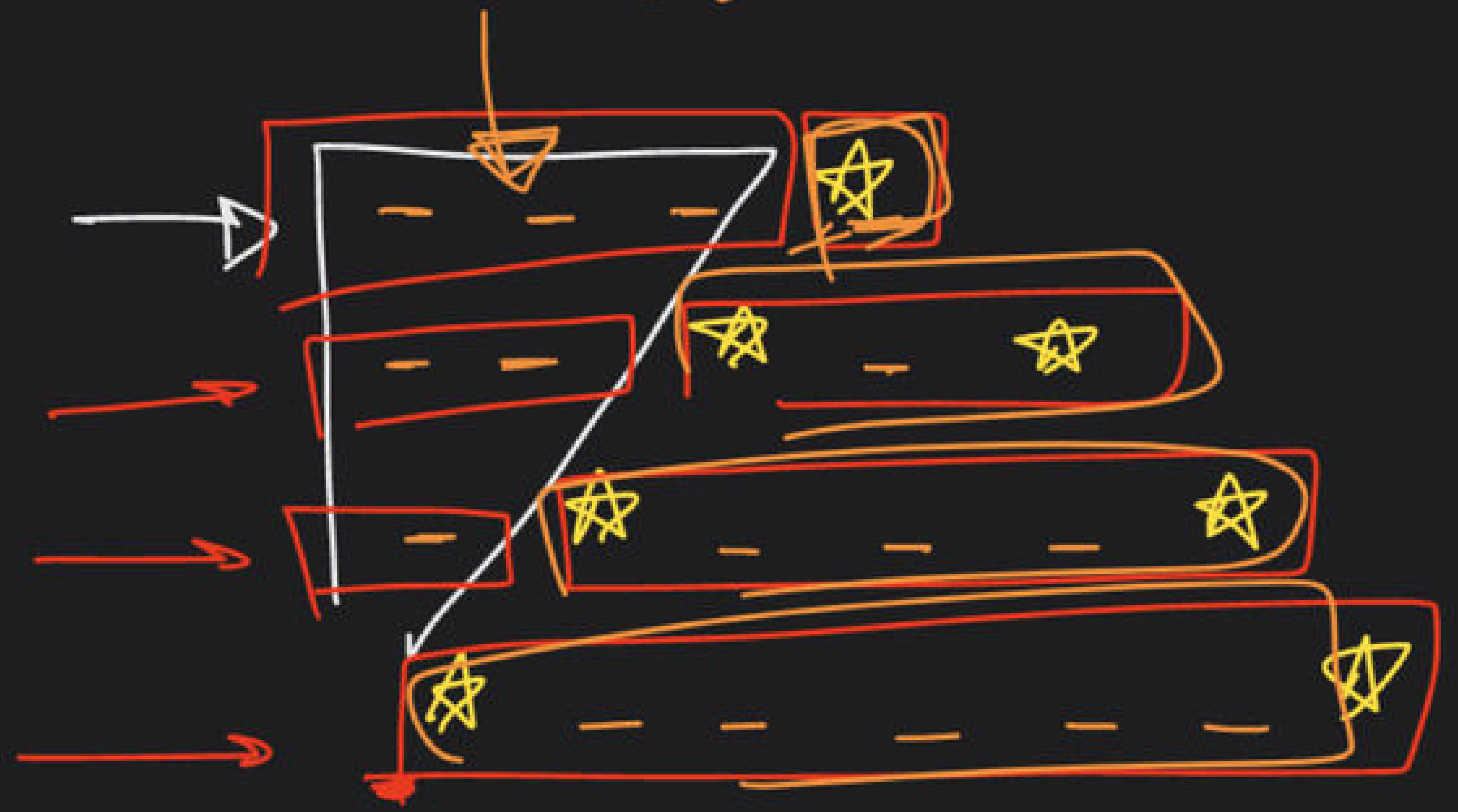
5

7

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

  //star
     2*row+1

}
```

Why

How



row=0

1+row= 0+1

row 0	3
row 1	2
row 2	1
row 3	0

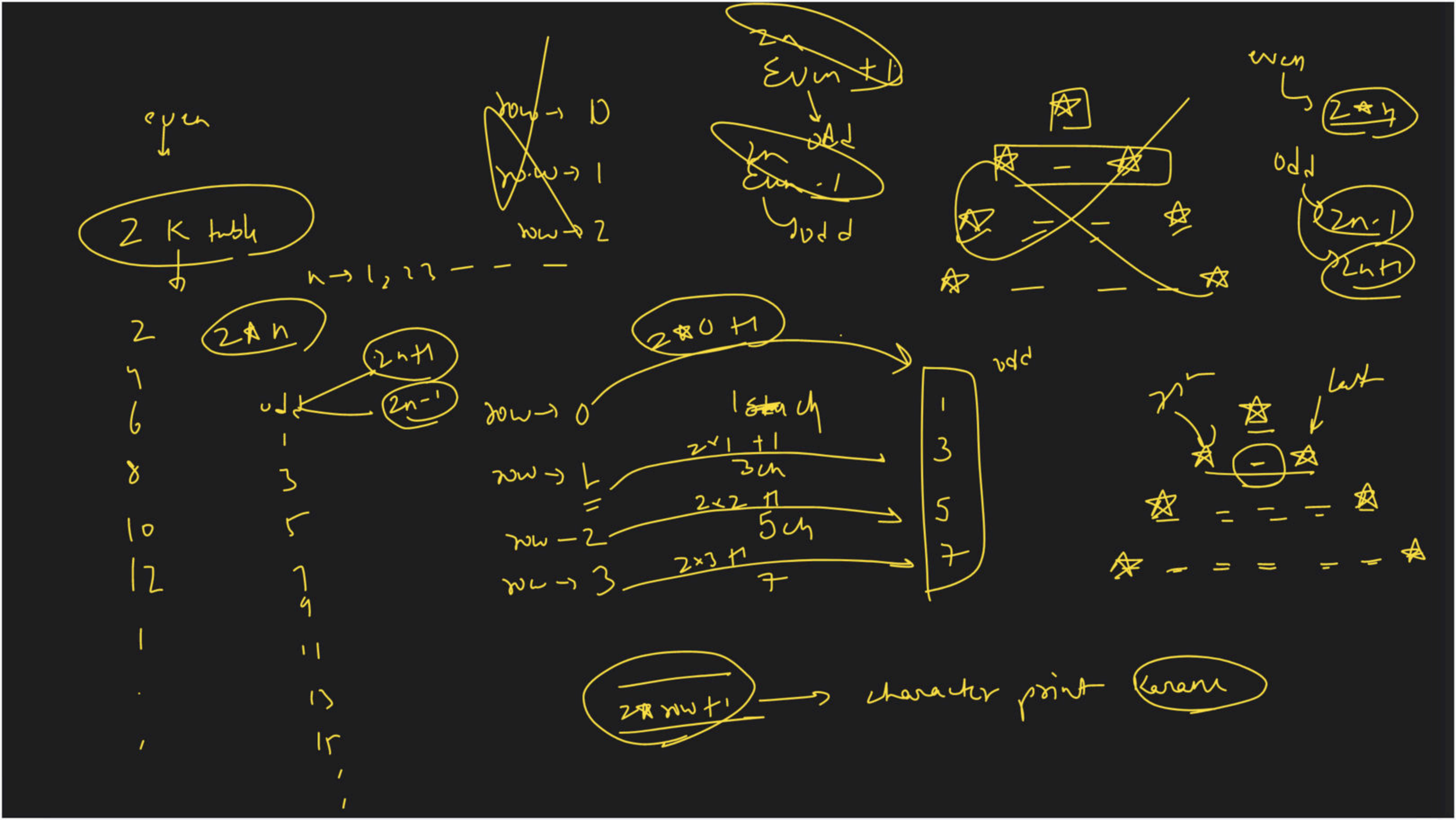
n=4

n-1 = 4-1

n-1 = 3

n-(1+row)

n-1-row


even
2 K table
2 * n
2
4
6
8
10
12
1
odd
2n+1
2n-1
1
3
5
7
9
11
13
15
row → 0
row → 1
row → 2
row → 3
n → 1, 2, 3 — — —
2 * 0 + 1
1st ch
2*1 + 1
3ch
2*2 + 1
5ch
2*3 + 1
7
1
3
5
7
odd
even + 1
odd
even - 1
odd
even
2 * n
odd
2n-1
2n+1
2 * row + 1
character print karana
first
last

```
for ( int col=0;    col < 2*row+1 ; col=col+1)
    {
    }
```

2row 2row+1

1 + 2 + 3 + 4 + 5

$< 2nw + 3$

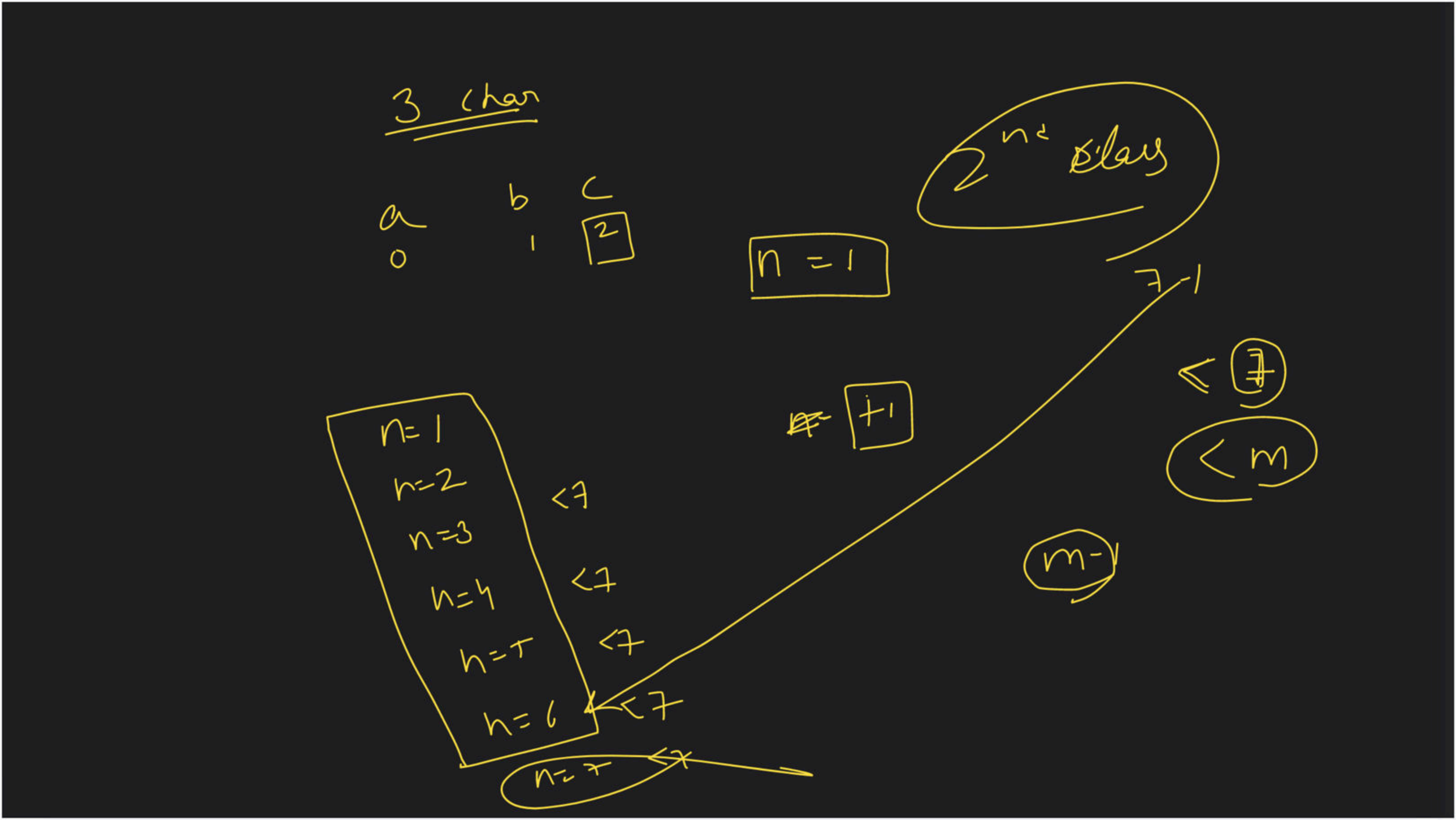
$2nw + 2$

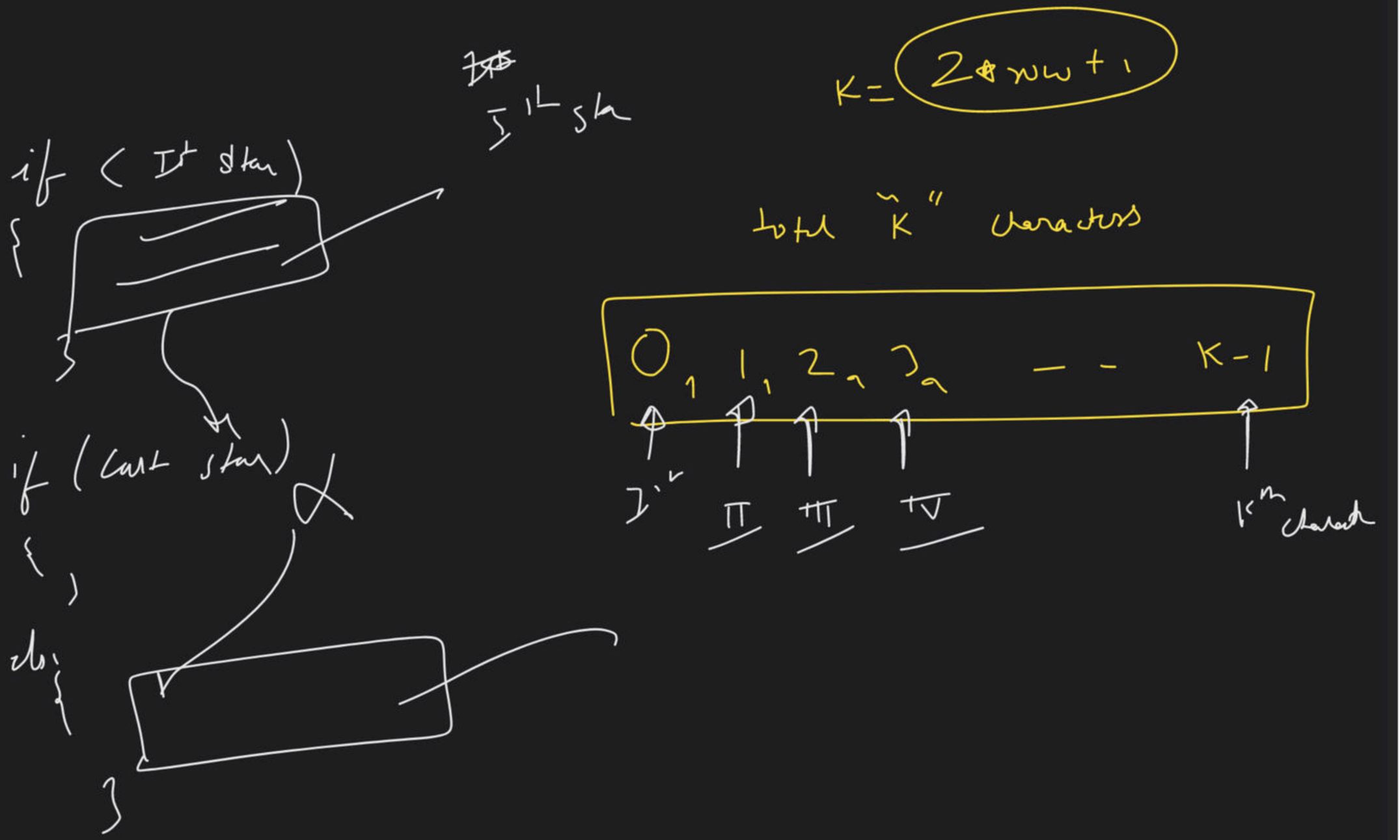
$< 2nw + 1$

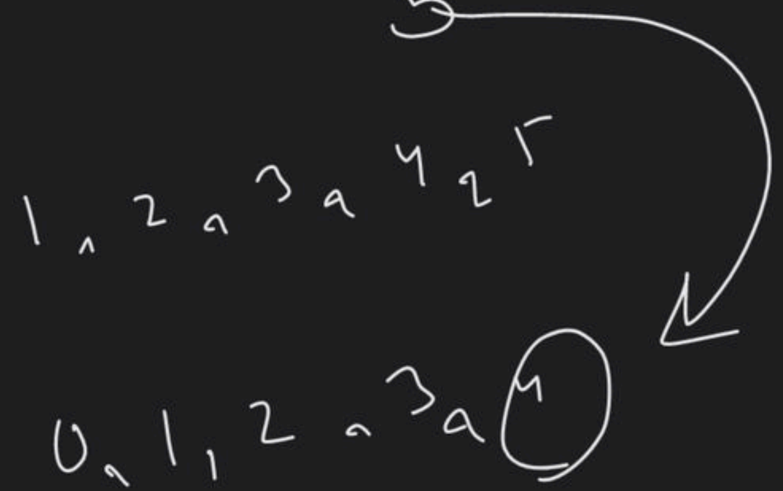
$2nw$

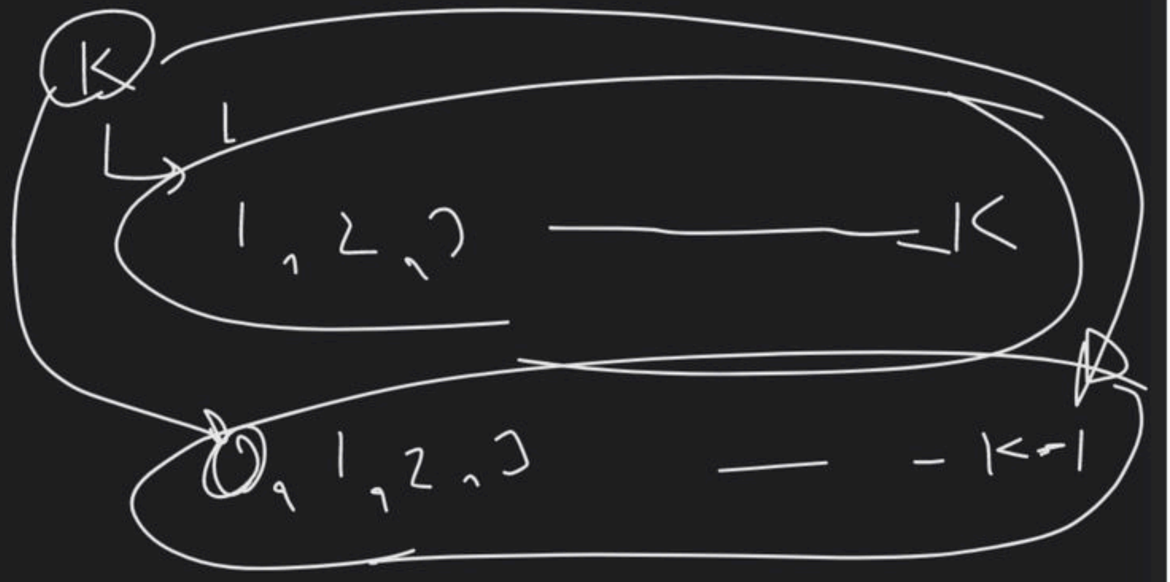
$< 2nw + 2$

$2nw + 1$

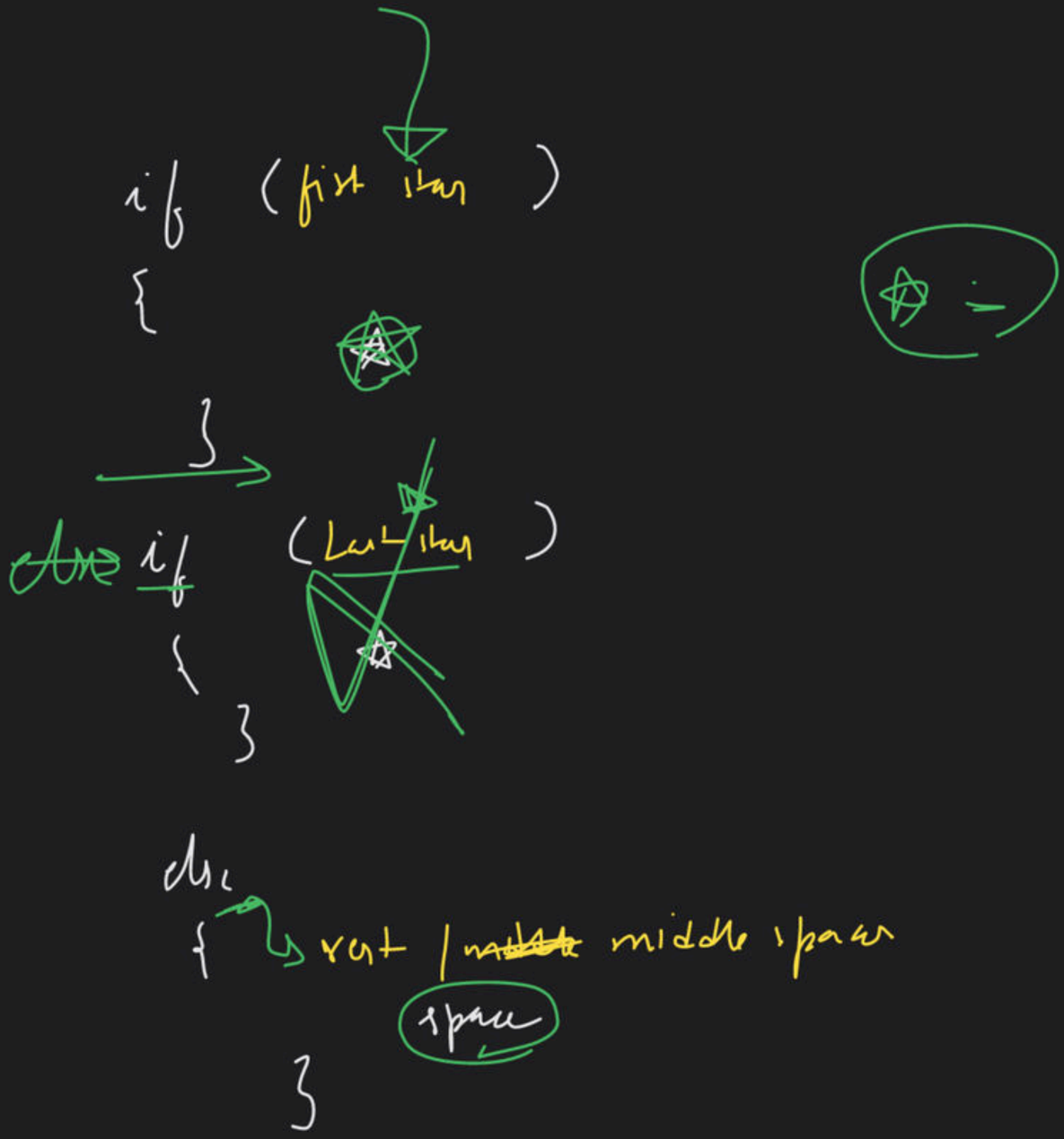

3 char
a
b
c
0
1
2
n = 1
2^n classes
n-1
< 7
< m
i+1
n=1
n=2
n=3
n=4
n=5
n=6
n=7
<7
<7
<7
<7

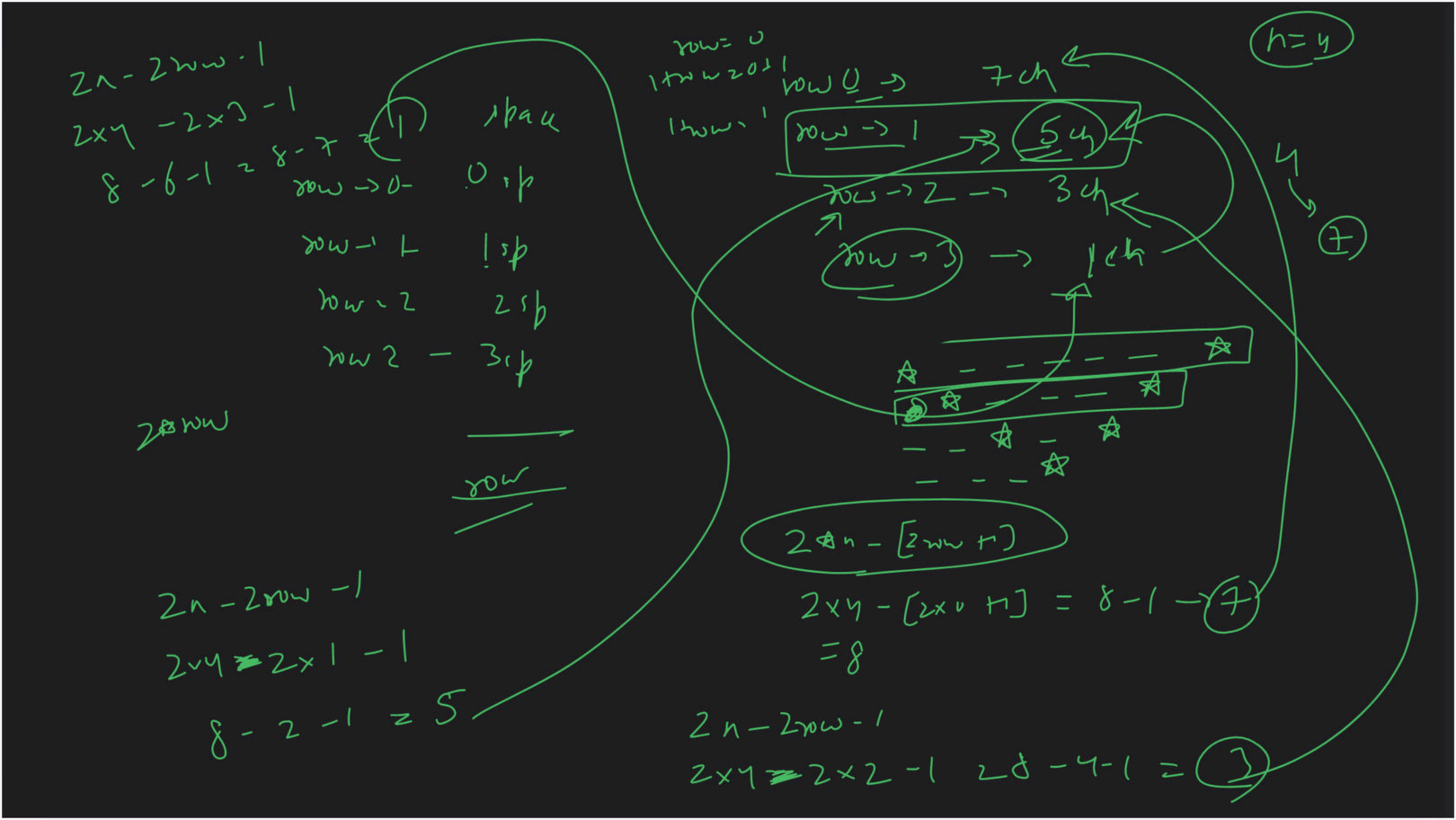

K = 2 * no + 1
total "K" characters
0 1 2 3 - - K-1
1st
II
III
IV
Kth character
if (I st stat)
{
}
if (call stat)
{
}
do
{
}
}


5
1, 2, 3, 4, 5
0, 1, 2, 3, 4

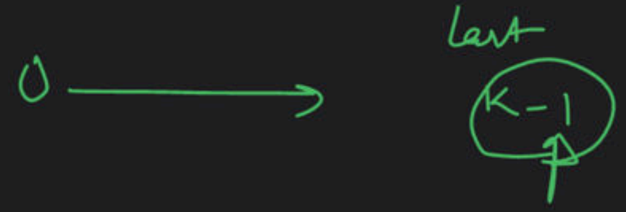

K
1
1, 2, 3
K
0, 1, 2, 3
K-1


first star


if (first star)
{
}
else if (last star)
{
}
else
{
return middle space
}


2n - 2row - 1
2×4 - 2×3 - 1
8 - 6 - 1 = 8 - 7 = 1
1 pair
row → 0 - 0 pair
row - 1 - 1 pair
row - 2 - 2 pair
row 2 - 3 pair
2 * row
row
2n - 2row - 1
2×4 = 2×1 - 1
8 - 2 - 1 = 5
row = 0
1 to row 2021
row 0 → 7 ch
row → 1 → 5 ch
row → 2 → 3 ch
row → 3 → 1 ch
h = 4
4 → 7
2 * n - [2row + 1]
2×4 - [2×0 + 1] = 8 - 1 → 7
= 8
2n - 2row - 1
2×4 = 2×2 - 1
2 8 - 4 - 1 = 3

Total K characters

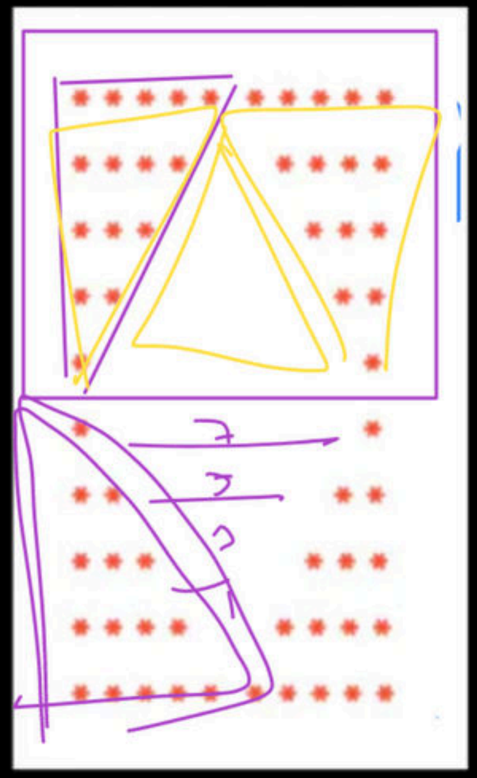


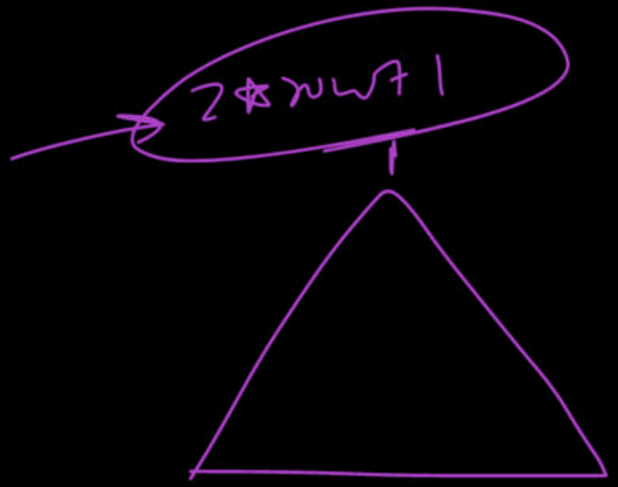
Total → $2n - 2ww - 1$

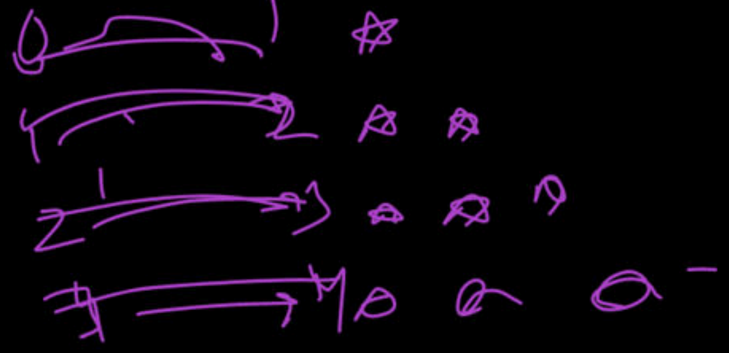
Last → $2n - 2ww - 1 - 1$

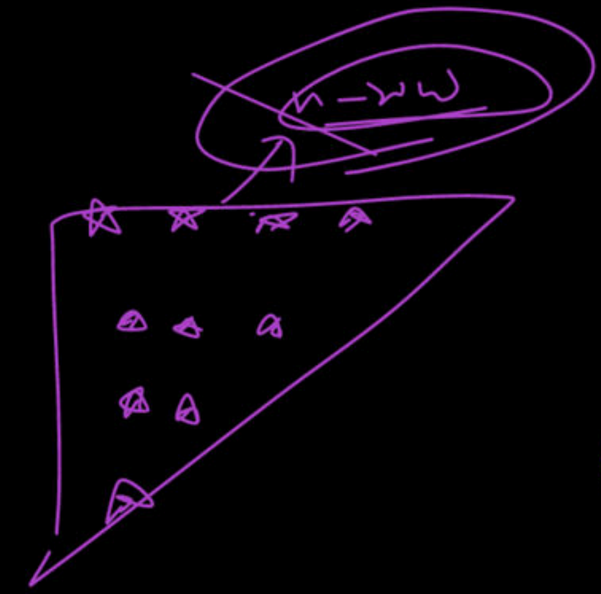
$= 2n - 2ww - 2$

Flipped Solid Diamond









row 0 → 4

row 1 → 3

row 2 → 2

row 3 → 1

n=4

(n)

n-1 = 4-1

n-(1) = 3

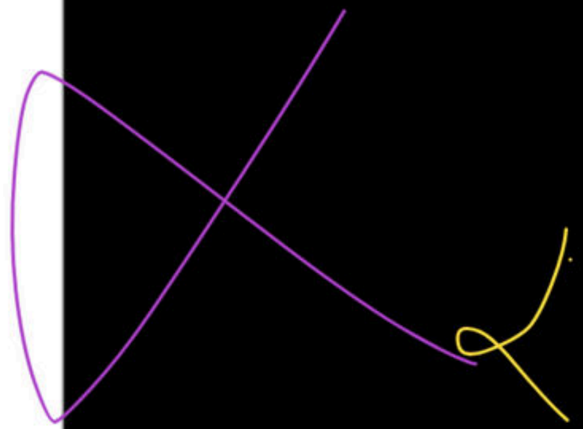
n-row

2n-2row-1

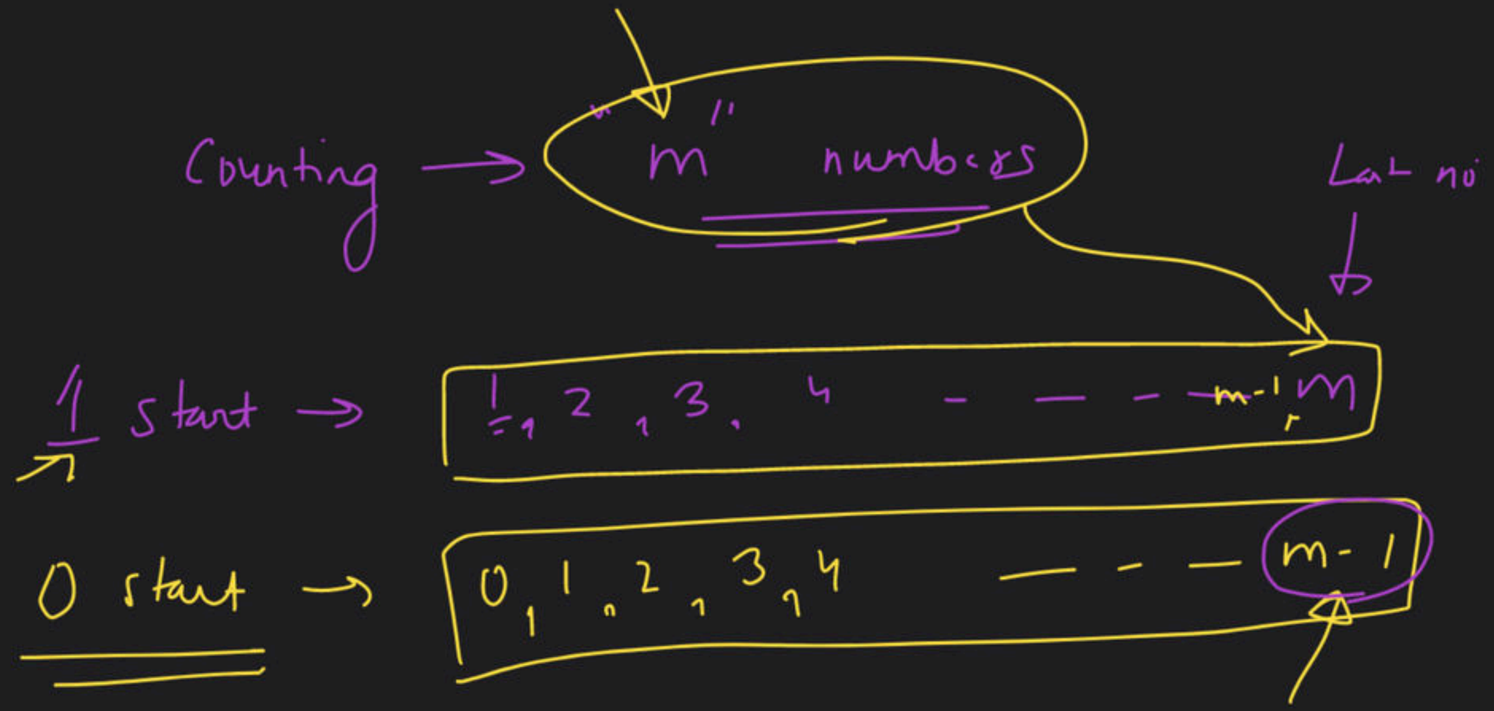
row+1

Fancy Pattern #1

```
3
44
555
6666
555
44
3
```



Fancy Pattern #2

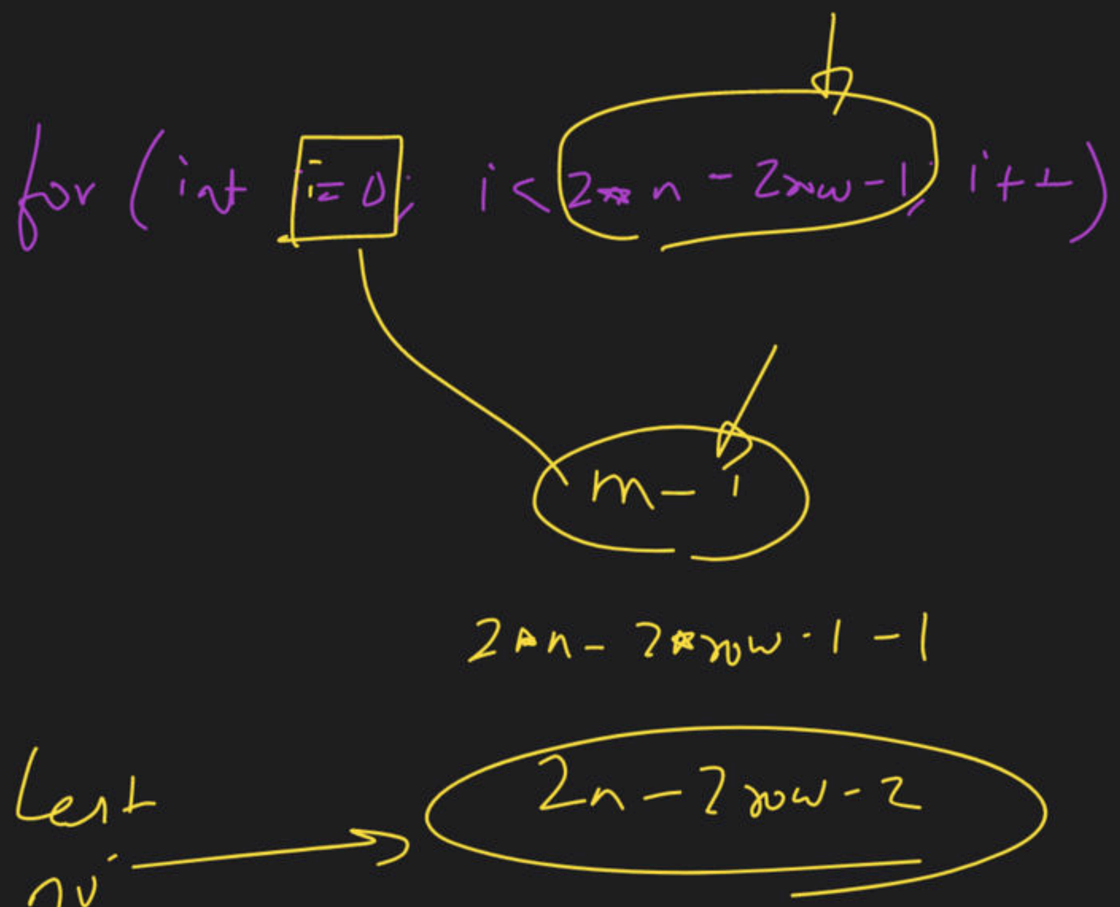


for (int col = 0; col < m - row; col++)

Last

m - 1

n - row - 1


for (int i=0; i<2*n-2*w-1; i++)
m-1
2*n-2*w-1-1
Last
2n-2w-2

for (int col = 0; col < n - row - 1; col = col + 1)

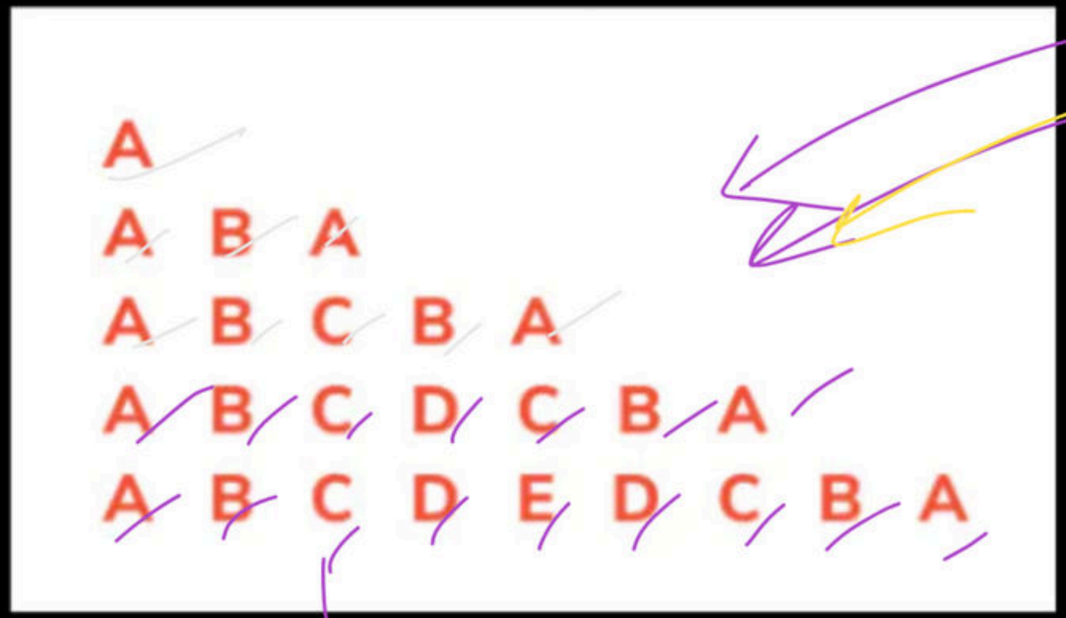
m

last ns -> m - 1

n - row - 1 - 1

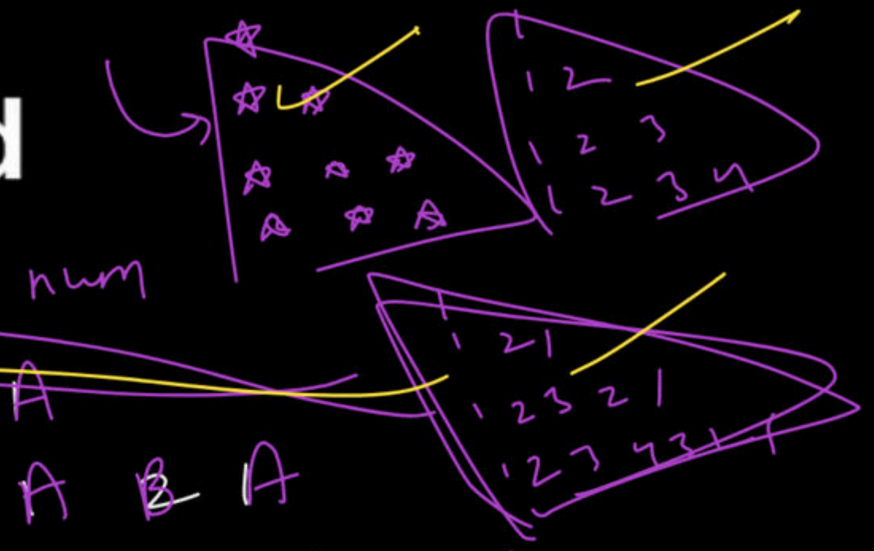
n - row - 2

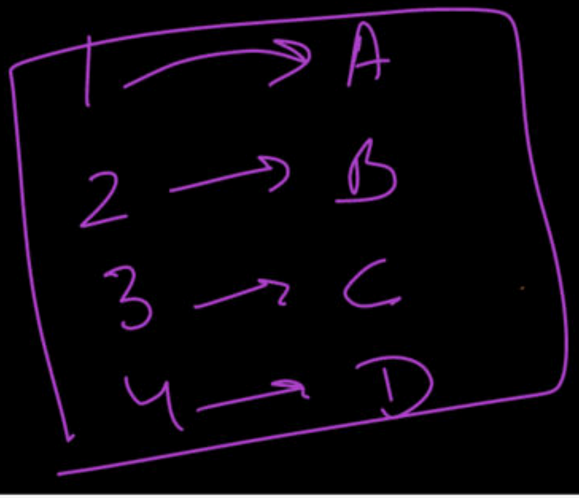
Alphabet Palindrome Pyramid



alphabet

num

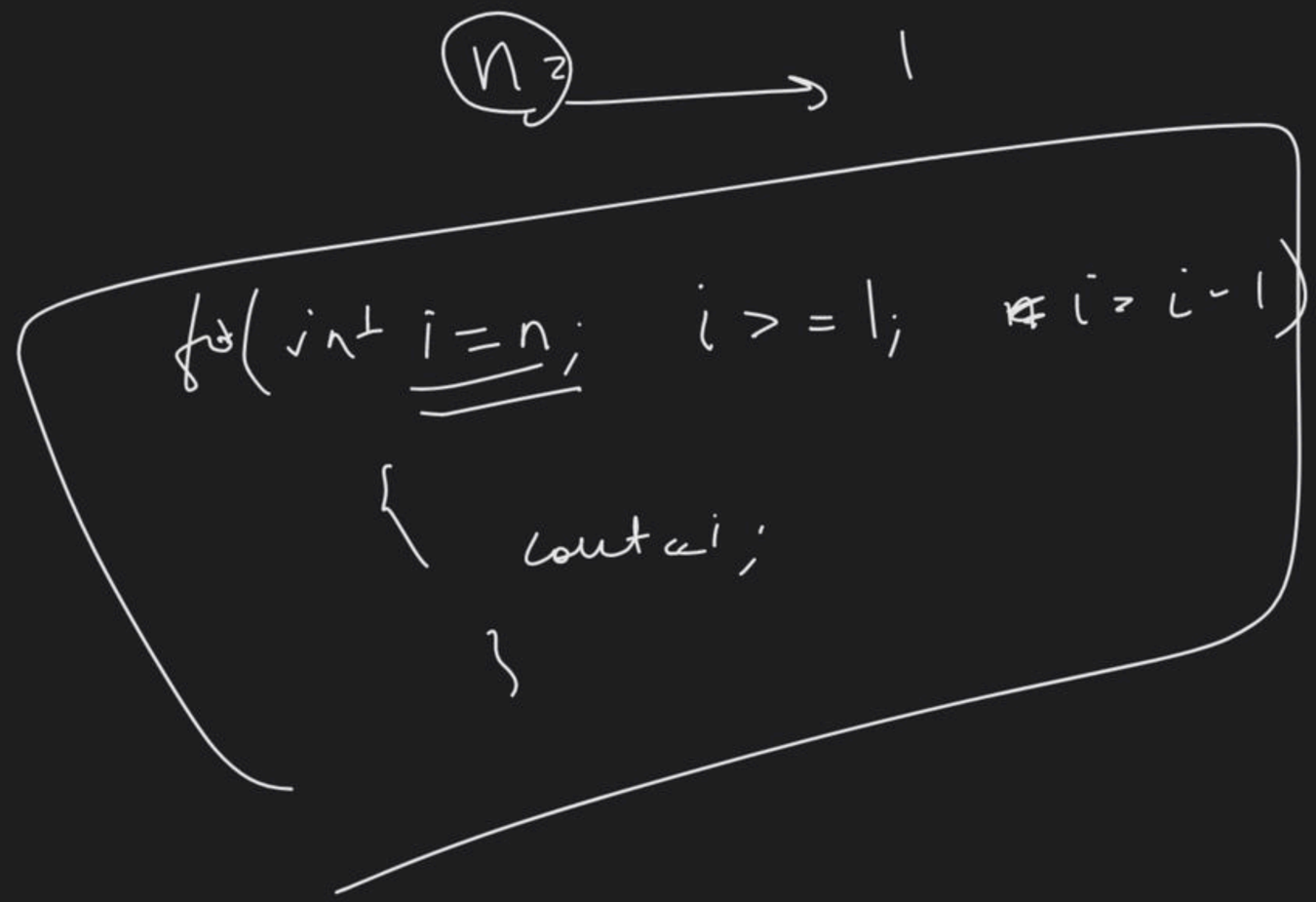




ch = m + 'A' - 1

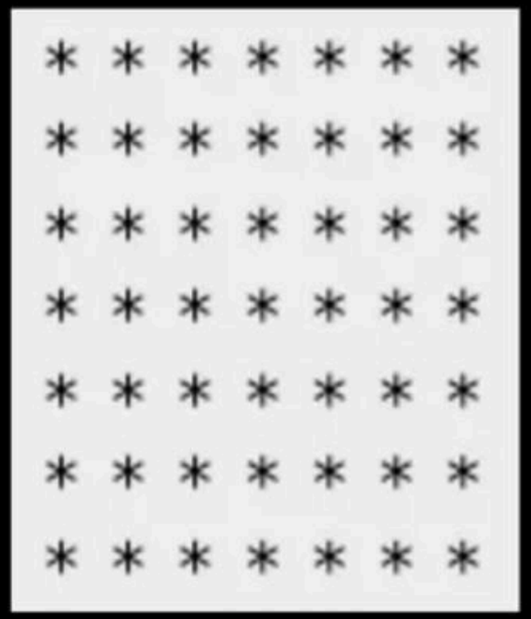
int m = 1

char ch = m + 'A'

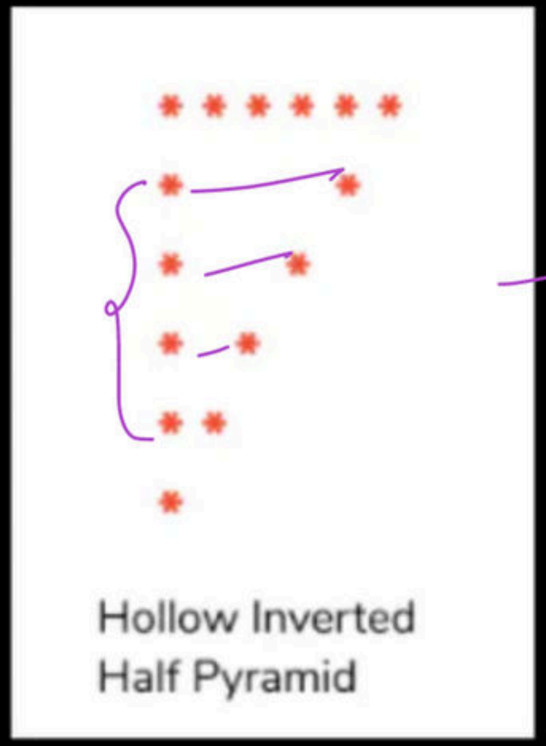

n
3
1
for(int i=n; i>=1; i=i-1)
{
cout<<i;
}

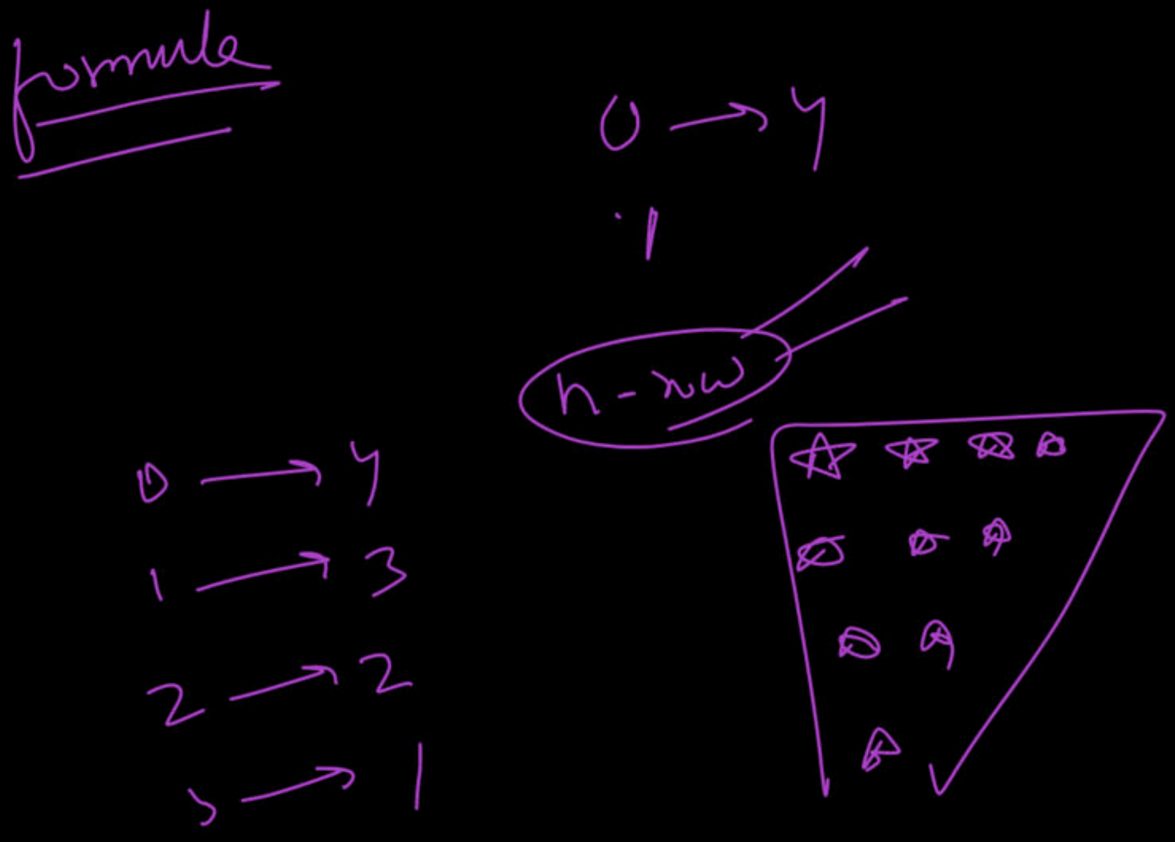
Homework:

Let's start with Simple Stuff
Solid Square Pattern[on the LEFT] && Hollow Square Pattern[on the RIGHT]

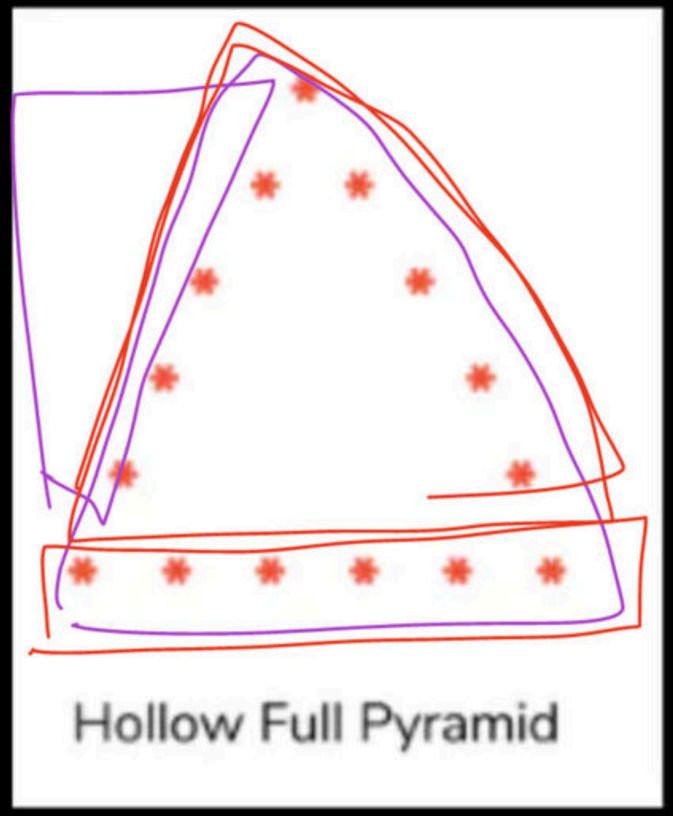


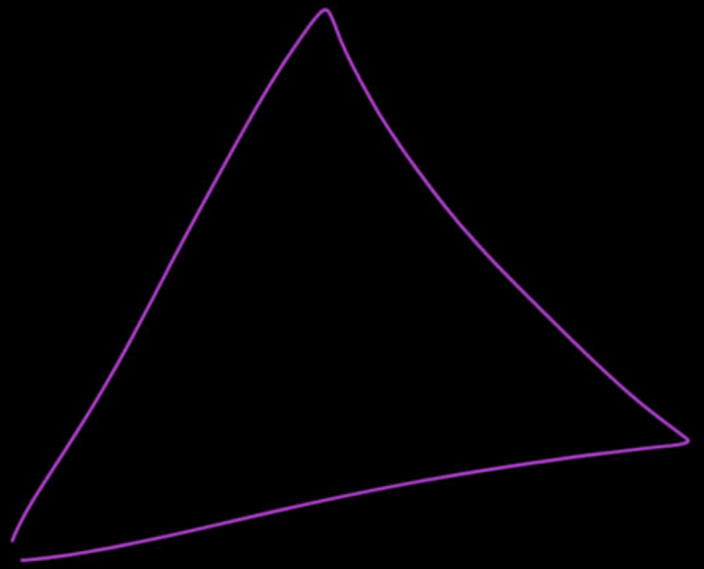
Hollow Inverted half Pyramid



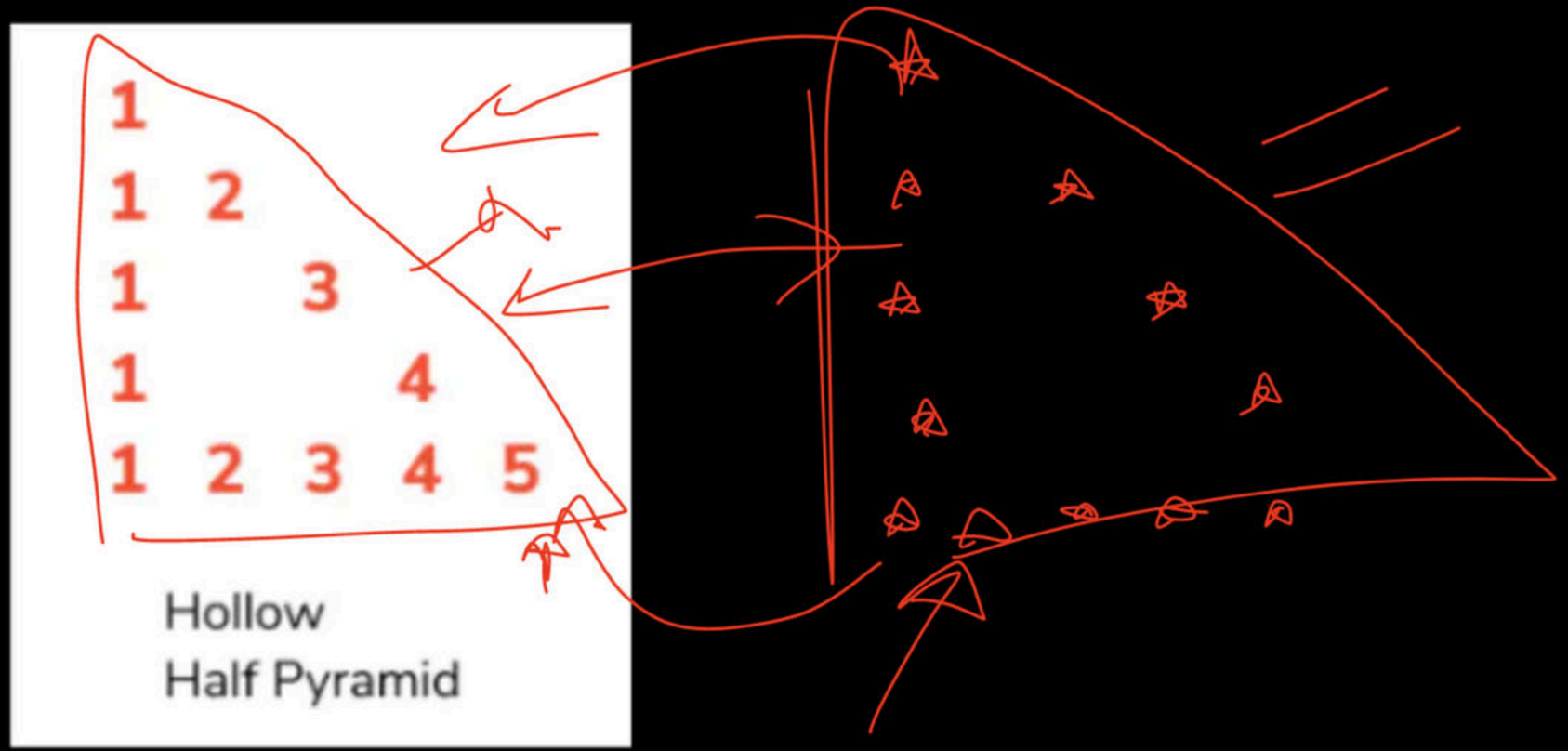


Hollow Full Pyramid

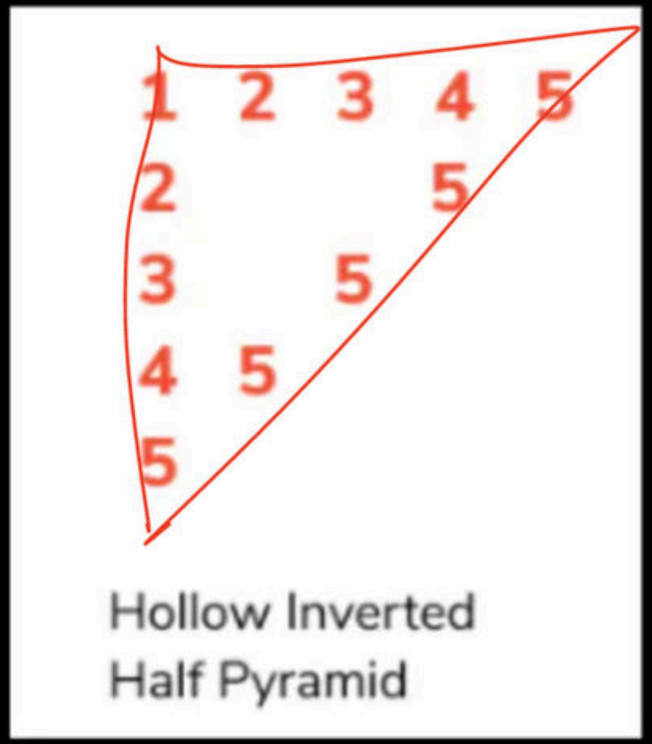


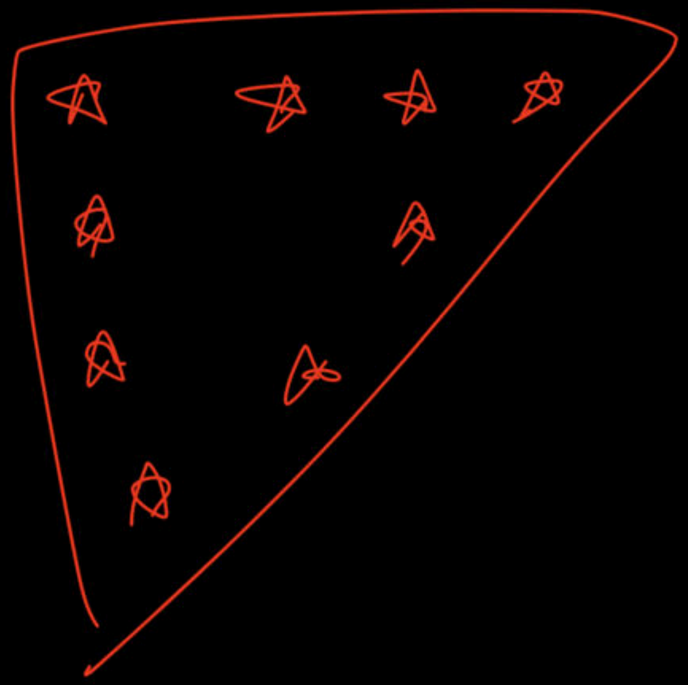


Numeric Hollow Half Pyramid

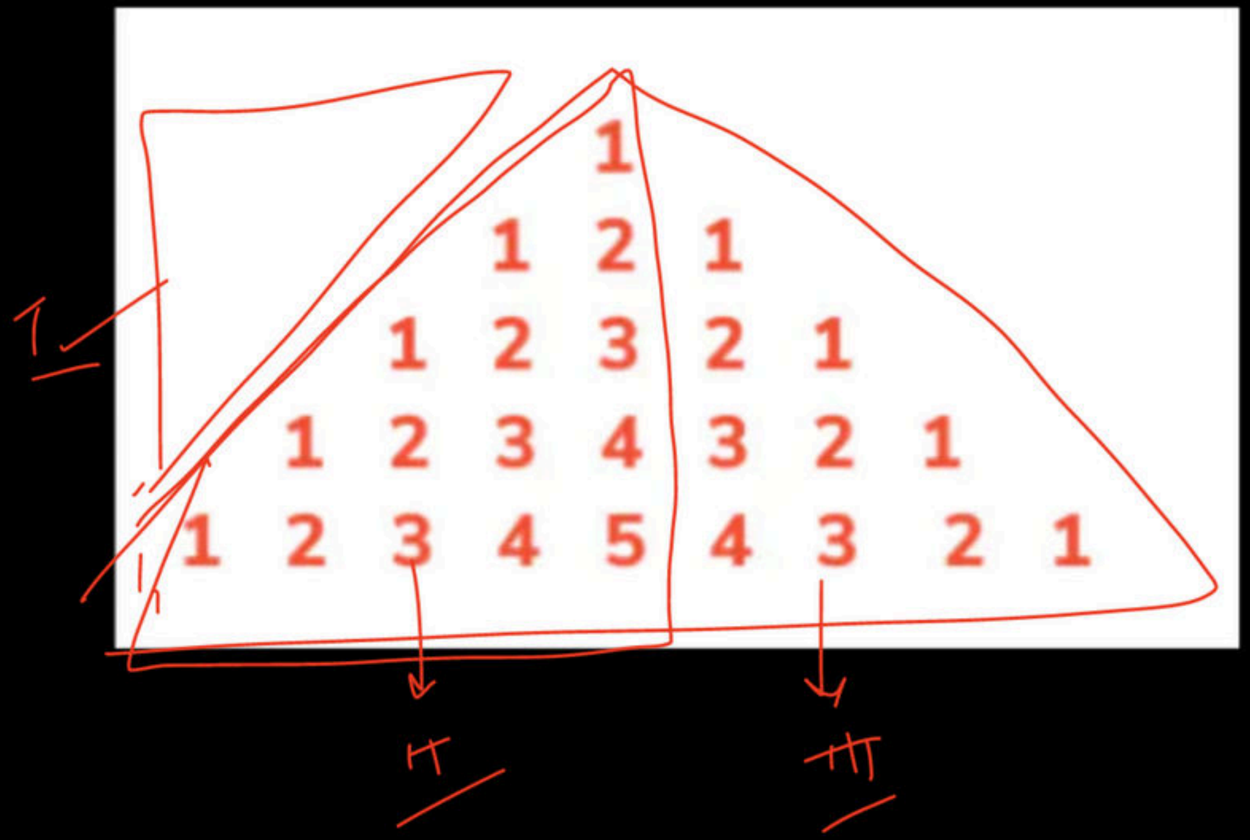


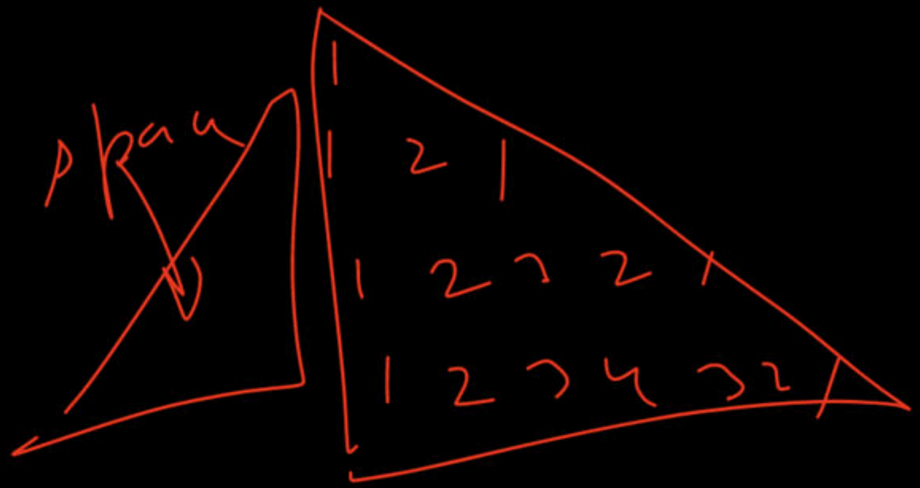
Numeric Hollow Inverted Half Pyramid



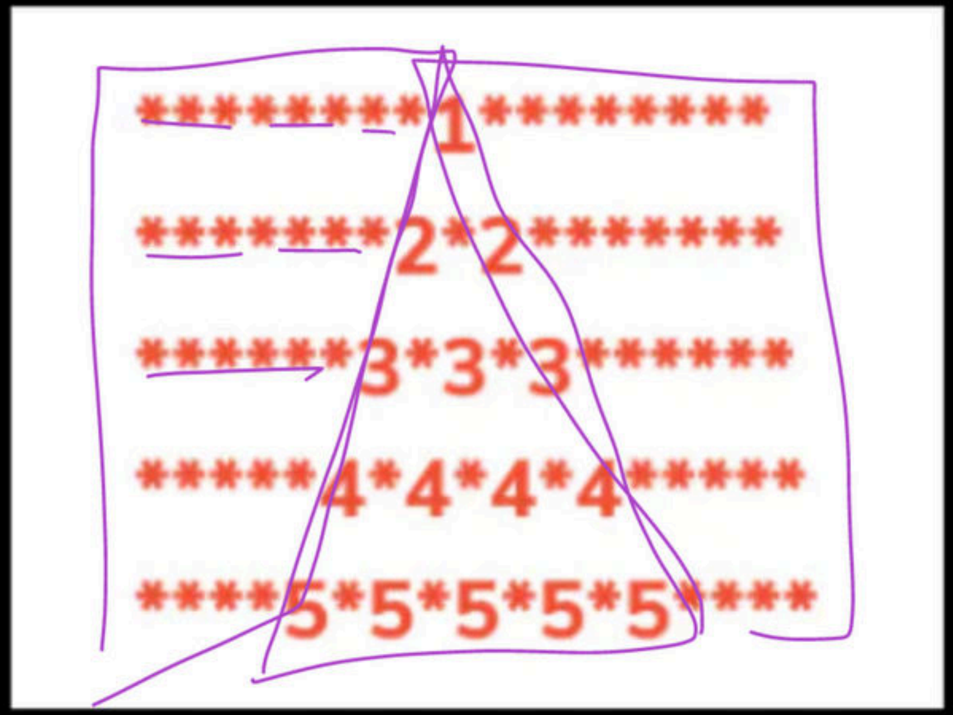


Numeric Palindrome Equilateral Pyramid





Fancy Pattern #1



n = 5

8
7
6
5
4

n+3

4
3
2
1

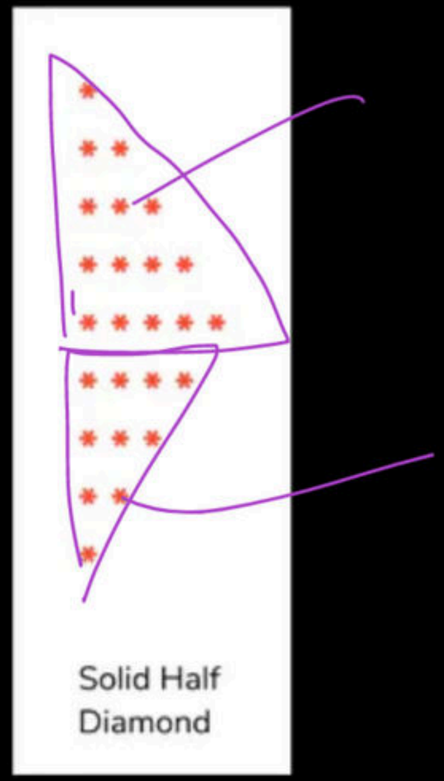
n+row+3

n+3-row

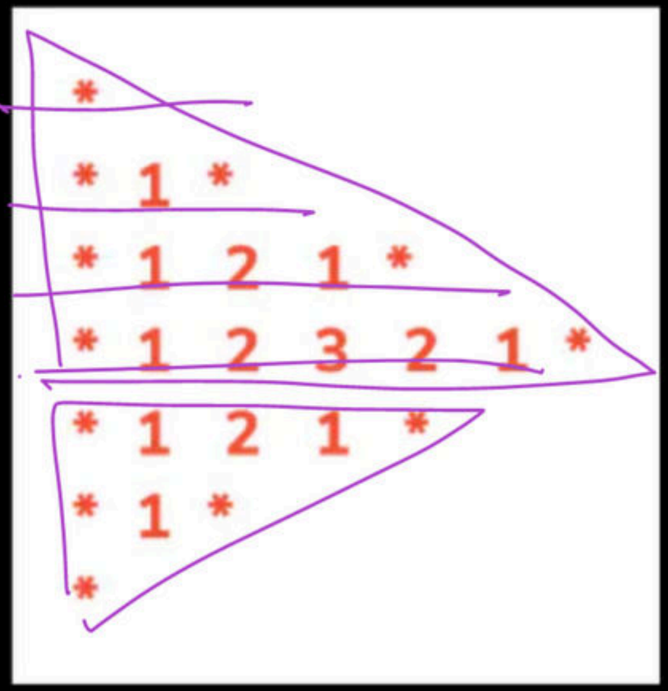
SB -0

ST-1

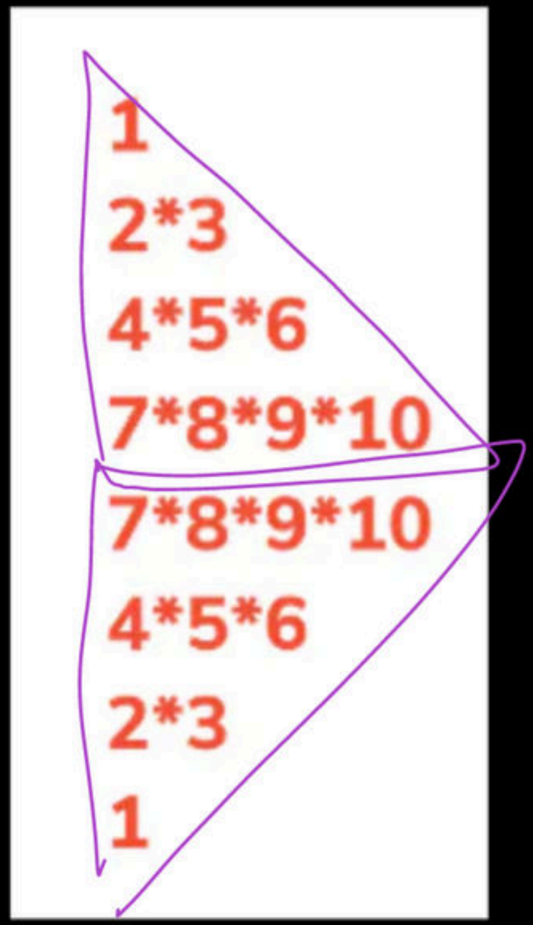
Solid Half Diamond



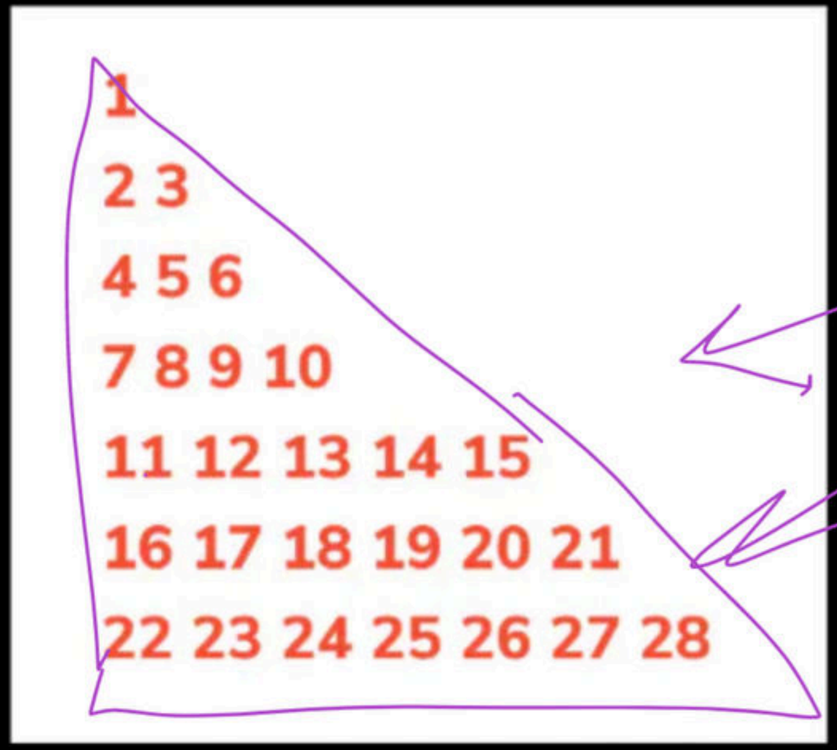
Fancy Pattern #3

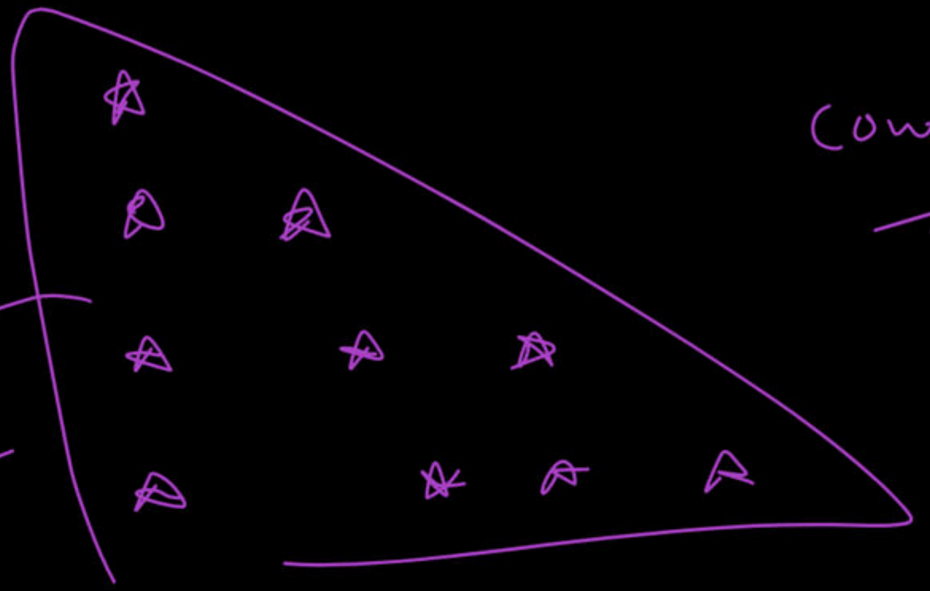


Fancy Pattern #2

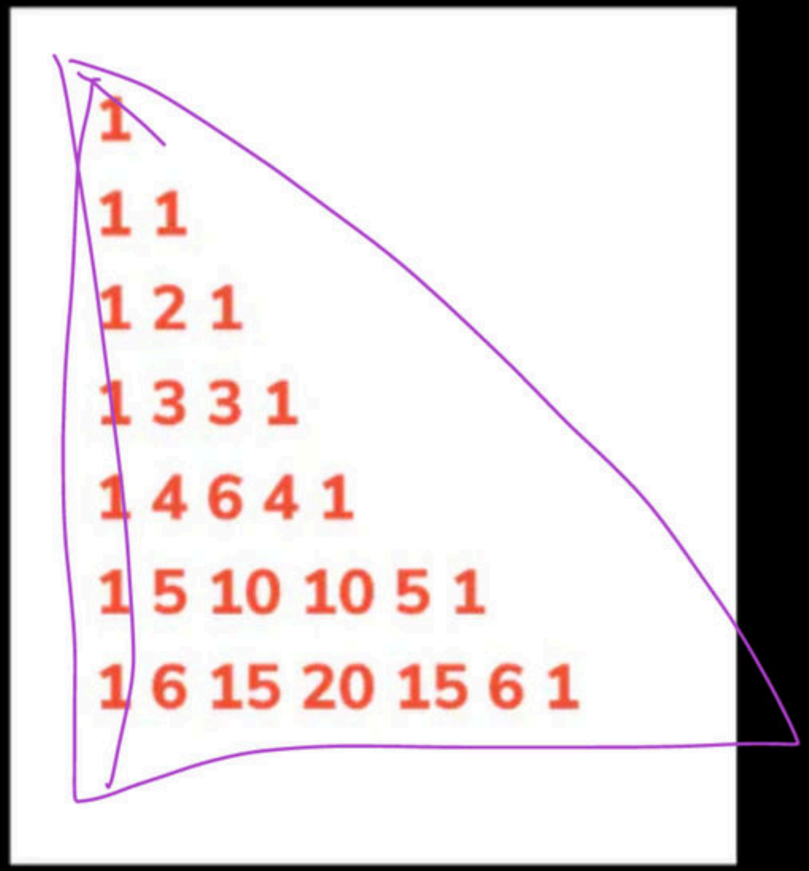


Floyd's Triangle Pattern





Pascal's Triangle Pattern



Butterfly Pattern