

Student

{ Name ✓  
Roll no ✓

(50)

int s1 = su ✓

100 ✓ [ ] [ ]

Roll no → Integer

✓ ✓  
int arr = {

✓ ✓ ✓ ✓  
Saleman 1 = [ ]

Saleman 2 = [

Saleman 3 = {

✓  
1000  
Combinat  
↓  
1000

(30)

# Two types of 2-D Arrays

Rectangular



row

column  
the length

Logic

Row 1 ← 0	0	1	2
Row 2 ← 1	4	5	6
Row 3 ← 2	7	8	9
	col 1	col 2	col 3

Jagged 2D Array

different length

3x3

1 2 3  
5 6 7 8 9  
10 11 12 13

```
int[][] arr = new int[Row_size][Col_size];
```

3  
↑

3

Logical

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

✓ X

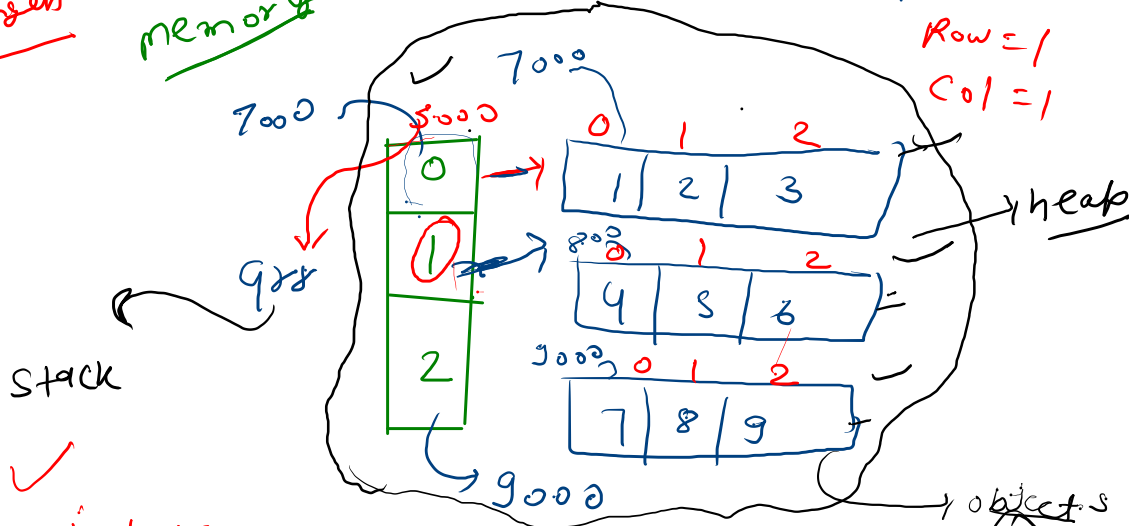
00Rs

qrr.length

memory

Row = 1  
Col = 1

3  
3 x 3  
qrr.length



Stack

objects  
④

✓ int rows = qrr.length ⇒ 3

✓ int cols = qrr[0].length ⇒

	4	
	6	
$R_1 \rightarrow$	2 3 8 7 0 4	
$R_2 \rightarrow$	<del>0 7 6 7 3 5</del>	X
$R_3 \rightarrow$	<u>0 0 8 1 0 8</u>	✓
$R_4 \rightarrow$	<del>9 1 9 5 3 0</del>	

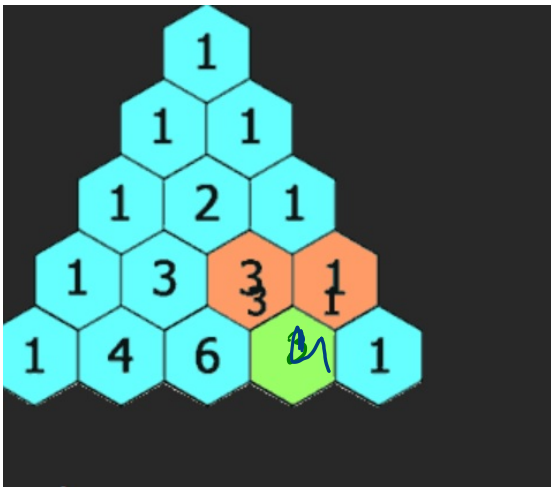
2 3 8 7 0 4  
0 0 8 1 0 8

Jagged

`int[][] arr = new int[rowSize][];`

pascal triangle

1 2 3  
4 5  
7 6 8 9



→ 1 col

→ 1 1

→ 1 2 1

→ 1 3 3 1

→ 1 4 6 4 1

rows = 3

cols = 3

✓

	0	1	2
0	(0,0)	(0,1)	(0,2)
1	(1,0)	(1,1)	(1,2)
2	(2,0)	(2,1)	(2,2)

0	1	2	3	4
5	6	7	8	9

↑  
arr[1]

$1 \leq 3$

for (int i=0; i < rows; i++) {  
    for (int j=0; j < cols; j++) {

        // ...  
    }

        i j = sc.nextInt();

        0 1 = "

        0, 2 = "

1, 0
1, 1
1, 2