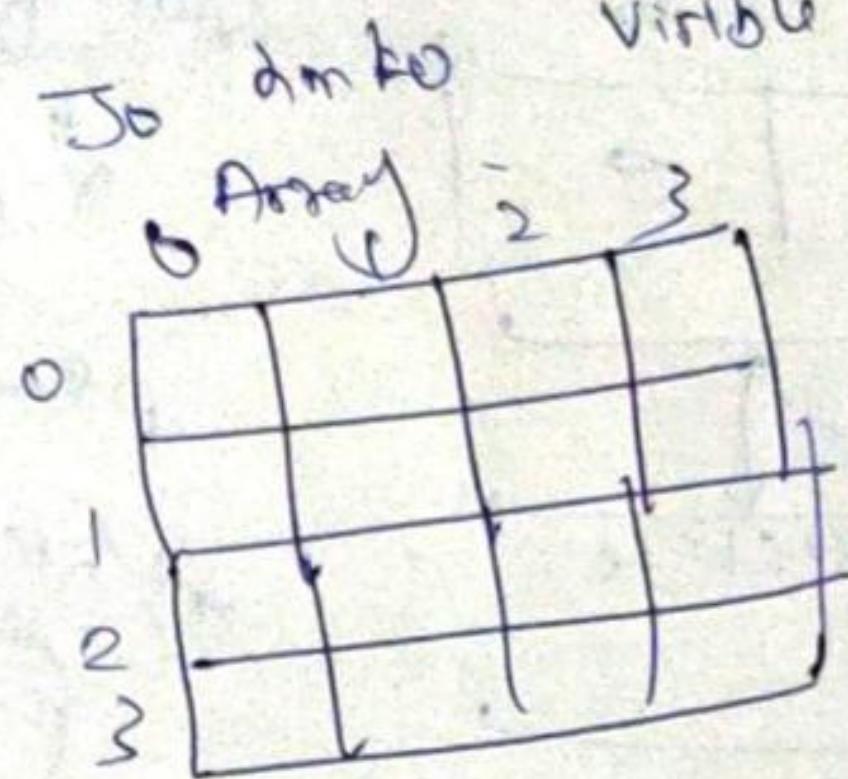


1st Jan | 2021

2D ARRAY

2D Array →

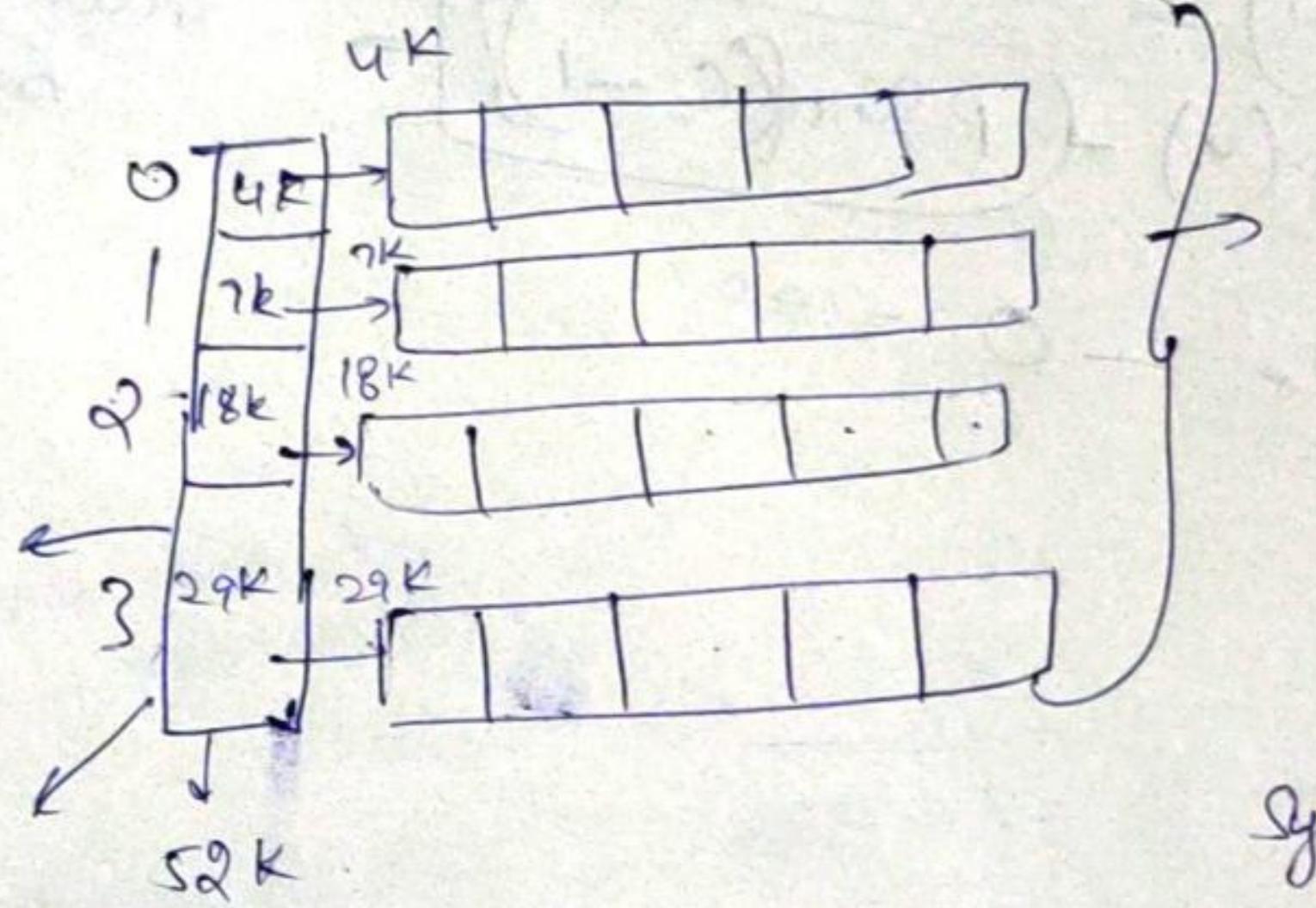


In Real Only 1D exist, work under 1 hor array hata h

29
18
58

address

arr.



arr[2]

syso(arr); → O/P → 52K

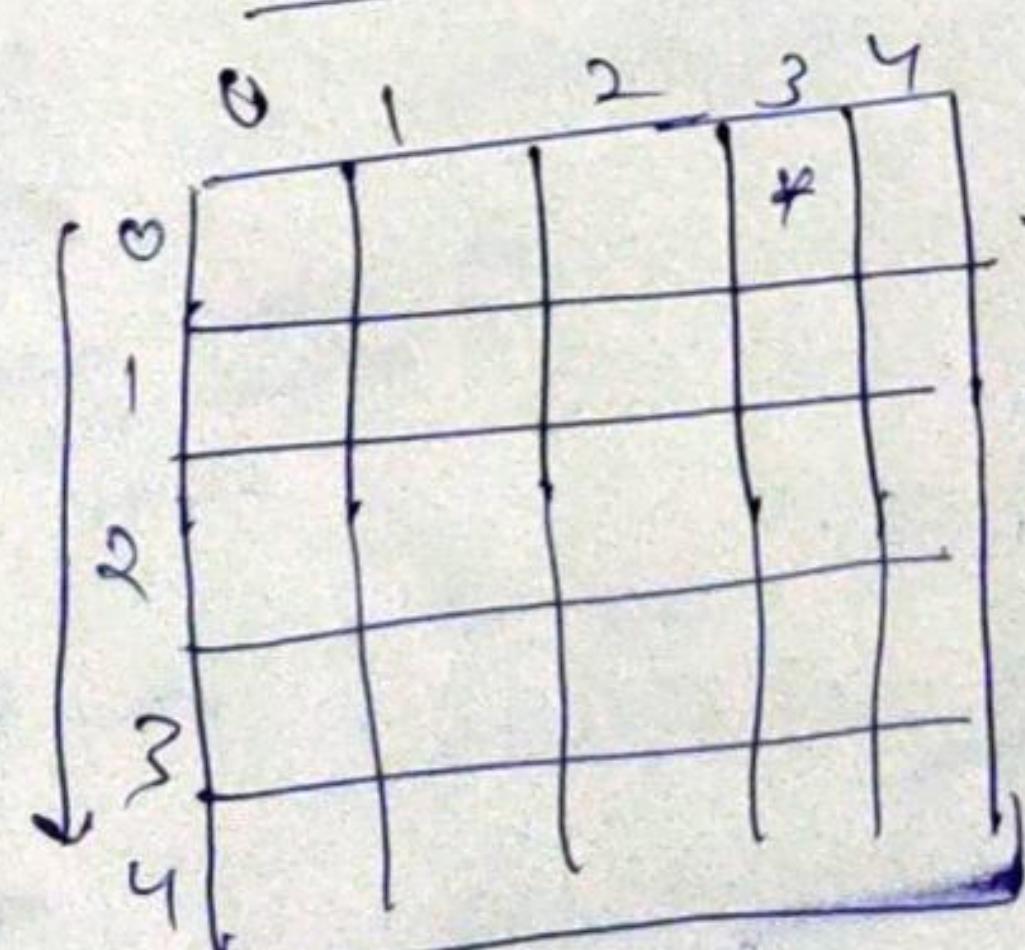
syso(arr[2])

O/P 18K address point

to jega.

So agr actual element point hoga.
Tike. Syso(arr[2][1])

Ex →



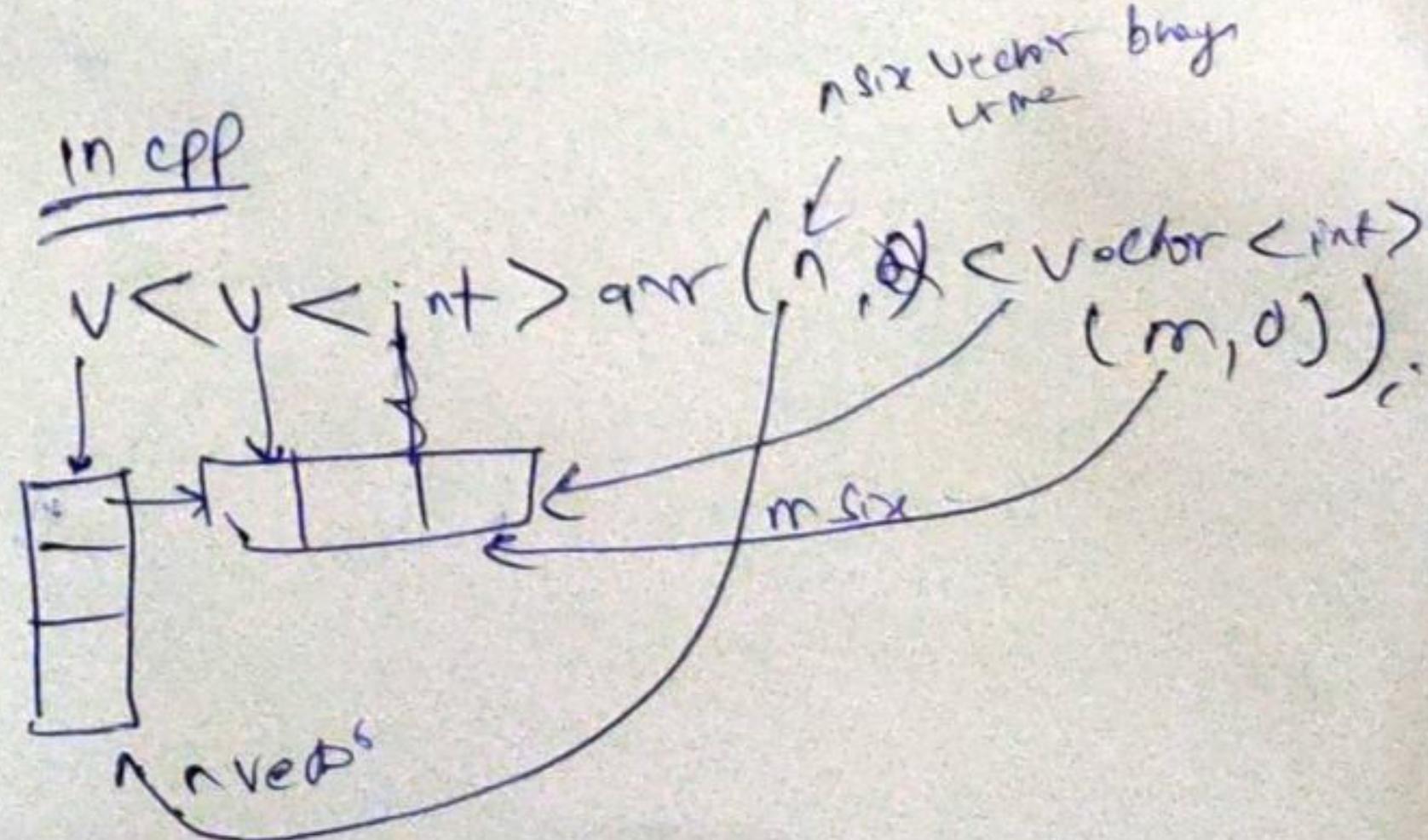
how to access element
arr[0][3]

hoga

how to declare 2D array in
Java.

int[][] arr = new int[n][n];

in CPP



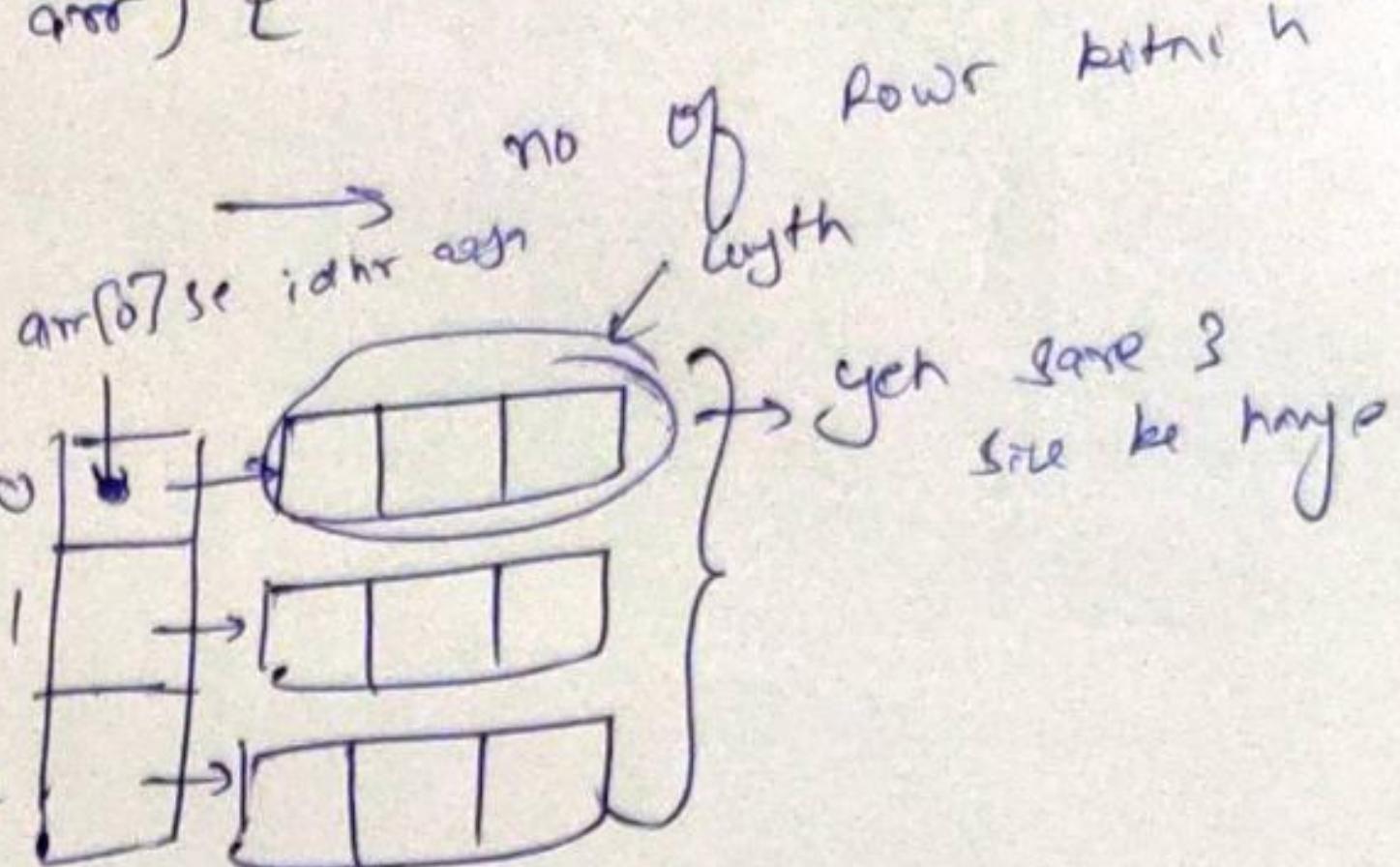
How to Take Input of 2D Array in Java.

O.P.S. void input (int[][] arr) {

int n = arr.length;
int m = arr[0].length;

why?

1st index par
Jo array k aur ke length
Chahiye; agr
uska nikal liya columns above same hole h



NO. of Row.

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

 no of columns → for (int j=0; j<m; j++) {

 arr[i][j] = scn.nextInt(); } }

$n \times m$
yeh sbhi type
for rotation h.

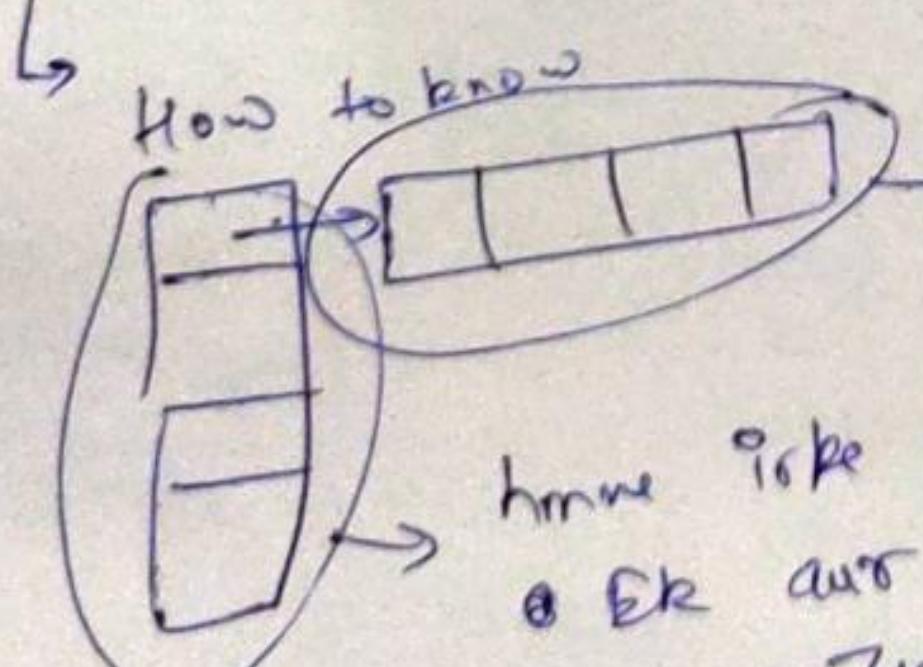
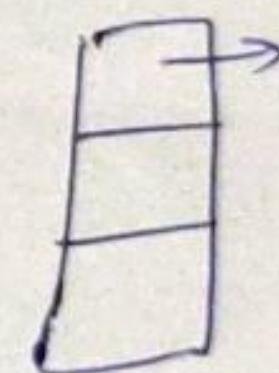
Display by For Each Loop → || Only for getting the values

for (int[] a : arr) {

means hm
urke and phleg
array chahiye.

for (int ebt: a) {

yeh a array jo bhyah
upsr 1, 10 kiske travel hole h.



(b)

ab hm iss array icke and r phleg h integer

hme iske upr loop kba yeh hai
a ek aur array ba means 1 no array h
Tin data type hogya.

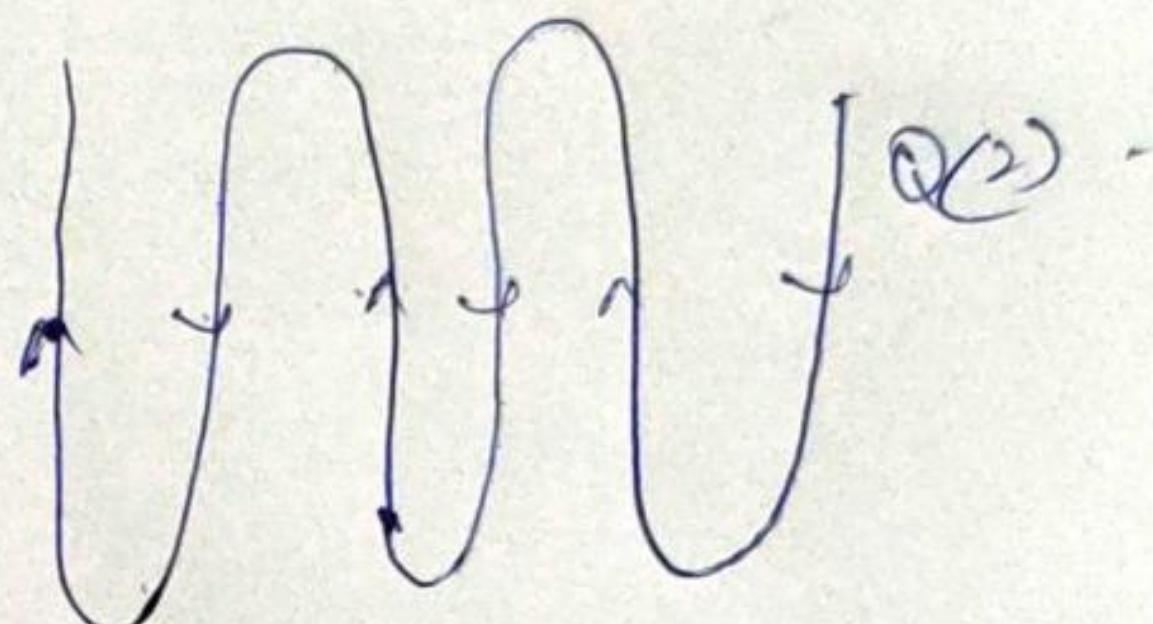
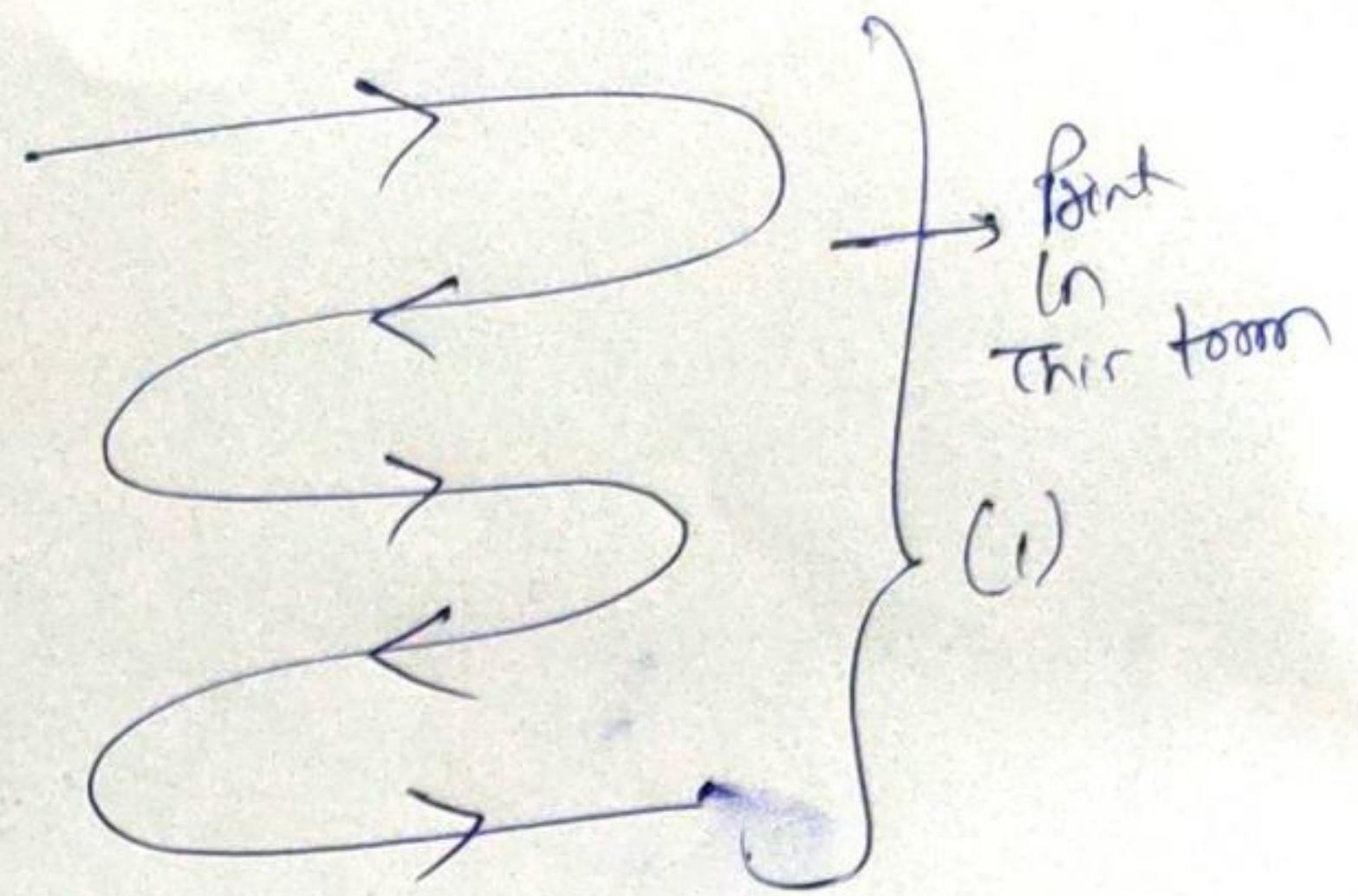
Q. Max

Q. Min

Q. Point spiral

States of Watermark 1

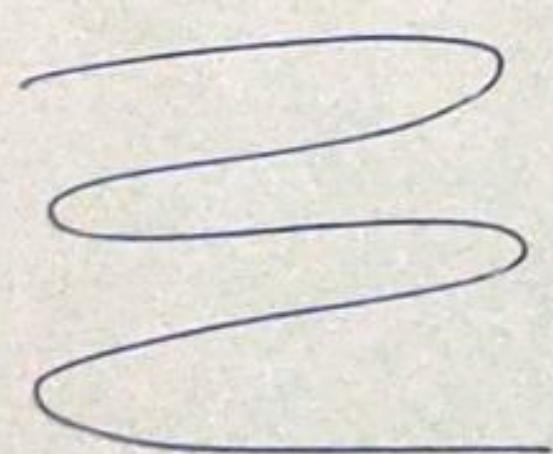
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
2	6	396	-3	-1
-2	7	1	0	137



(3)

Q

Q(1)



0	1	2	3	4
1	1	2	3	4
2	10	9	8	7
3	11	12	13	14
4	20	19	18	17
5	21	22	23	24
6				25

row - column
0 → 0 → 4
1 → 4 → 0
2 → 0 → 4
3 → 4 → 0
4 → 0 → 4

Note :-

Observe that 3 loop is dependent on 0, means To go over loop, be value 7 to left To Right

$$q_f (\% 2 == 0)$$

loop direction
→ zigzag move

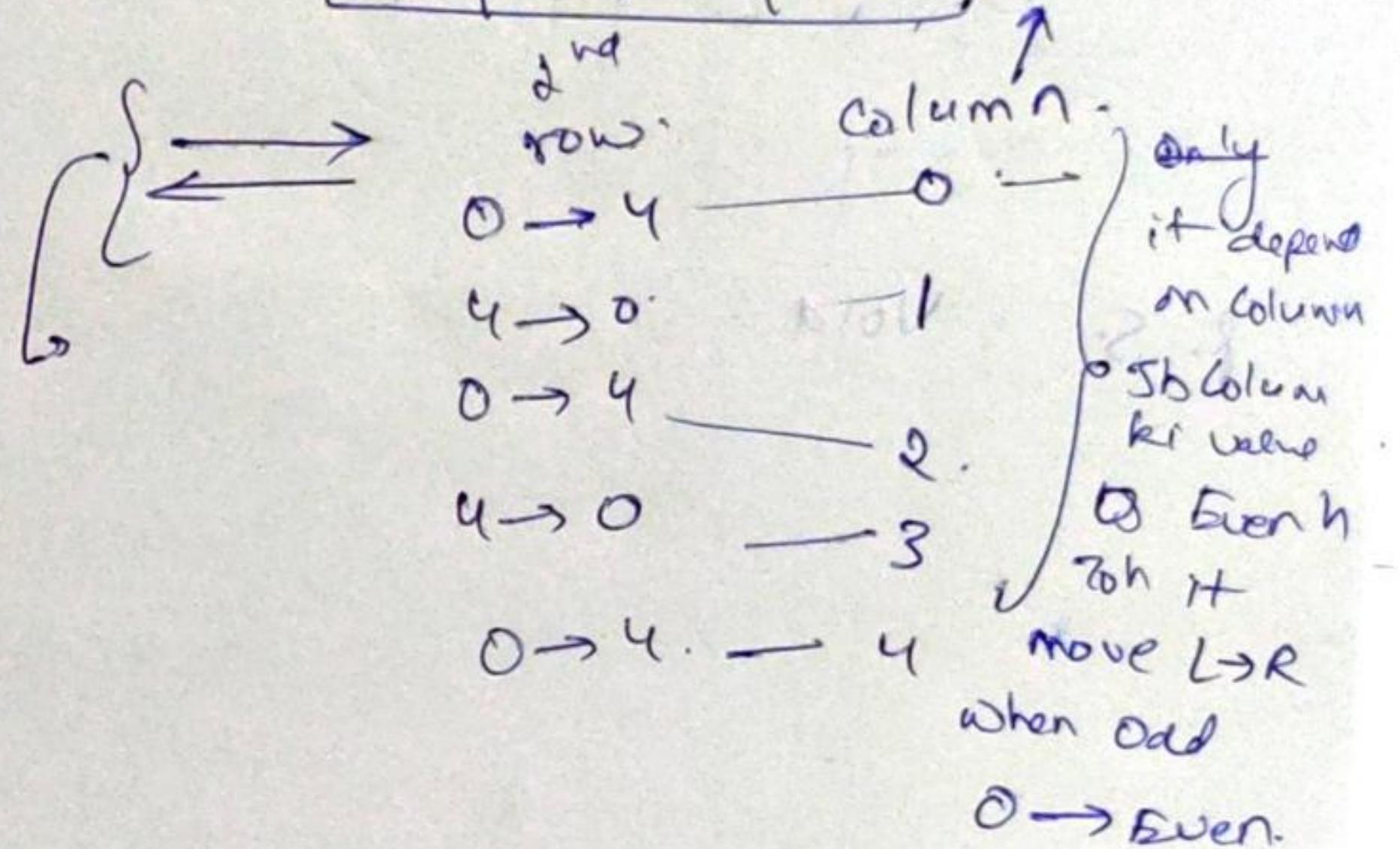
$$\frac{2}{0} \rightarrow \frac{2}{2} \rightarrow \frac{2}{0} \rightarrow L \rightarrow R$$

i → only for Row
j → " " columns

(2) are trashed up and down

```
for (int j=0; j<column; j++) {  
    if (j%2 == 0) {  
        for (int i=0; i<n; i++) {  
            System.out.print("*");  
        }  
        System.out.println();  
    }  
}
```

	0	1	2	3	4
0	1	2	3	4	5
1	10	9	8	7	6
2	11	12	13	14	15
3	20	19	18	17	16
4	21	22	23	24	25



Diagonal traversal -

Bare
Diagonal

point

gap 4

gap 3

gap 2

gap 1

gap 0

Bare
Diagonal

Do all Diagonal question with gap technique. Coz. ap no use bkt h.

fg ↪ gap → 2.

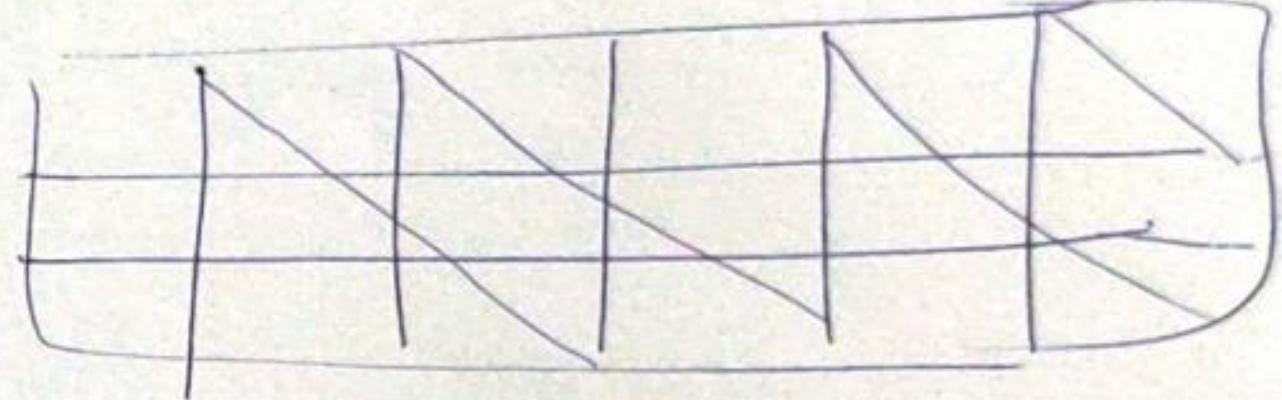
is always 0

(i) is start with gap

and Diagonal move move with the same speed.
Toh always move move h

Eg: $\text{gap}^0 \rightarrow (0,0)$ } i, j ab into fixed Rate so age move known
 $\text{gap}^1 \rightarrow (0,1)$
 $\text{gap}^2 \rightarrow (0,2)$
 $\text{gap}^3 \rightarrow (0,3)$
 $\text{gap}^4 \rightarrow (0,4)$

Time Complexity
 $\rightarrow \frac{n \times m}{2}$.



f. S. Void

Q. Rotate by 90° .

Q. Swap 6 Pairs

Q(3) Rotate by 90°

	1	2	3
0	11	12	13
1	21	22	23
2	31	32	33
3	41	42	43

90° rotate -

41	31	21	11
42	32	22	12
43	33	23	13
44	34	24	14

T.C. $(N \times N)$ always
Swapped

Sol Transpose length and \Rightarrow Row Column to reverse krdge
Transpose \rightarrow up triangular Area \hookrightarrow who swap krdge i_j to j_i
 \Rightarrow swap breg.

11	12	13	14
21	22	23	24
31	32	33	34
41	42	43	44

after swap \rightarrow

j_1	1	31	41
12	22	32	42
13	23	33	43
14	24	34	44

for full transpose
by 90° .

to swap the rows use 1 more while loop.

int $j_1 = 0$; $j_2 = m-1$;
while ($j_1 < j_2$) {

or row change with regt till
 j_0 to n .

for (int $i = 0$; $i < n$; $i++$) {

int $temp = arr[i][j_1];$ swap

$arr[j_1] = arr[i][j_2]$

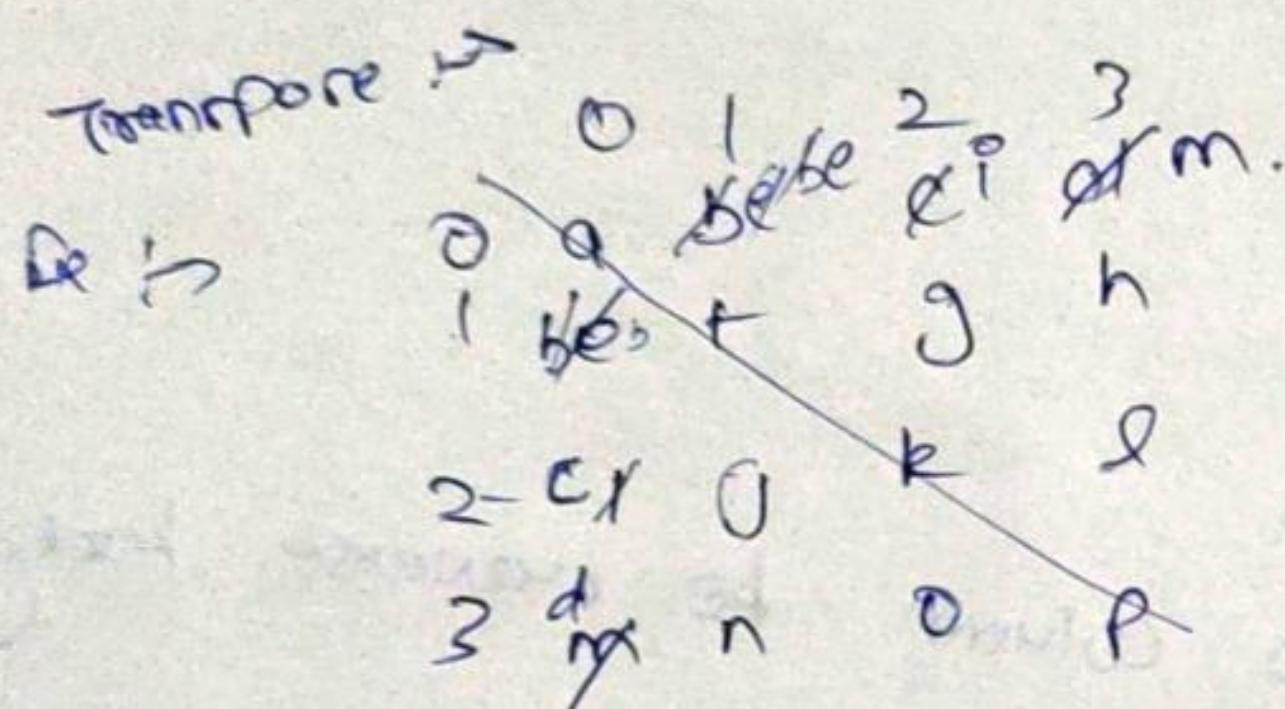
$arr[i][j_2] = temp;$ } } }

$$\text{Sol} \quad \text{Transpose} \rightarrow \begin{bmatrix} 6 & 4 & 3 & 2 & 4 \\ 1 & 9 & 8 \end{bmatrix}^T = \begin{bmatrix} 6 & 1 \\ 4 & -9 \\ 2 & 4 & 8 \end{bmatrix}$$

$$\text{Transpose} \cdot \begin{array}{l} \text{edge} : \\ e_{ij} \end{array} \downarrow \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow \text{OP} \rightarrow \begin{bmatrix} 1 & 4 & ? \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

$$O_1, i=0 \\ O_1, 1 = 2, 4$$

swap (e_{ij}, e_{ji}) , and traverse in any 1 triangle.



$$\begin{array}{l} O-O \xrightarrow{\text{swap}} O,O \\ O-1 \rightarrow 1,O \\ O-2 \xrightarrow{\text{swap}} 2,O \\ O-3 \rightarrow 3,O \end{array}$$

$O-O \rightarrow$ agar isko dabegi
agar swap krega toh
ar it is agar

So, JB swap krega toh
 agar upper triangle me move krega.
 agar pure me swap krega toh hogya hain hi,
 So agar jo hai vo 0' se start krdge, and agar
 0 ke kya toh pura triangle move hogya.

Transpose loop

```
for (int i=0; i<arr.length; i++) {
    for (int j=0; j<arr[i].length; j++) {
        int temp = arr[i][j];
        { swap
            arr[i][j] = arr[j][i];
            arr[j][i] = temp;
        }
    }
}
```

i	0	1	2
0	a	b	c
1	d	e	f
2	g	h	i

$$\begin{array}{l} \text{temp} = \text{arr}(0,1) \\ (0,1) = (0,0) \\ (0,1) = (1,0) \\ (0,0) = (0,0) \\ (1,0) = (0,1) \end{array}$$

$$\begin{array}{l} (0,0) \xrightarrow{\text{swap}} 1-0 \\ 0-1 \xrightarrow{\text{swap}} 2-0 \\ 0-2 \xrightarrow{\text{swap}} 3-0 \\ 0-3 \xrightarrow{\text{swap}} 1-0 \\ 1-0 \rightarrow 0-1 \\ 1-1 \rightarrow 0-0 \end{array}$$

}

If we start $j=0$ ←
 Transpose
 $(1-0) - (0-1)$
 Same versa logic

0	1	2	
0	a	b d	c g
1	b d	c	f h
2	c g	f	i

Transpose
 \rightarrow
 O/P →

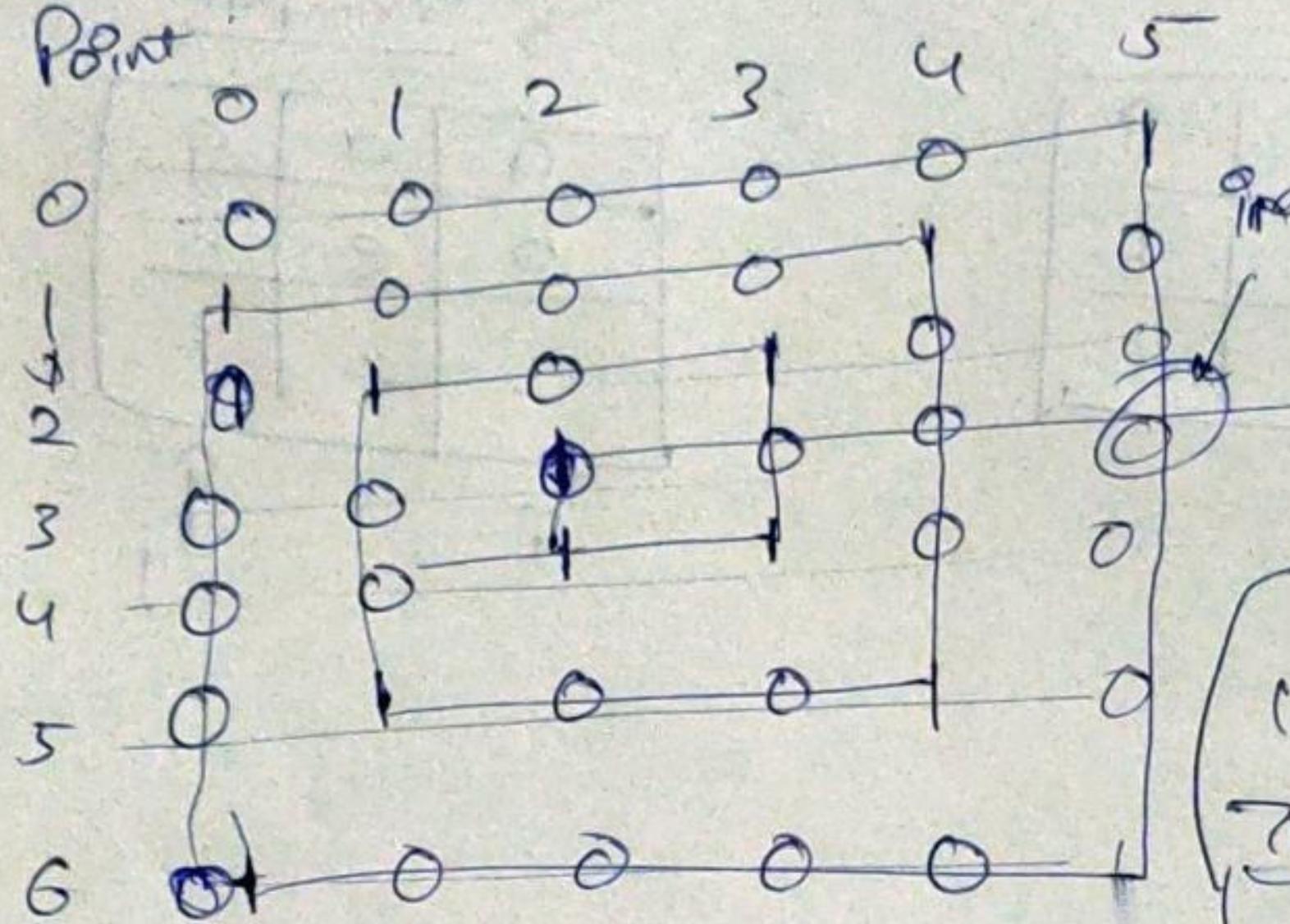
g	a	9
h	e	b
9	f	c

Transpose

a	a	g
b	e	h
c	f	i

2nd Jan / 2D ARRAY

Q1 Start Point



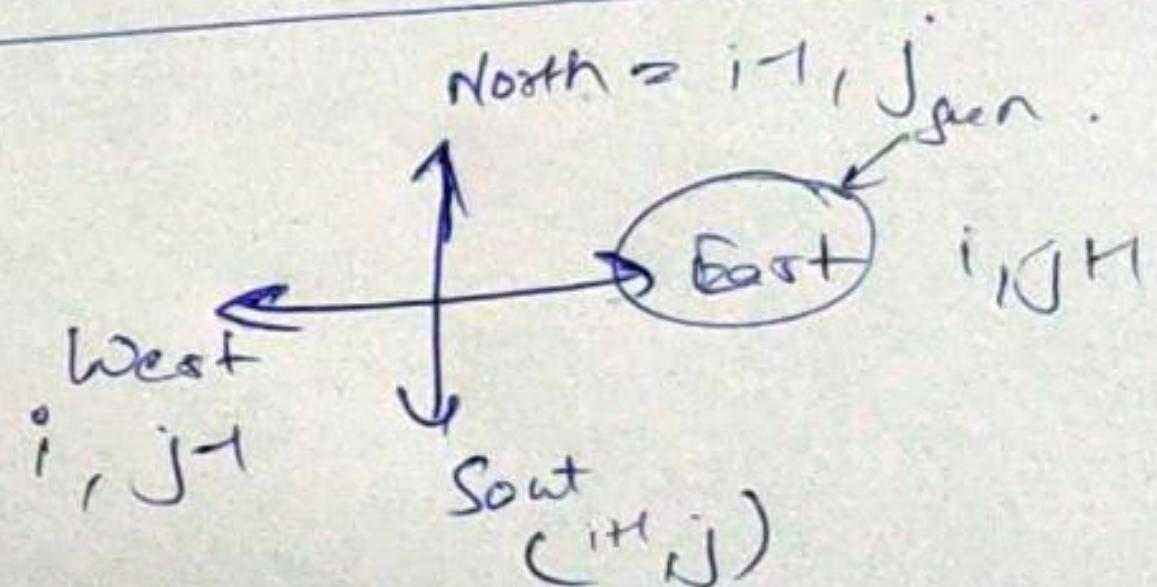
Jitter 1mleg
Index or se Right turn
line.

or take straight
move long h

Wherever, you will get the
I take, Right turn from
there, and when it short (P)
means we have
straight

or Vo index blank & Jitter so
matrix se bhar Nikol the h.

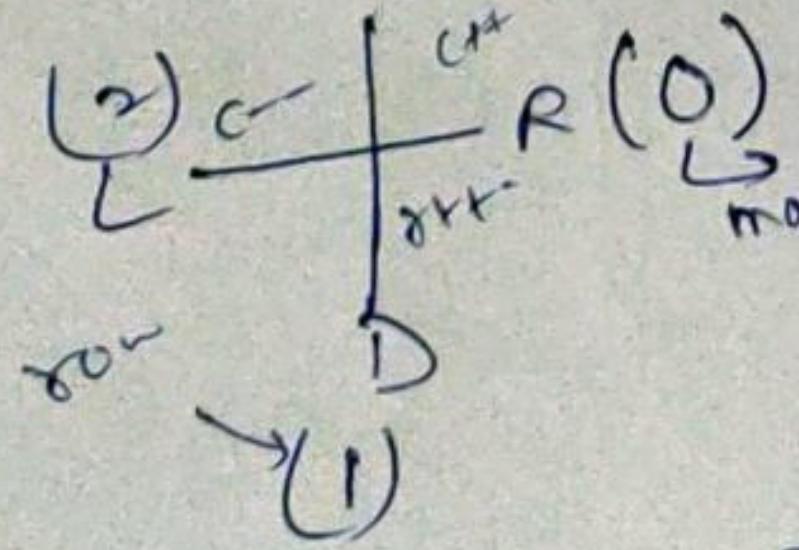
we have specify
start given the index where from which
exited



Q1 Total Direction present in this Ques.

L,D,T,B

$i, j+1$



Right when direction val is 0.

be General
u/o R/o R/o
o - o - o - o

$$dir = \frac{dir + arr[i][j], \% 4}{4}$$

Downward

$$d = XXB4/56$$

C=5

$$1+1=2$$

left

$$dir = ?$$

$$r = 6$$

$$c = 6 \times 4B71$$

6+1

52

is yet done
range - CD
actual value
miles

$$\gamma = 0$$

$$C = 1X85$$

0+3 → Right M
gye

$$0,5 \rightarrow 0+1=1$$

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	1	0	0	0	1	0
2	0	1	0	1	0	0
3	0	0	1	0	0	0
4	0	0	1	1	0	0
5	0	4	0	0	1	0
6	-1	0	0	0	0	1

Exit point

So here we have 4 direction
Turn Right where we get 1

Total Direct $\rightarrow 4$, So use Direction loop.

$$\text{row} = 0, \text{col} = 5$$

$$row = 0, col = 5$$

$$\text{direct} = \text{direct} + arr[i][j]$$

$$0 + 1 = 1$$

$$0 + 1 = 1$$

$$0 + 1 = 1$$

arr[i][j]

11%4

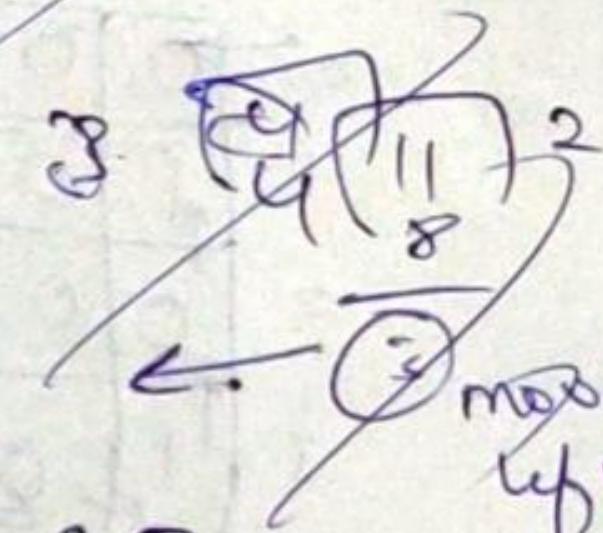
$$0 + 1 = R_1$$

u(3)

L(2) \leftarrow

R(0)

move right
when value 950.



$$\text{direct} = \text{direct} + arr[i][j]$$

$$0 + 5 = 5$$

$$0 + 5 = 5$$

$$0 + 5 = 5$$

$$0 + 5 = 5$$

move downward

$$1 + 0 = 1 \downarrow$$

$$r = 1 \times 6 \Rightarrow 6$$

move left

$$r = 6$$

$$c = 5$$

$$r = 1$$

$$c = 4$$

$$0 + 0 = 0$$

$$\text{direct} = 3$$

$$4 \sqrt{4} \Rightarrow 0 \rightarrow \text{direct}^n$$

$$\text{direct}^n = 0 + 1 = 1$$

$$\text{direct}^n = 1 + 1 = 2$$

$$1 + 1 = 2$$

$$0 + 1 = 4 \% 4 = 0$$

Do this with the loop.

P.S. void exitPoint(int arr[4][4]) {
int r=0, c=0, dir=0;
while (true) {
 infinite loop ki break h
 arr are here note P+R+L+U h
 me break h.

$$arr = (dir + arr[i][j]) \% d;$$

Note: Jo % hot h
vo koi bhi
cheez agar kya
de klar askar kya
me break h.

if (dir == 0) { → mean time right
 direct move hoga.

c++;
if (c == arr[0].length) { ← breakpoint
 break ki condition
 kya logi, mean matrix ki
 range me hi R हो.
 arr[0] → 1stumbo
 row point
 arr[c-1]; → 1st row
 break;

}

~~first time~~

	0	1	2	3
0	0	8	9	0
1	1	0	0	11
2	0	0	0	1
3	1	4	2	0

0+1 = 1 → down move
dir → arr[i][j]

$$direct = 1 + 1 = 2$$

$$direct = 2 + 1 = 3 \uparrow$$

$$4 \% 4 = 0$$

$$direct \Rightarrow 1$$

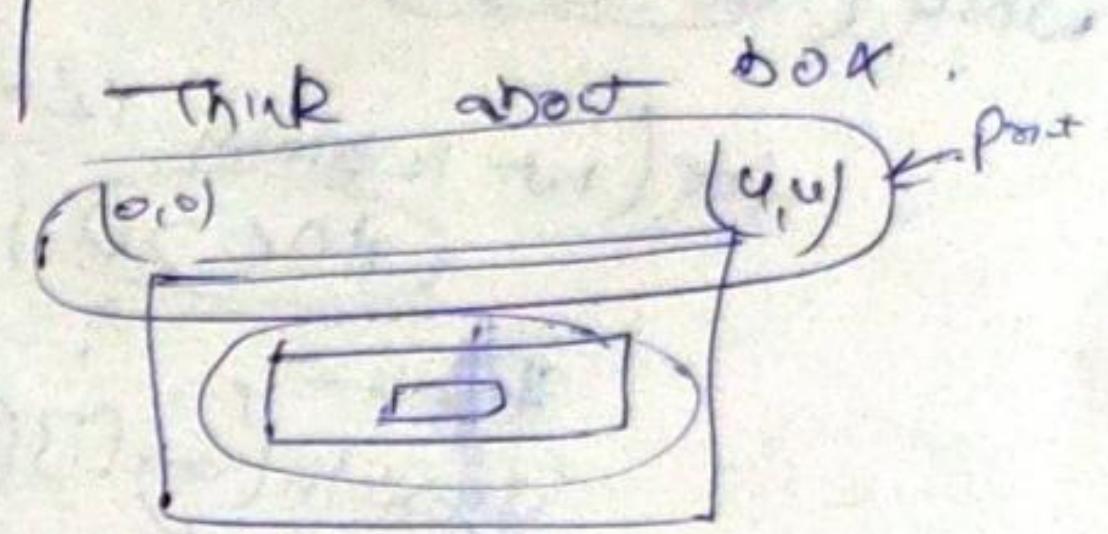
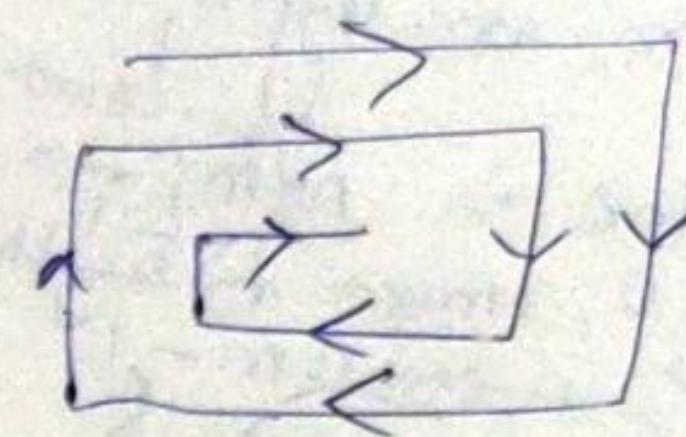
0	0	0
0	0	1
0	0	2 $\Rightarrow 1$
1	1	2
1	2	2
1	3	2.
2	3	1
2	3	0.
3	2	0
3	1	0
0	1	1
0	1	2
0	1	3
1	3	3
2	2	2
2	2	1
2	2	0
2	2	exit

Q

Spiral Display

11	12	13	14	15
21	22	23	24	25
31	32	(33)	34	35
41	42	43	44	45
51	52	53	54	55

move in spiral form



① 4 loop + 1 while required
loop with sb
(3b tab yet sb
executed Nth hole 2b tab)

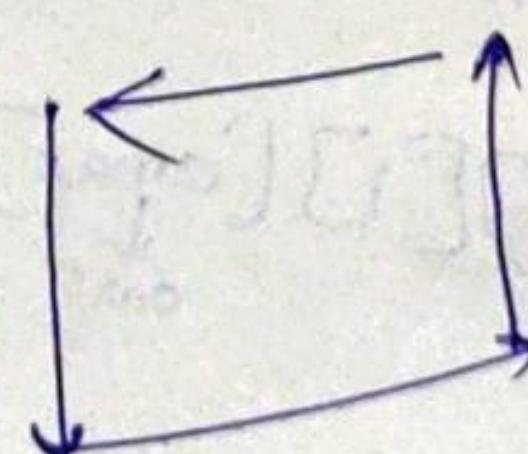
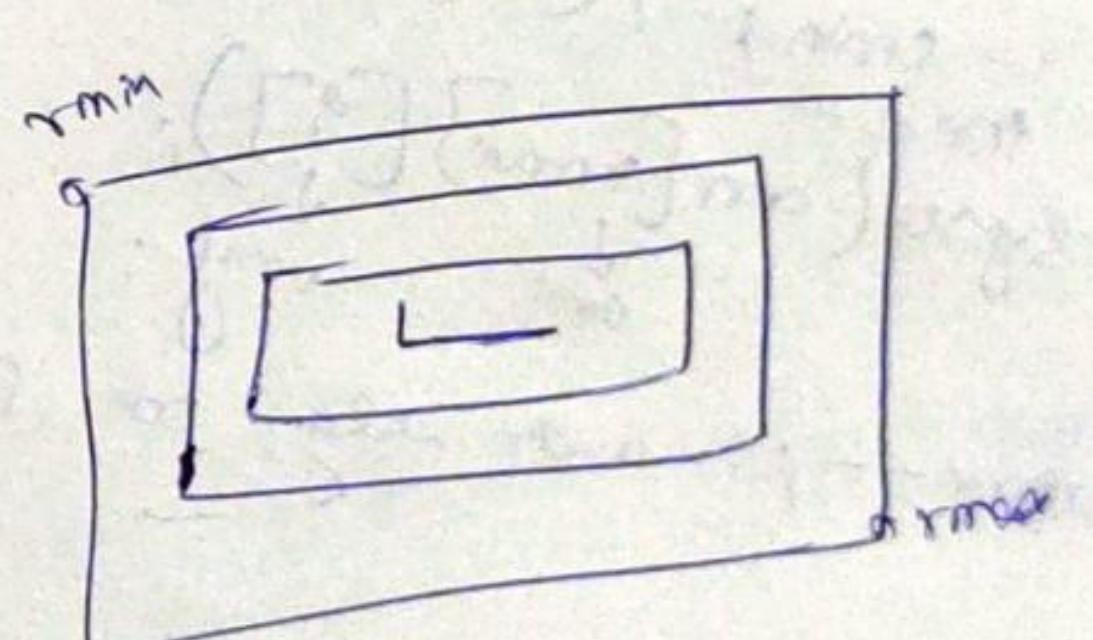
rmin
rmax
cmin
cmax

① $[r_{\min}] \rightarrow [c_{\min} \rightarrow c_{\max}]$
Constant page
move
to move in next row $\rightarrow r_{\min}++;$

② $[c_{\max}] \rightarrow [r_{\max} - r_{\min}]$
const
move
in next column $= c_{\max} -;$

③ $[r_{\max}] \rightarrow [c_{\max} - c_{\min}]$
 $r_{\max} --;$

④ $[c_{\min}] \rightarrow [r_{\max} - r_{\min}]$
left to down
~~cross~~; $c_{\min}++;$



```

public static void spiral(int arr[][]) {
    int m = arr[0].length;
    int n = arr.length; m = arr[0].length;
    int tne = n * m; count
    int rmin = 0, rmax = m - 1;
    cmin = 0 + count - 1;
    while (tne > 0) {
        for (int i = rmin; i <= rmax && tne > 0; i++) {
            tne--;
            System.out.println(arr[i][cmin]);
            if (rmin == rmax) break;
            cmin++;
        }
        rmax--;
        if (rmin == rmax) break;
        for (int i = rmax; i >= rmin && tne > 0; i--) {
            tne--;
            System.out.println(arr[i][cmax]);
            if (rmin == rmax) break;
            cmax--;
        }
        rmin++;
        if (rmin == rmax) break;
        for (int i = cmax; i >= cmin && tne > 0; i--) {
            tne--;
            System.out.println(arr[rmin][i]);
            if (rmin == rmax) break;
            rmin--;
        }
        cmin++;
    }
}

```

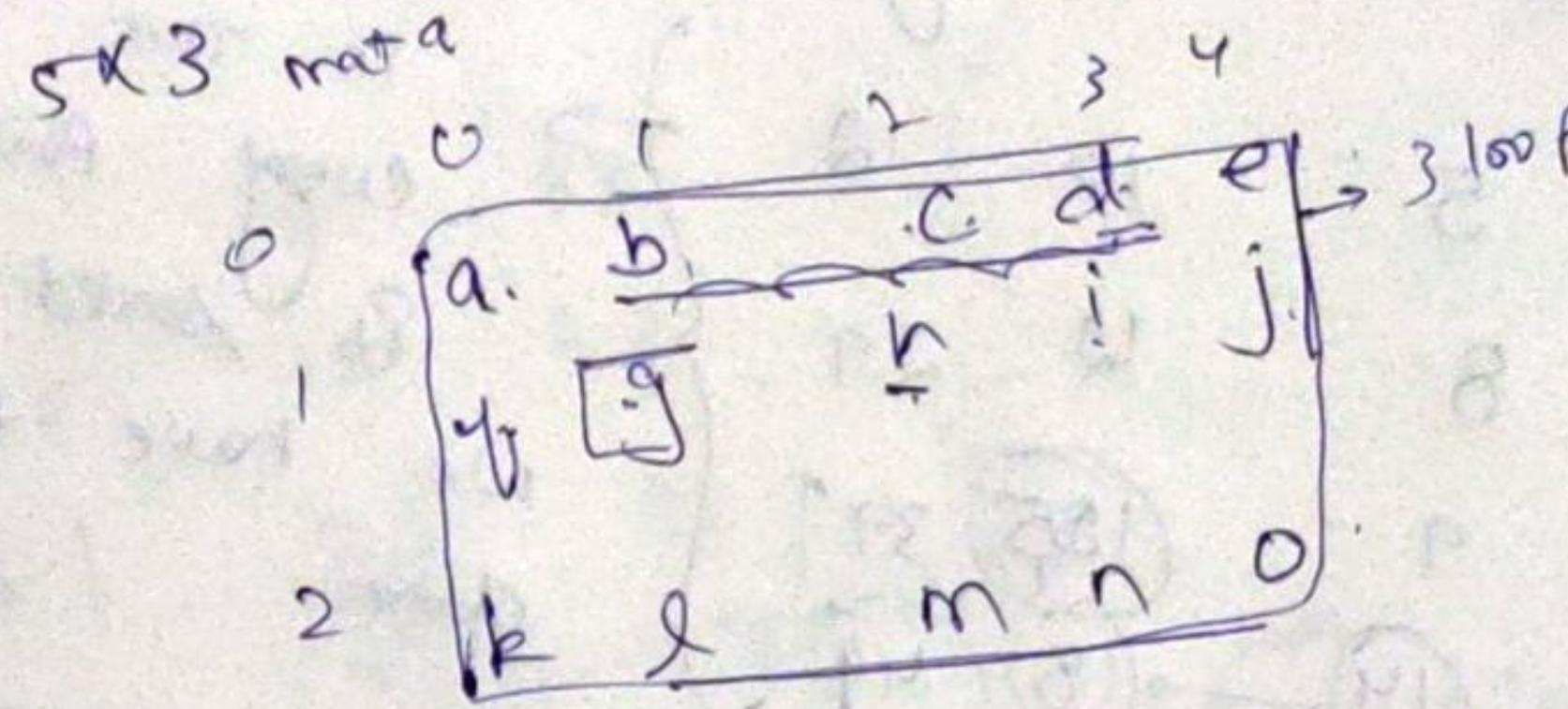
| Count variable
 or i - kote size

Column Constant
 Renge and we move
 from Row to Park

Why tne > 0?
 G2. 5x3 matrix me box jo
 r & l bar point hoge h
 so make sure point hote h
 i bar point hoge.

Compress
 r & l bar point hoge
 cmin & cmax
 and r & l bar point hoge.

Why the check?



$$r_{\min} = 1 \quad c_{\min} = 1(2)$$
$$r_{\max} = 2 \quad c_{\max} = 3 \cdot 2$$

O.P.

tre \rightarrow f3t2 f3

af k l m n o. je b c d.

age the wala check na yake toh. g point ho jaega.

g h i - h \rightarrow wrong ans.

so kar element 1 hour visit ho aitna;

Time Complexity \rightarrow Nibaleye her element kitni bar point hue h.

Ex \rightarrow 12 35

1	2	3	4
12	13	14	15
11	16	15	6
10	9	8	7

Q4) Search in Sorted 2D Array

			10	12	}
	4 → 5		12	29	
	7 8				
	8 9	15	39 ↑		
13	14	18	49		

In this Matrix
every Row, Column
is sorted.

We have to find the
data.

Approach : We have to start
probability to get 15
go down if not
so we will go right
start from anywhere, for ex: & what
will be, we cannot decide
if left or right
start from there.

(i) 49 As it is smaller than 15
choose start, so we cannot decide and hence we cannot
start from there.

(ii) Eg 13 be start, 10 find
choose range up, so
14 be right side in up.
and got the data.

```
while (i ≥ 0 & j ≤ m) {
    if (arr[i][j] > data) {
        i++;
    } else if (arr[i][j] < data) {
        j--;
    }
}
```

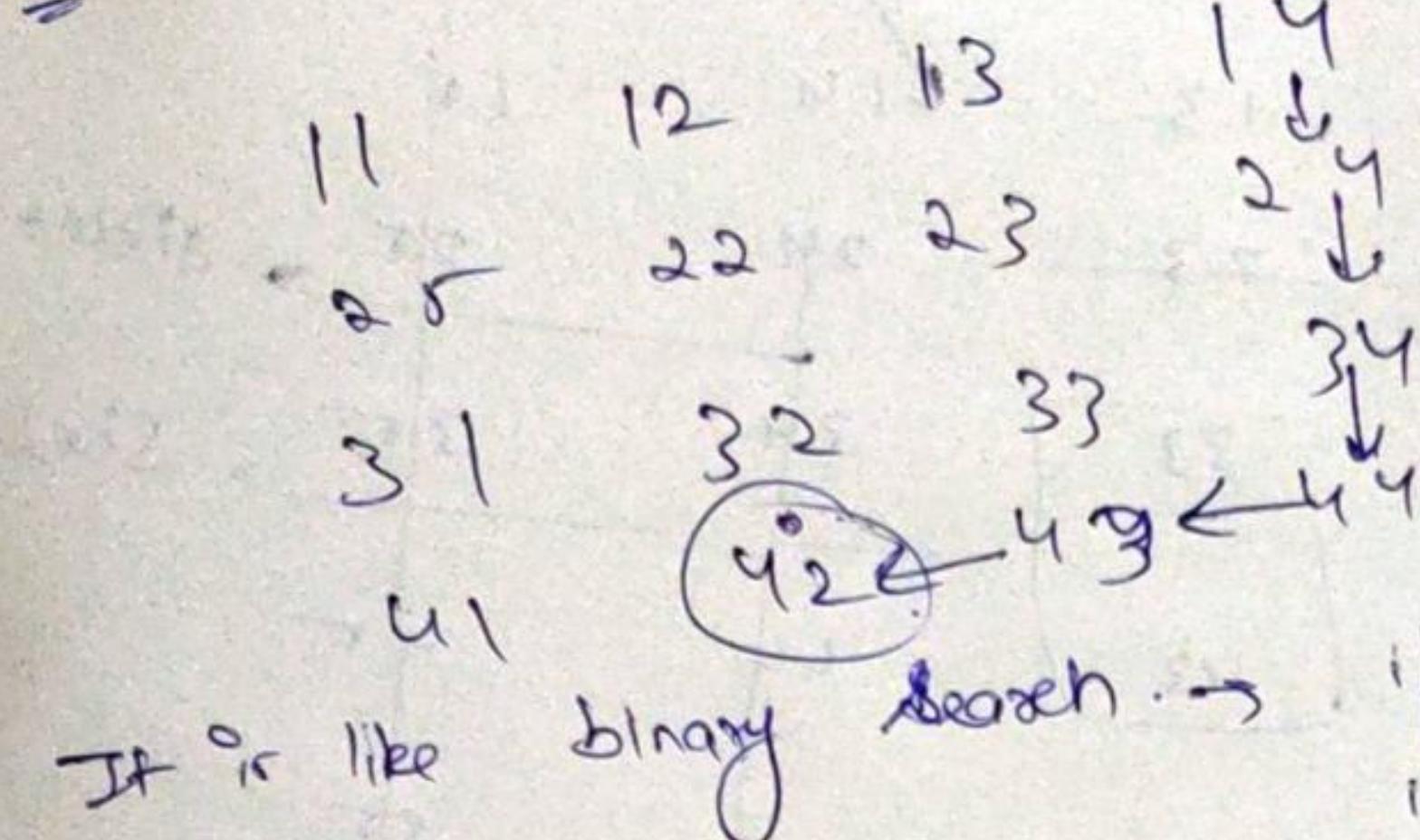
} else { → means data milga h
return print (i, j); } return;

```
3.
3.
print ("Not found");
```

0	1	2	3
4	5	10	12
7	8	12	29
8	9	15	39
13	14	18	49

Time Complexity
 $n \times m \rightarrow$ agor right move by
 take n elements or to
 find one element
 move up to m ,
 At m .

D. Sol Search in a Sorted 2D Array



data = 42

Compare 14 with 42.
bigger value are downward

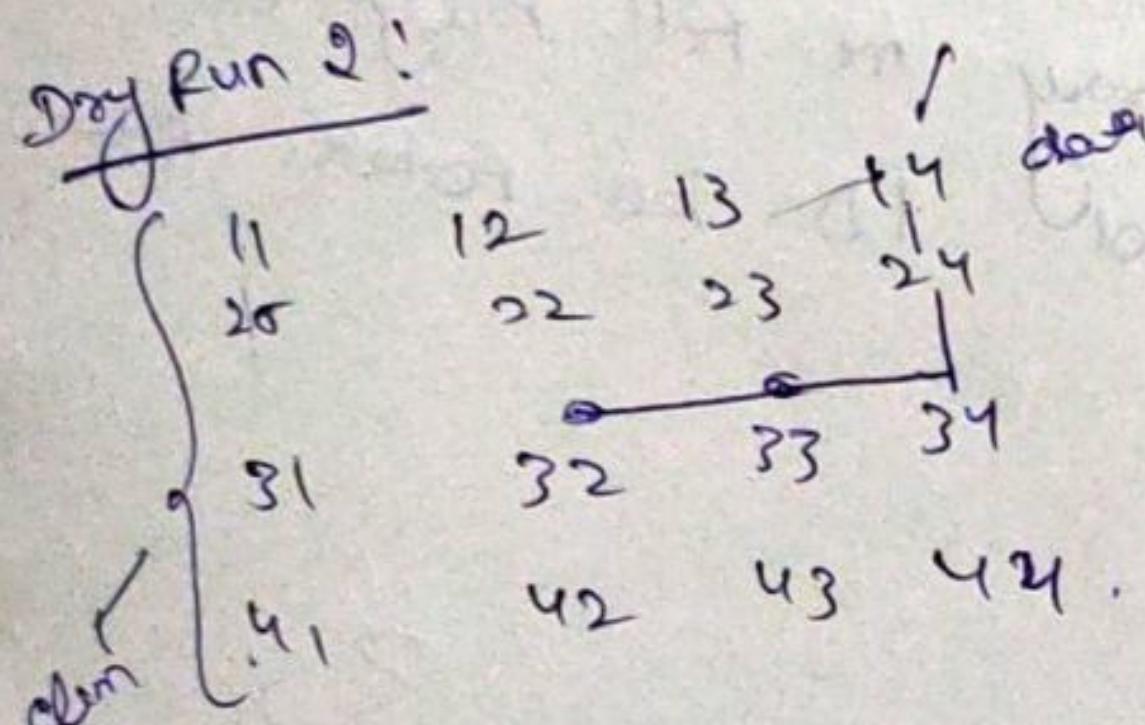
If it's like binary search →

```

if (x > elem)
if (x < elem)
else got data(x)
    
```

move downward
left

Dry Run 2:



data = 32

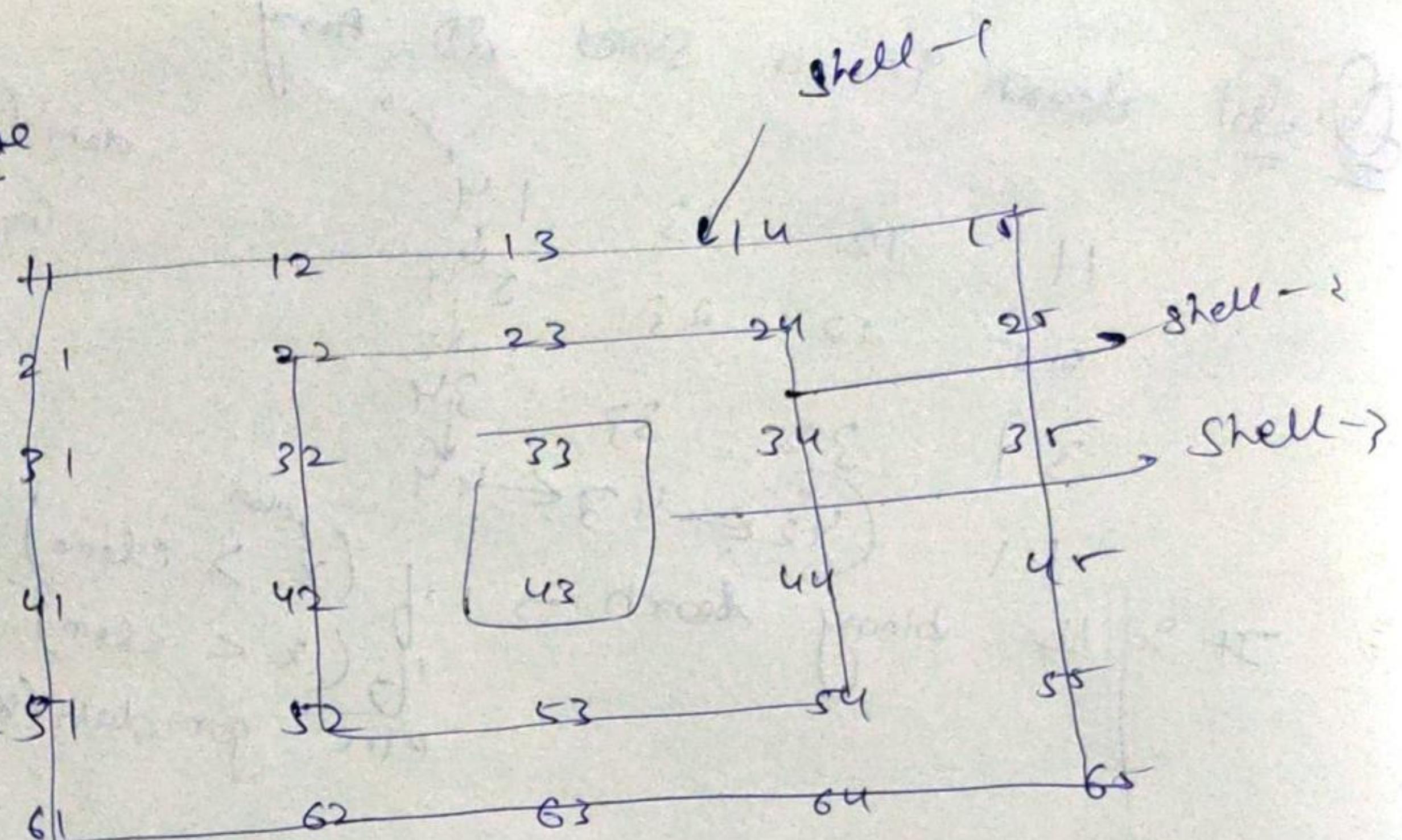
```

int i = 0;
int j = arr[0].length - 1;
while (i < arr.length & j >= 0) {
    if (x == arr[i][j]) {
        cout << "Element found";
        cout << endl;
        return;
    } else if (x < arr[i][j]) {
        j = j - 1; // x is in column m
    } else {
        i++;
    }
}
cout << "Not found";
    
```

Top Right Corner So Start here

Q. Shell Rotate

$m = 1$
ago we no ton
Anti clockwise
move merge 1 so
and ago ~~clockwise~~
move clockwise 51



sol 1
To BN shell n usko 1D array me fill bridge or
usko rotate kro
wps shell me fill bridge
1D to rotate kro

1. shell → fill 1D

2. 1D Rotate

3. Fill shell with 1D

33

public static int []

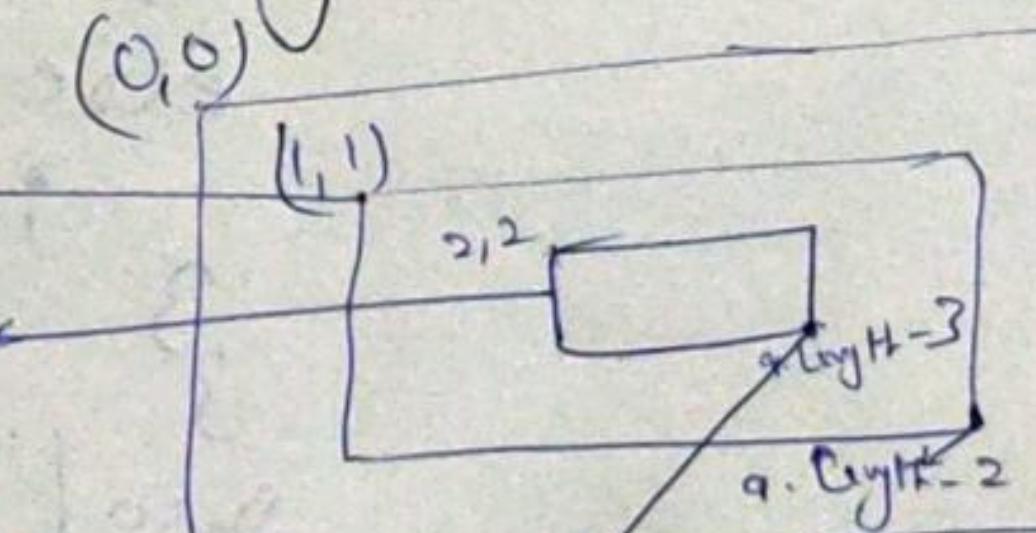
Fill 1D from (int [] arr, int s) {

To shell h
ere hme
apna 1D
fill kro return
merge

Spiral Jere hi loga

extra lga brach h. array?

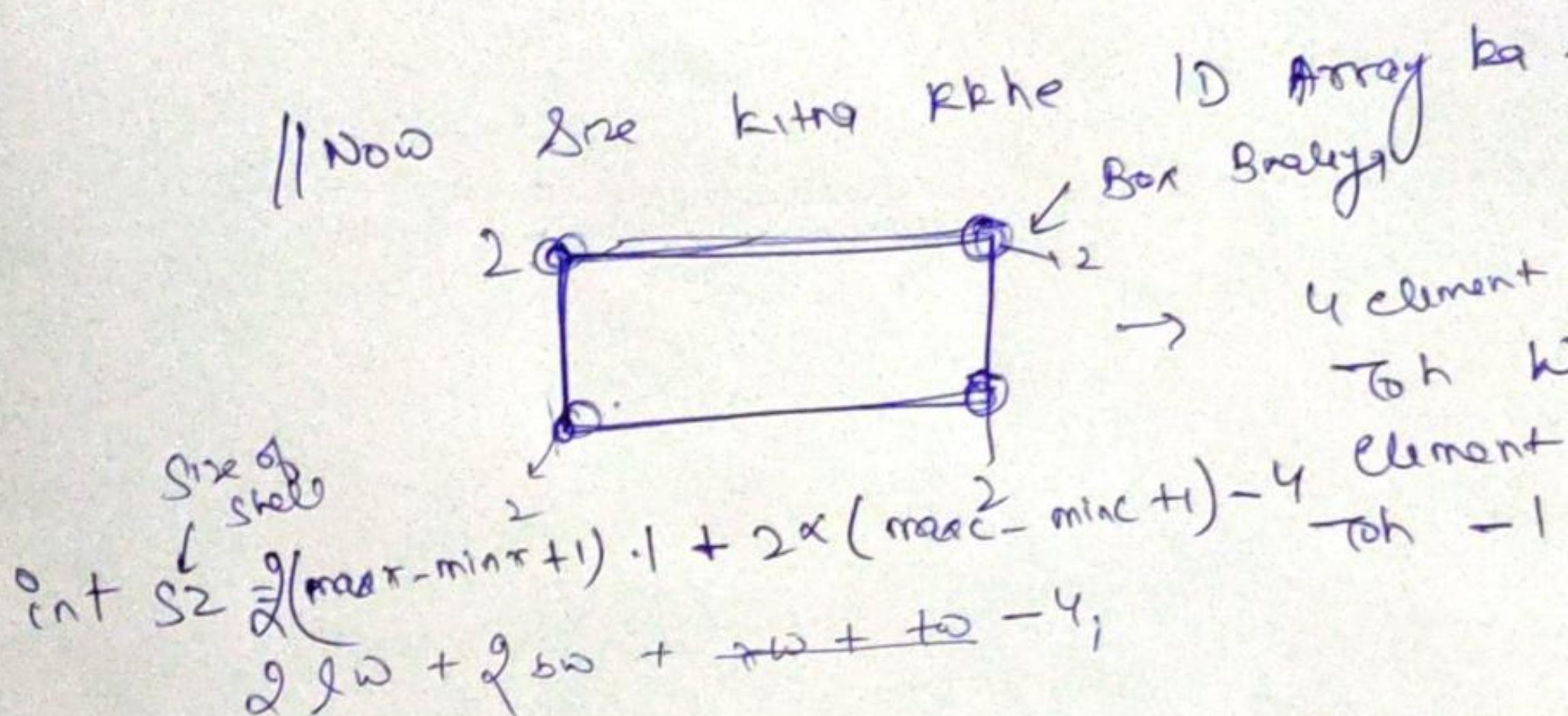
(0,0)
(top left, shell-1)
shell-3



arr. by th-1 &
arr[0]. key th-1 C

arr. length - shell
arr[0]. length - shell

int minr = 8-1, minc = 8-1, maxr = arr.length - shell, maxarr[0].length - shell;



~~$S2 = 2\maxr - 2\minr + 2 + 2\maxc - 2\minc$~~

\rightarrow

~~$S2 = 2\maxr - 2\minr + 2\maxc - 2\minc$~~

Take 2 corner

~~$2 \times [\maxr - \minr + \maxc - \minc]$~~

Verify

22	23	24	25
32	33	34	35
42	43	44	35
52	53	44	55

int [] oneD = new int [S2];

int index = 0; to store

//if & else

→ spiral loop

oneD[index] = arr[i][j];
index++;

3.

4 element 2 bear same h
Toh we want ki har
element 1 bear aye
Toh -1 berge har ek kg

idhar se idhar tak
kitne bade h?
 $5-1=4$. but hai
5.
So High - low + 1
 $5-8+1 = 5$

1 So hm agr dekhe
up & down me kitne bade h
go calculate by maxr -
 $\minr + 1$ {Include krega
(top or bottom)
to bw}

Total elements = 12
 $2 \times (4-1) +$
 $4-1)$
 $2 \times (3+3)$
 $2 \times 6 = 12$.