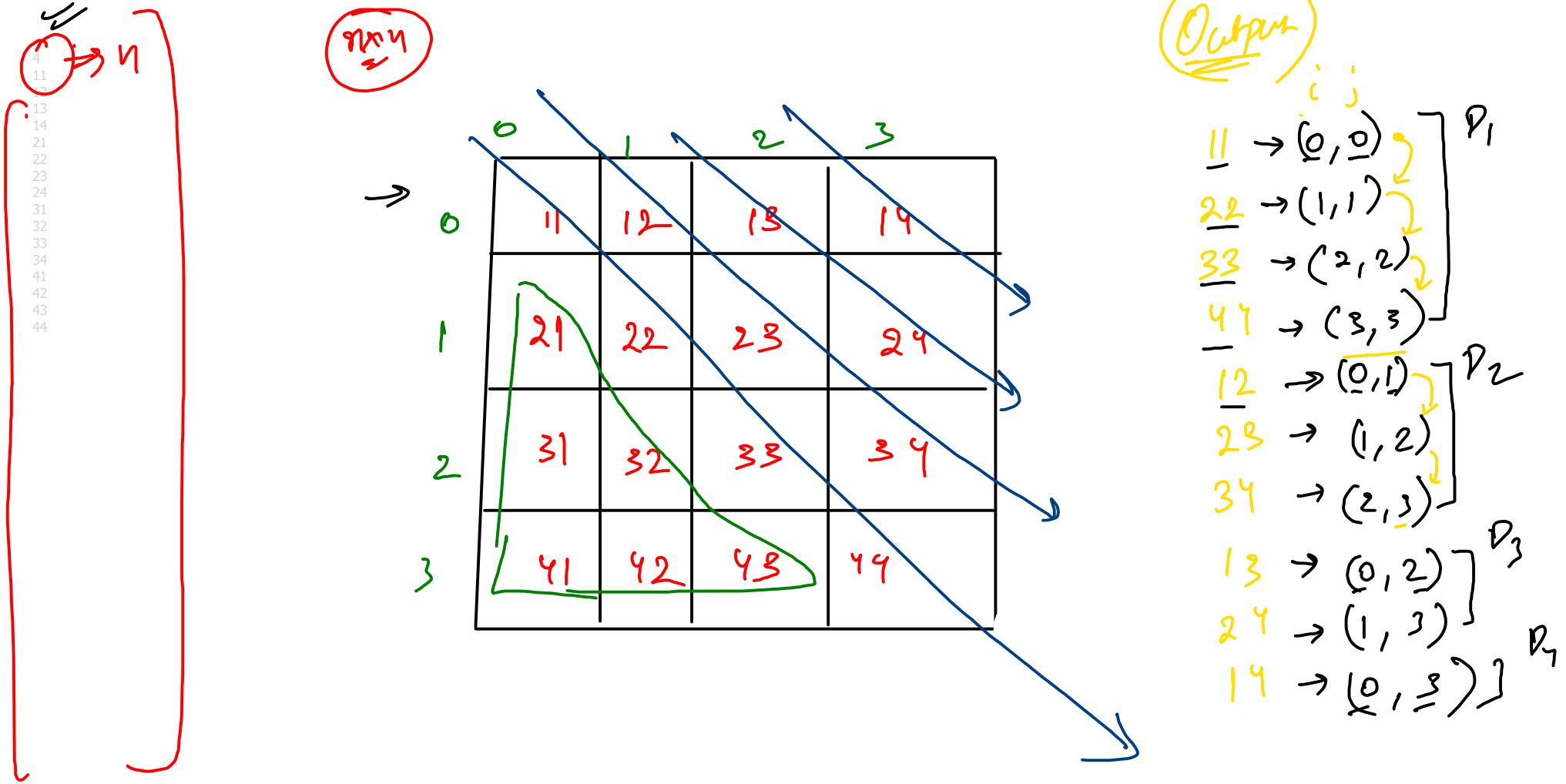


Output

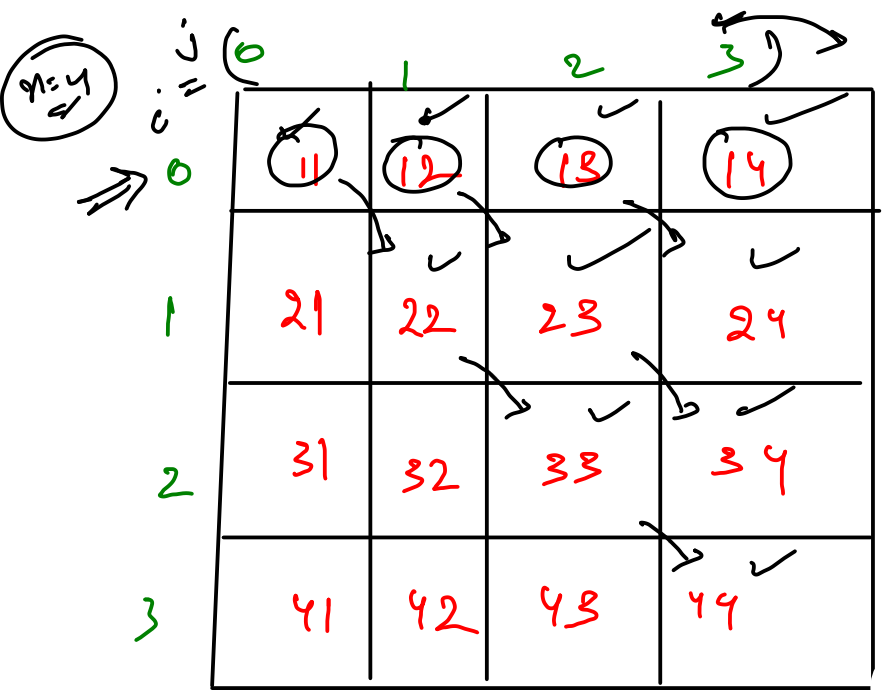
i j

11 → (0,0) ⎫ D1
22 → (1,1) ⎪
33 → (2,2) ⎪
44 → (3,3) ⎭

12 → (0,1) ⎫ D2
23 → (1,2) ⎪
34 → (2,3) ⎭

13 → (0,2) ⎫ D3
24 → (1,3) ⎭

14 → (0,3) ⎫ D4




```java
for(int i = 0 , j = 0 ; j < n ; j++){
    for(int r = i , c = j ; c < n ; r++,c++){
        System.out.println(mat[r][c]);
    }
}
```

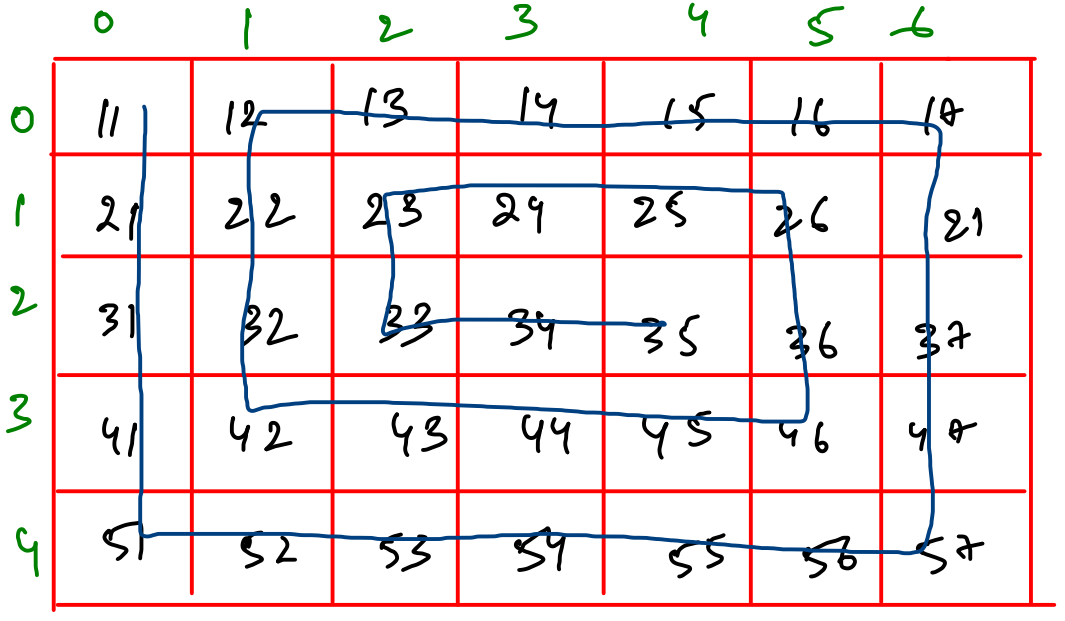
11
22
33
44
12
23
34
13
24
14

i	j	r	c
0	0	0	0
		1	1
		2	2
		3	3
0	1	0	1
		1	2
		2	3
0	2	0	2
		1	3
0	3	0	3

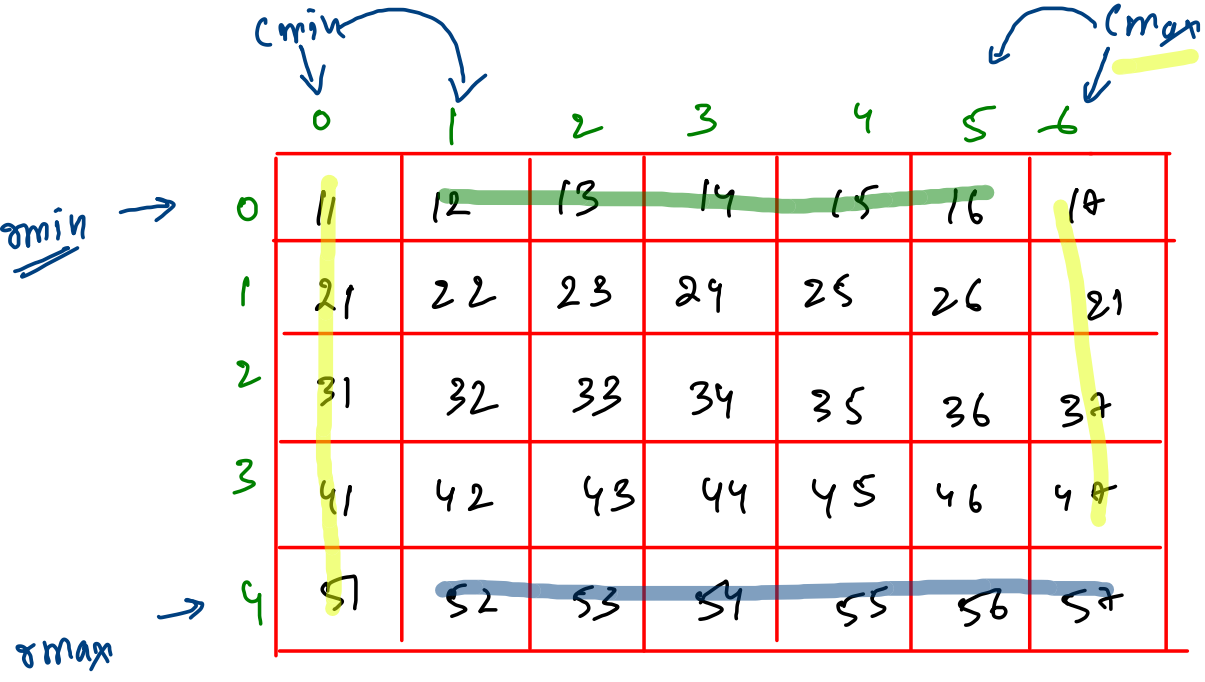
Spiral Display

5x7

Mat



11 16 25
21 15 24
31 14 23
41 13 33
51 12 34
52 22 35
53 32
54 42
55 43
56 44
57 45
4A 46
37 31
27 21
17



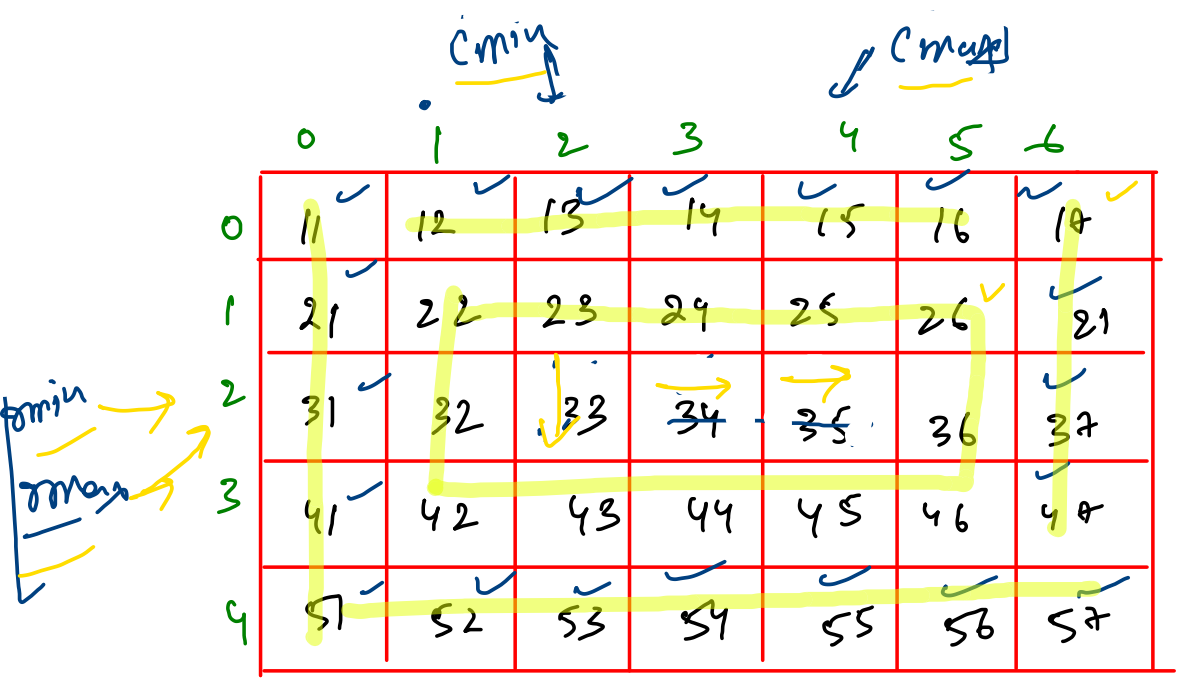
left

bottom

right

top

```java
// leftWall
for(int r = rmin , c = cmin ; r < rmax ; r++){
    System.out.println(mat[r][c]);
}
```



tEle => 35

32

```
while(count < tEle){
    // leftWall
    for(int r = rmin , c = cmin ; r <= rmax ; r++){
        System.out.println(mat[r][c]);
        count++;
    }

    // BottomWall
    for(int r = rmax , c = cmin+1 ; c <= cmax ; c++){
        System.out.println(mat[r][c]);
        count++;
    }

    // RightWall
    for(int r = rmax-1 , c = cmax ; r >= rmin ; r--){
        System.out.println(mat[r][c]);
        count++;
    }

    // TopWall
    for(int r = rmin , c = cmax-1 ; c >= cmin+1 ; c--){
        System.out.println(mat[r][c]);
        count++;
    }

    rmin++;
    cmin++;
    rmax--;
    cmax--;
}
```

(1) (rmin <= rmax && cmin <= cmax)

count => 32

33

34

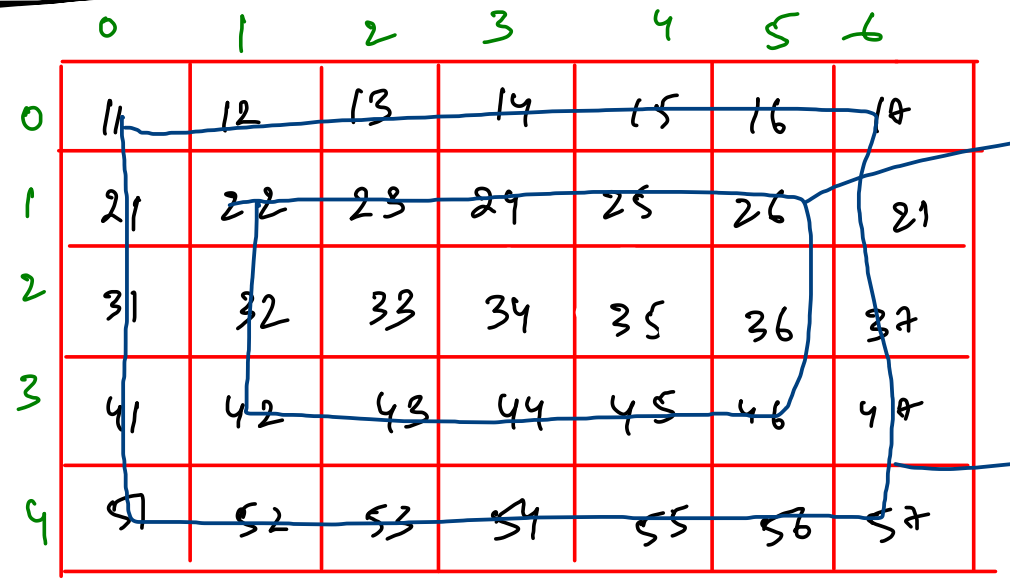
35

r => 2

c => 34

RING ROTATE

shell 2, (rotate by 3)



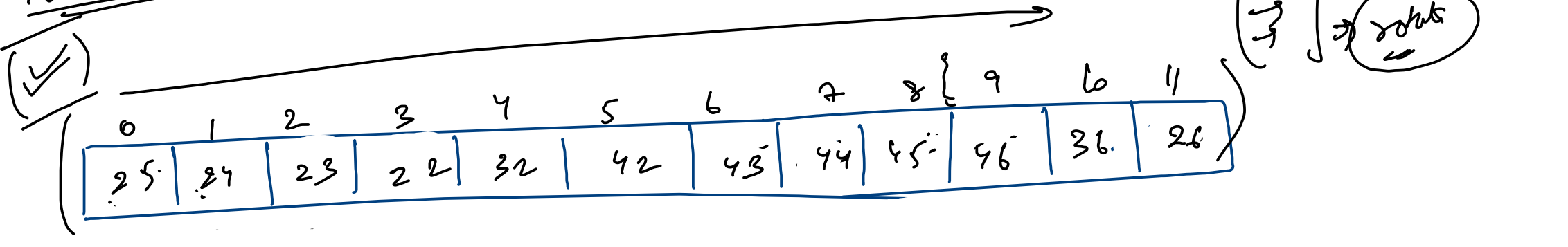
Input

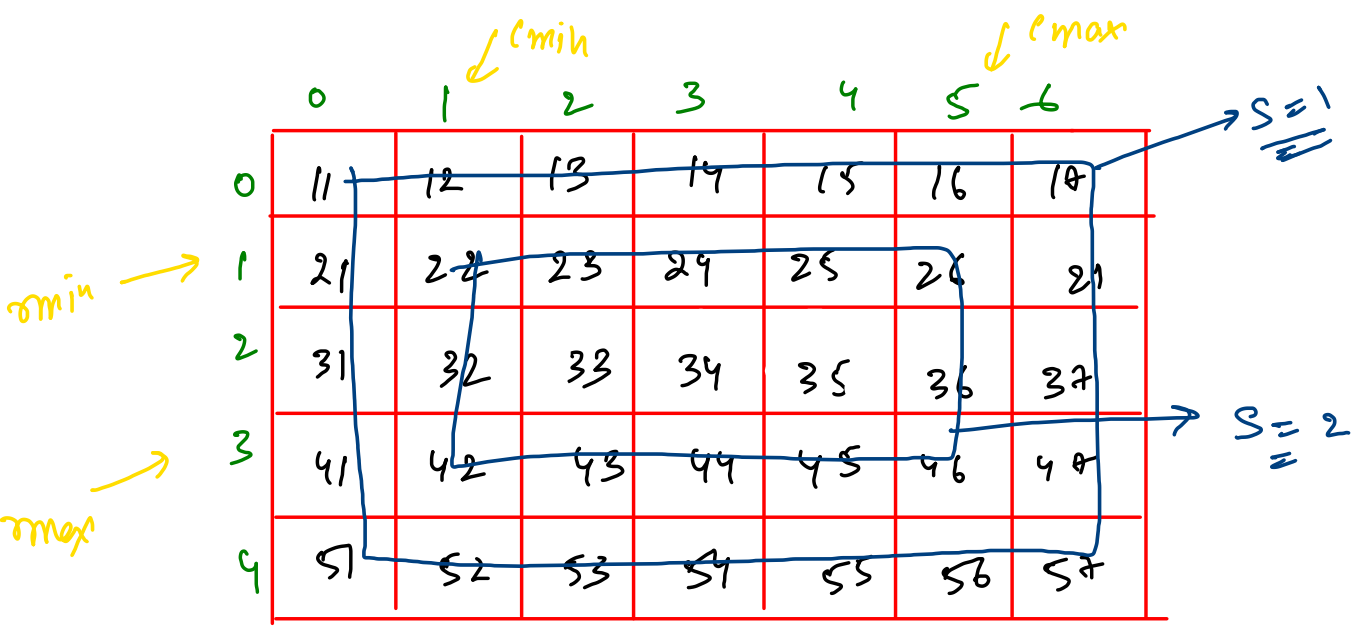
	0	1	2	3	4	5	6
0	11	12	13	14	15	16	17
1	21	22	23	24	25	26	27
2	31	32	33	34	35	36	37
3	41	42	43	44	45	46	47
4	51	52	53	54	55	56	57

Output

	0	1	2	3	4	5	6
0	11	12	13	14	15	16	17
1	21	25	26	36	46	45	27
2	31	24	33	34	35	44	37
3	41	23	22	32	42	43	47
4	51	52	53	54	55	56	57

Rotate an Array !





S	rmin	cmin	rmax	cmax
1	0	0	4	6
2	1	1	3	5
	(S-1)	(S-1)	(n_r - S)	(n_c - S)

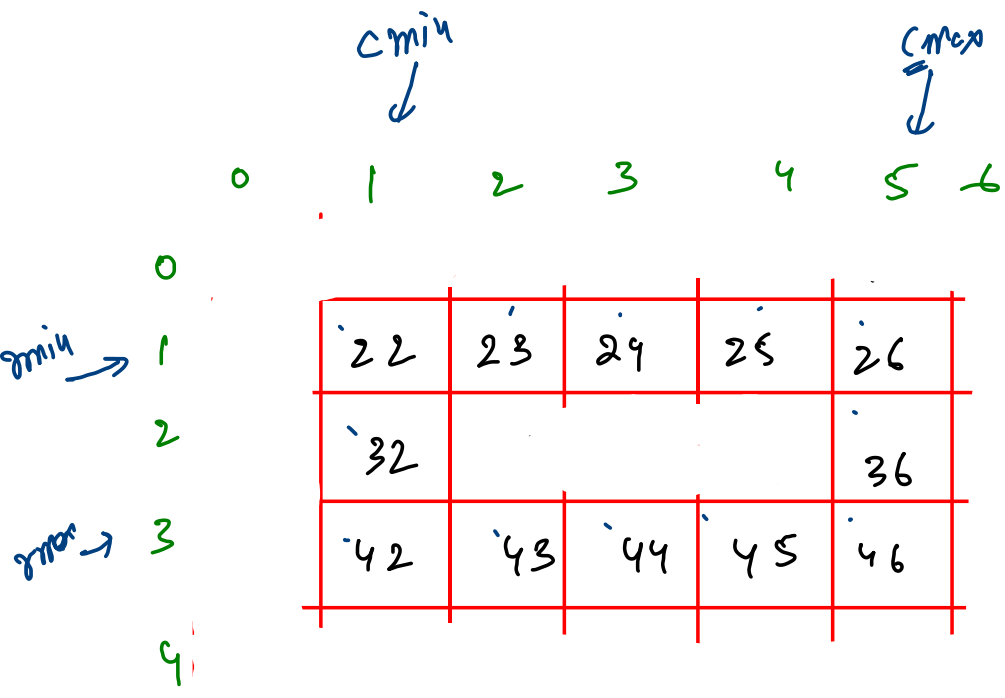
$n_r \Rightarrow 5$

$n_c \Rightarrow 7$

① oned array?

② Traversal → spiral

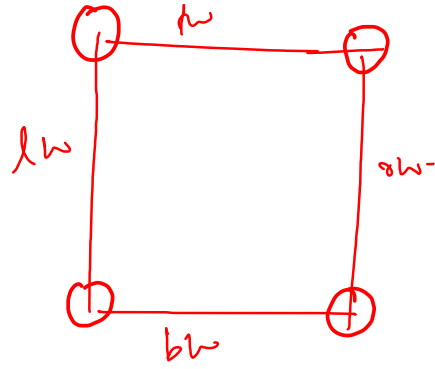
cmin, rmin, cmax, rmax ?



No. of elements

$n(lw) + n(bw) + n(rw) + n(tw) - 4$

(Since $\Rightarrow n(lw) = n(rw)$, $n(bw) = n(tw)$)



$n(l.w.) \Rightarrow r_{max} - r_{min} + 1$

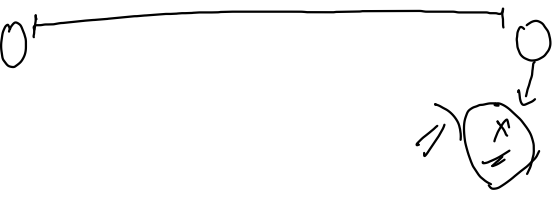
$n(b.w.) \Rightarrow c_{max} - c_{min} + 1$

$N \Rightarrow 2\left[n(lw) + n(bw)\right] - 4$

$N \Rightarrow 2\left[r_{max} - r_{min} + 1 + c_{max} - c_{min} + 1\right] - 4$

$N \Rightarrow 2(r_{max} - r_{min} + c_{max} - c_{min})$

$= 2(3 - 1 + 5 - 1) \Rightarrow 2.(2 + 4) \Rightarrow 12$



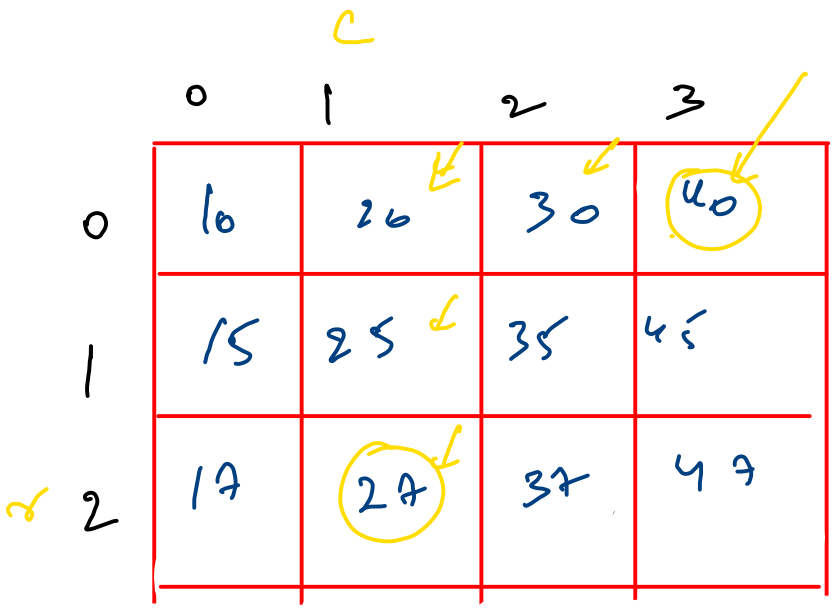
Given, matrix
→ sorted row wise & column wise

	0	1	2	3
0	11	12	13	14
1	21	22	23	24
2	31	32	33	34
3	41	42	43	44

29

	0	1	2	3
0	10	20	30	40
1	15	25	35	45
2	17	27	37	47
3	19	29	39	49

24



flag = 0 1

ele = 27

2

1

```java
// logic

int r = 0 , c = n - 1;
int flag = 0;
while(c >= 0 && r < n){
    if(mat[r][c] == ele){
        // element found
        System.out.println(r+"\n"+c);
        flag = 1;
        break;
    }else if(ele < mat[r][c]){
        c--;
    }else if(ele > mat[r][c]){
        r++;
    }
}
if(flag == 0){
    System.out.println("Not Found");
}
```

smaller = 40

length =

Input

	0	1	2	3
0	11	12	13	14
1	21	22	23	24
2	31	32	33	34
3	41	42	43	44

Transpose

(1)

	0	1	2	3
0	11	21	31	41
1	12	22	32	42
2	13	23	33	43
3	14	24	34	44

reverse all elements row-wise.

(2)

Output

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

(1) Try it yourself
(2) read an article on gfg
(3) Solution video

reach

Discuss with me