

* In Order to learn how to do
Spiral matrix



first, it is necessary to learn
about boundary traversal

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

(6x6)

1 2 3 4 5 6 12 18 24 30 36
35 34 33 32 31 25 19 13 7

if it 6x6

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

reverse

reverse

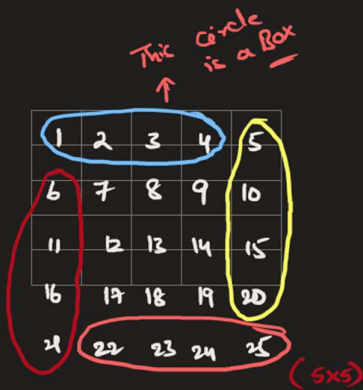
(6x6)

5 no's from 0th row

5 no's from last col

5 no's from last row in reverse

5 no's from 0th col in reverse



if it is $(n \times n)$ size

$(n-1)$ no's from 0th row

$(n-1)$ no's from last col

$(n-1)$ no's from last row in reverse

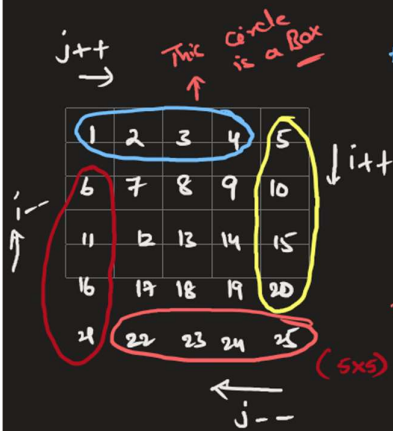
$(n-1)$ no's from 0th col in reverse

Observation:

Printing is done as follows:



' $n-1$ ' is common, because $(n \times n)$ matrix



for Blue Box

```
for (int k=0; k<(n-1); k++)
{
    print(arr[i][k]);
    j++;
}
```

for Orange Box

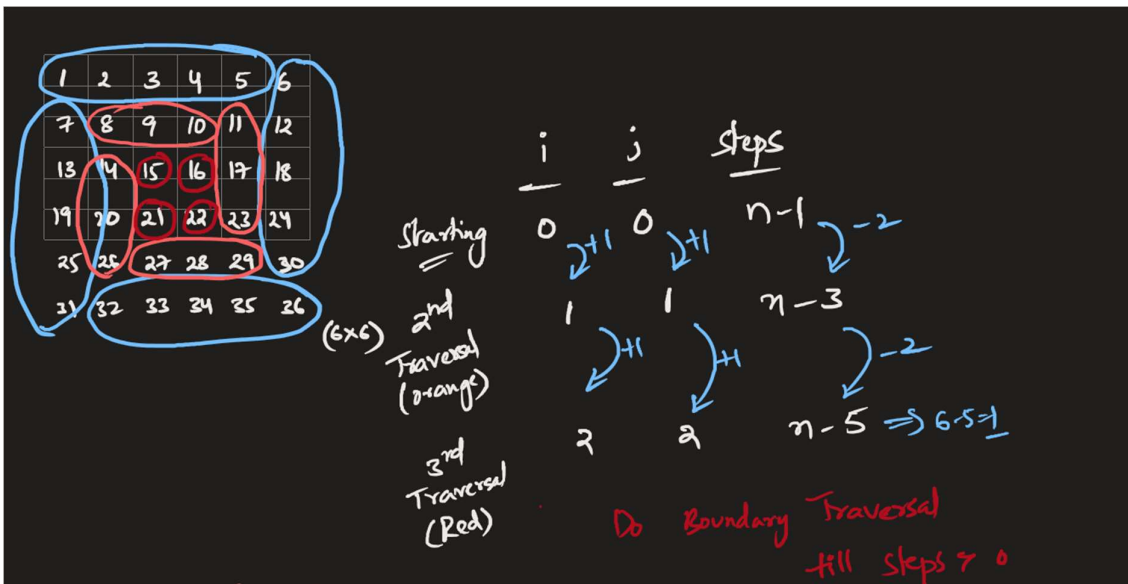
```
for (int k=0; k<(n-1); k++)
{
    print(arr[k][j]);
    i++;
}
```

for yellow Box

```
for (int k=(n-1); k>0; k--)
{
    print(arr[i][k]);
    i++;
}
```

for red Box

```
for (int k=(n-1); k>0; k--)
{
    print(arr[k][j]);
    j--;
}
```



```
int i=0, j=0, steps = n-1;
```

```
while (Steps > 0)
```

```
{
```

Blue loop
 yellow loop
 orange loop
 red loop

⇒ Written above

```
i++;
```

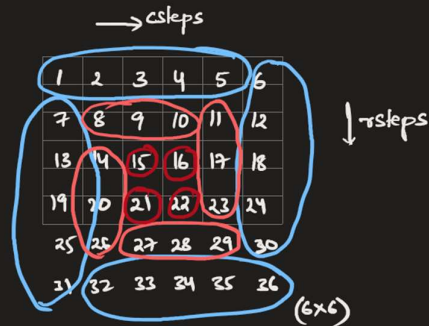
```
j++;
```

```
steps -= 2;
```

```
}
```

* if $(n \times n)$ matrix \rightarrow then $(n-1)$ common in every loop
 \parallel
Steps are only one in every loop

if $(m \times n)$
 row steps \swarrow col steps
 $m-1$ $n-1$



• Updating Steps and Position:

- After completing a full cycle (right, down, left, up), reduce rsteps and csteps by 2 to account for the outer layer being processed.
- Move the starting point inward by incrementing i and j .

	0
0	1
1	7
2	13
3	19
4	25

5x1

	0	1	2
0	1	2	3
1	7	8	9
2	13	14	15
3	19	20	21
4	25	26	27

5x3

In such cases, either rsteps or csteps will equal to 0
 \Rightarrow We have to handle such cases separately

• Edge Cases:

- If there is only one row left to process (rsteps == 0), traverse it from left to right.
- If there is only one column left to process (csteps == 0), traverse it from top to bottom.

\hookrightarrow look at code for algo