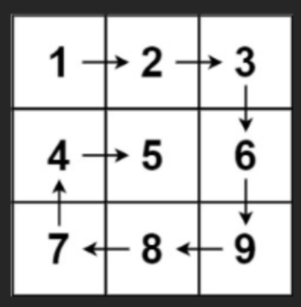
Spiral Matrix

54. Spiral Matrix

Medium | Topics | Companies | Hint

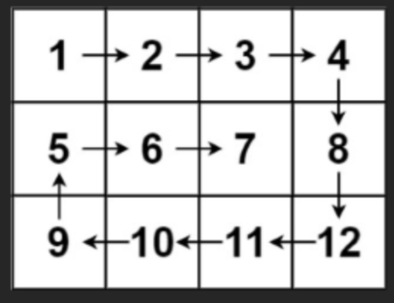
Given an `m x n` `matrix`, return *all elements of the* `matrix` *in spiral order.*

Example 1:




```
Input: matrix = [[1,2,3],[4,5,6],[7,8,9]]
Output: [1,2,3,6,9,8,7,4,5]
```

Example 2:




```
Input: matrix = [[1,2,3,4],[5,6,7,8],[9,10,11,12]]
Output: [1,2,3,4,8,12,11,10,9,5,6,7]
```

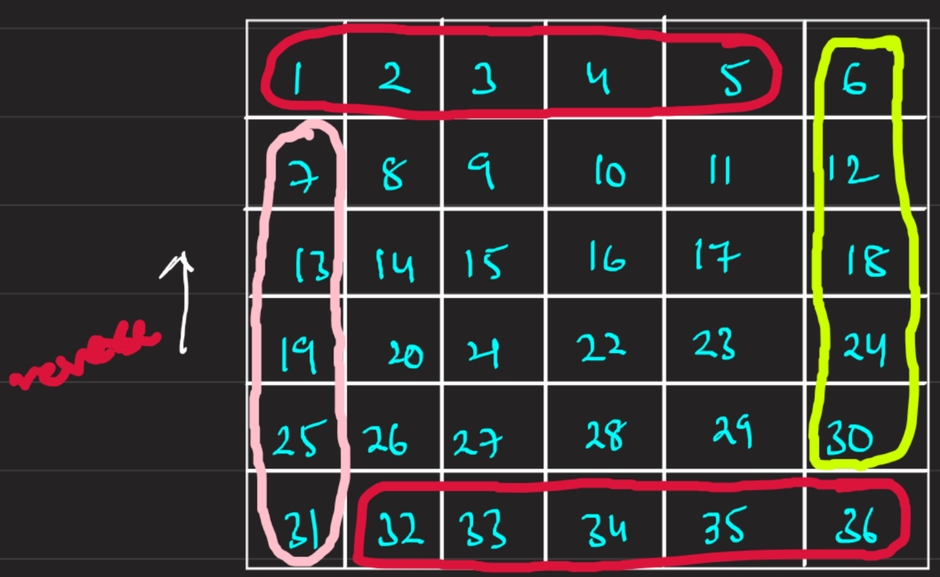
In order to learn about Spiral matrix clearly...

* It is necessary to think about Boundary traversal

output:

1 2 3 4 5 6 12 18 24 30

36 35 34 33 32 31 25 19 13 7



if it is (6x6)

in output, 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)

name these circled portions → suppose, blue box

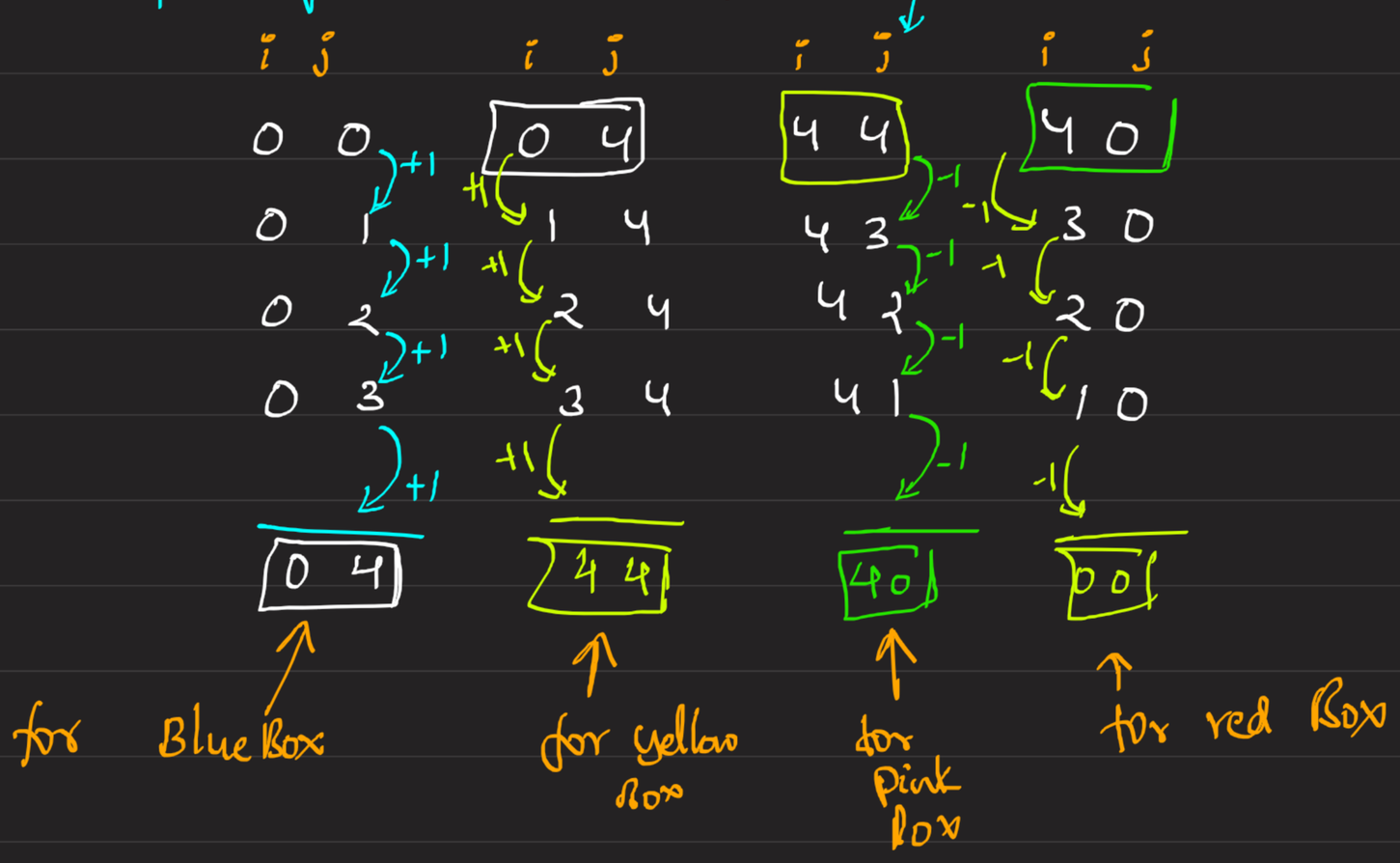
* if it is (nxn) size;

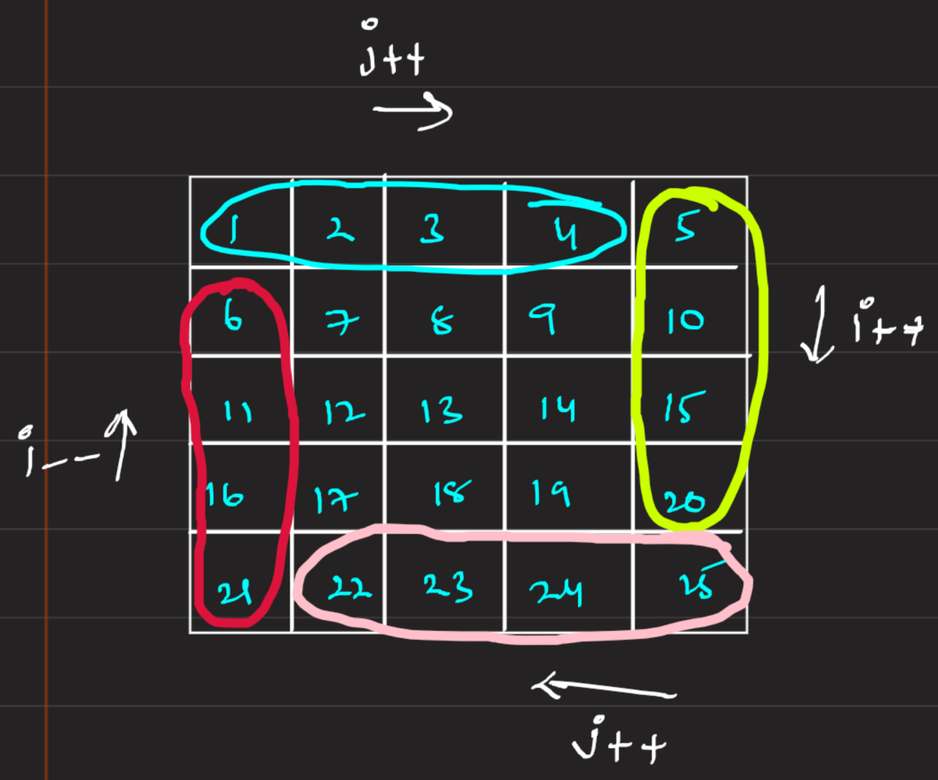
(n-1) no's from 0th row
(n-1) no's from last col
(n-1) no's from last row (in reverse)
(n-1) no's from 0th col in reverse

(5x5)

Observation

printing is done as follows :-





for Blue Box

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

runs for same (n-1) times

for yellow Box

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

for pink Box

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

for red Box

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

Now.... Use this Boundary traversal and Construct Spiral traversal...

direction in every 5 elements we print

	i	j	steps	
first Boundary traversal starts at	0	0	n-1	→ for(6x6)
		+1		
			-2	
Second time from (8)	1	1	n-3	→ we reduce by 2
		+1		
	2	2	n-5	

∴ if you at steps = 5, for (6x6) and do boundary traversal and reduce steps by 2 and again

till **steps > 0** → we get Our Spiral matrix

```
int i=0, j=0, steps = n-1;
while (steps > 0)
{
    // Blue loop
    // yellow loop
    // pink loop
    // red loop

    i++;
    j++;
    step -=2
}
```

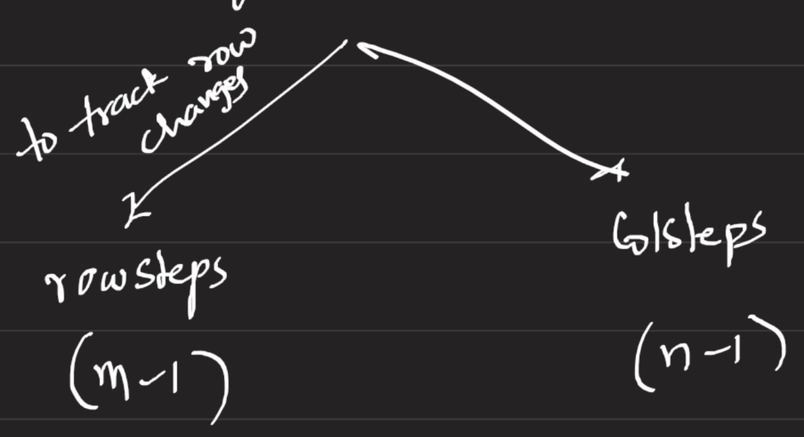
But, But, But....

In our example, we took (n x n) matrix, that's why in every new Boundary traversal, Steps are decreasing by 2

But what if rows & col's are <u>different</u>?

* if (nxn) matrix → then steps are same in rows & col, so we use only one variable to track changes

* if (mxn)



* How to Update rowsteps, colsteps ?

→ After Completing a full Boundary traversal, reduce (rsteps by 2 / csteps by 2) & don't forget to increment (i & j)

ij ::

i	j	rsteps	csteps
0	0	4	0

1
5
15
20
25

1	2	3
7	8	9
13	14	15
19	20	25
27	28	29

i	j	rsteps	csteps
0	0	4	2
+1	+1	-2	-2
1	1	2	0

In such cases, either rsteps Or csteps will become zero

* We have to print all the elements of remaining row / column

* refer to code for clear grasp

```
if(rsteps == 0){
    for(int k = 0;k <= csteps;k++){
        ans.add(matrix[i][j]);
        j++;
    }
}else if(csteps == 0){
    for(int k = 0;k <= rsteps;k++){
        ans.add(matrix[i][j]);
        i++;
    }
}
```

→ if rsteps are zero traverse through csteps & print

because we are traversing column → increment j