* Matrix multiplication :

$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 1 & 2 & 3 \\ 1 & 4 & 5 & 6 \end{array} \quad * \quad \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 10 & 11 \\ 1 & 20 & 21 \\ 2 & 30 & 31 \end{array}$$

mat1 2×3 $m_1 \times n_1$

mat2 3×2 $m_2 \times n_2$

=

	0	1
0	1*10 + 2*20 + 3*30	1*11 + 2*21 + 3*31
1	4*10 + 5*20 + 6*30	4*11 + 5*21 + 6*31

$m_1 \times n_2$

2×2

* you can only multiply $m_1 \times n_1$ matrix with $m_2 \times n_2$ matrix only if $n_1 == m_2$ i.e; no. of cols in mat1 = no. of rows in mat2

* The resultant matrix will be $m_1 \times n_2$

$$(2 \times \cancel{3}) * (\cancel{3} \times 2) = (2 \times 2)$$

$$(m_1 \times n_1) * (m_2 \times n_2) = (m_1 \times n_2)$$

* idea to code,
  - → pick a row from 1st matrix → for
    - → pick a col from 2nd matrix → for
      - → multiply respective r, c and add all → for

$$
\begin{array}{c|ccc}
 & 0 & 1 & 2 \\
\hline
0 & a_{00} & a_{01} & a_{02} \\
1 & a_{10} & a_{11} & a_{12}
\end{array}
\quad * \quad
\begin{array}{c|cccc}
 & 0 & 1 & 2 & 3 \\
\hline
0 & b_{00} & b_{01} & b_{02} & b_{03} \\
1 & b_{10} & b_{11} & b_{12} & b_{13} \\
2 & b_{20} & b_{21} & b_{22} & b_{23}
\end{array}
\quad = \quad
\begin{array}{c|cccc}
 & 0 & 1 & 2 & 3 \\
\hline
0 & c_{00} & c_{01} & c_{02} & c_{03} \\
1 & c_{10} & c_{11} & c_{12} & c_{13}
\end{array}
$$

2×3 $\qquad$ 3×4 $\qquad$ 2×4

$r\,c$

$$c_{00} = \overset{0}{a_{00} \cdot b_{00}} + \overset{1}{a_{01} \cdot b_{10}} + \overset{2}{a_{02} \cdot b_{20}}$$

$$c_{01} = a_{00} \cdot b_{01} + a_{01} \cdot b_{11} + a_{02} \cdot b_{21}$$

$$c_{02} = a_{00} \cdot b_{02} + a_{01} \cdot b_{12} + a_{02} \cdot b_{22}$$

$$c_{10} = a_{10} \cdot b_{00} + a_{11} \cdot b_{10} + a_{12} \cdot b_{20}$$

$$c_{12} = \underset{0}{a_{10} \cdot b_{02}} + \underset{1}{a_{11} \cdot b_{12}} + \underset{2}{a_{12} \cdot b_{22}}$$

$$c[r][c] = a[r][?] * b[?][c]$$

$$c[1][2] = a[1][?] * b[?][2]$$

$k \Rightarrow 0 \to 2$

$k = 0 \to$ no. of rows(b) $\}$ same
$= 0 \to$ no. of cols (a)

$$c[\overset{r}{0}][\overset{c}{3}] = a[0][K] * b[K][3]$$

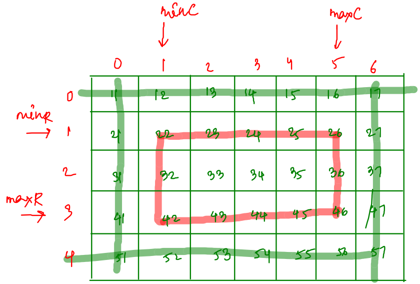
0^{th} Row $\to$ a
3^{rd} Col $\to$ b

$$= a[\overset{r}{0}]\overbrace{[0] * b[0]}^{0}[\overset{c}{3}] + a[\overset{r}{0}]\overbrace{[1] * b[1]}^{1}[\overset{c}{3}] + a[\underset{r}{0}]\underbrace{[2] * b[2]}_{2}[\underset{c}{3}]$$

* Spiral Traversal :

	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

op: 11 12 13 14 15 16 17 27 37 47 57 56
55 54 53 52 51 41 31 21 22 23 24 25
26 36 46 45 44 43 42 32 33 34 35



① topwall → minR
② rightwall → maxC
③ bottom wall → maxR
④ leftwall → minC

boundaries of every layer/box

① Top wall, (belongs to minR, goes from minC → maxC)

```
for (let c = minC; c <= maxC; c++) {
    psw( mat[minR][c] + " ");
}
minR++;
```

(once the Job is completed move to next top wall)

② right wall, (belongs to maxC, goes from minR → maxR)

```
for (let r = minR; r <= maxR; r++) {
    psw (mat [r] [maxC] + " ");
}
maxC--;
```

③ Bottom wall (belongs to maxR, goes from maxC → minC)

```
for (let c = maxC; c >= minC; c--) {
    psw (mat [maxR] [c] + " ");
}
maxR--;
```

④ left wall, (belongs to minC, goes from maxR → minR)

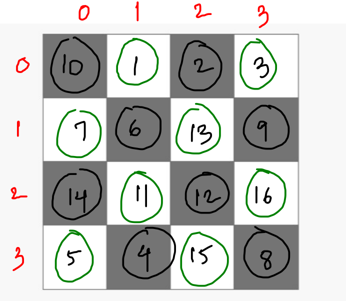
```
for (let r = maxR; r >= minR; r--) {
    psw (mat [r] [minC] + " ");
}
minC++;
```

★ TC : O (rows × cols)
O(N²)
SC : O(1)

check how many times each ele is accessed/touched here it is only once.

★ you need keep running these 4 loops so they will print layer by layer until all the elements in the matrix are covered/printed.

★ Alternate Matrix Sum :



4 x 4 Chess Board

op : 10 + 2 + 6 + 9 + 14 + 12
+ 4 + 8 = bsum

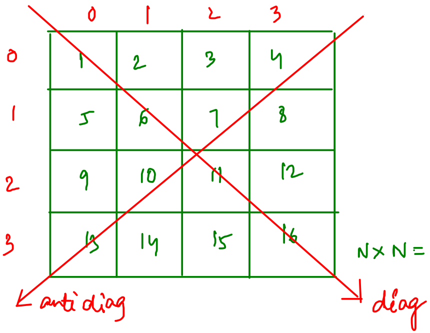
1 + 3 + 7 + 13 + 11 + 16
+ 5 + 15 = wsum

black	white
(0, 0) = 0	(0, 1) = 1
(0, 2) = 2	(0, 3) = 3
(1, 1) = 2	(1, 0) = 1
(1, 3) = 4	(1, 2) = 3
(2, 0) = 2	(2, 1) = 3
(2, 2) = 4	(2, 3) = 5
(3, 1) = 4	(3, 0) = 3
(3, 3) = 6	(3, 2) = 5
even	odd

```
let bsum = 0, wsum = 0;

for (let r = 0; r < rows; r++) {
    for (let c = 0; c < cols; c++) {
        if ( r+c % 2 == 0) {
            bsum += mat[r][c];
        }
        else {
            wsum += mat[r] [c];
        }
    }
}
```

* Diagonal Difference :



$N \times N = 4 \times 4$

op: $diag = 1 + 6 + 11 + 16$

$adiag = 4 + 7 + 10 + 13$

$abs(diag - adiag)$

$= abs(34 - 34)$

$= 0$

diagonal

(0, 0)

(1, 1)

(2, 2)

(3, 3)

$r = c$

(r, r)

(c, c)

anti diagonal

$(0, 3) \Rightarrow (0, n-0-1) \Rightarrow (0, 4-0-1)$

$(1, 2) \Rightarrow (1, n-1-1) \Rightarrow (1, 4-1-1)$

$(2, 1) \Rightarrow (2, n-2-1) \Rightarrow (2, 4-2-1)$

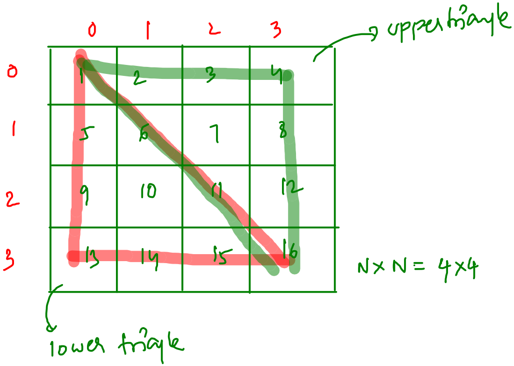
$(3, 0) \Rightarrow (3, n-3-1) \Rightarrow (3, 4-3-1)$

$(r, n-r-1)$

n = rows = cols (square matrix)

```
for (let r = 0; r < n; r++) {
    diag += mat[r][r];
    adiag += mat[r][n-r-1];
}
```

* Sum of upper and lower Triangular Matrix :



lower triangle $(r >= c)$

(0,0)
(1,0) (1,1)
(2,0) (2,1) (2,2)
(3,0) (3,1) (3,2) (3,3)

upper triangle $(r <= c)$

(0,0) (0,1) (0,2) (0,3)
(1,1) (1,2) (1,3)
(2,2) (2,3)
(3,3)