

Rotate by 90 degree

$$0, 2 = 0 + 2 = 2$$

0	1	2
1	2	3
2	3	4

matrix

0	1	2
1	7	4
2	8	5

n last

$$\begin{aligned} [2][0] &= [0][0] \\ [2][1] &= [1][0] \\ [2][2] &= [2][0] \end{aligned}$$

$$\begin{aligned} i+n &= n-1 \\ n &= (n-1-i) \end{aligned}$$

$$\begin{aligned} [0][0] &= [0][2] \\ [0][1] &= [1][2] \\ [0][2] &= [2][2] \\ [1][0] &= [0][1] \\ [1][1] &= [1][1] \\ [1][2] &= [2][1] \\ [2][0] &= [0][0] \\ [2][1] &= [1][0] \\ [2][2] &= [2][0] \end{aligned}$$

$$\begin{aligned} i+n &= n-1 \\ n &= \boxed{n-1} \end{aligned}$$

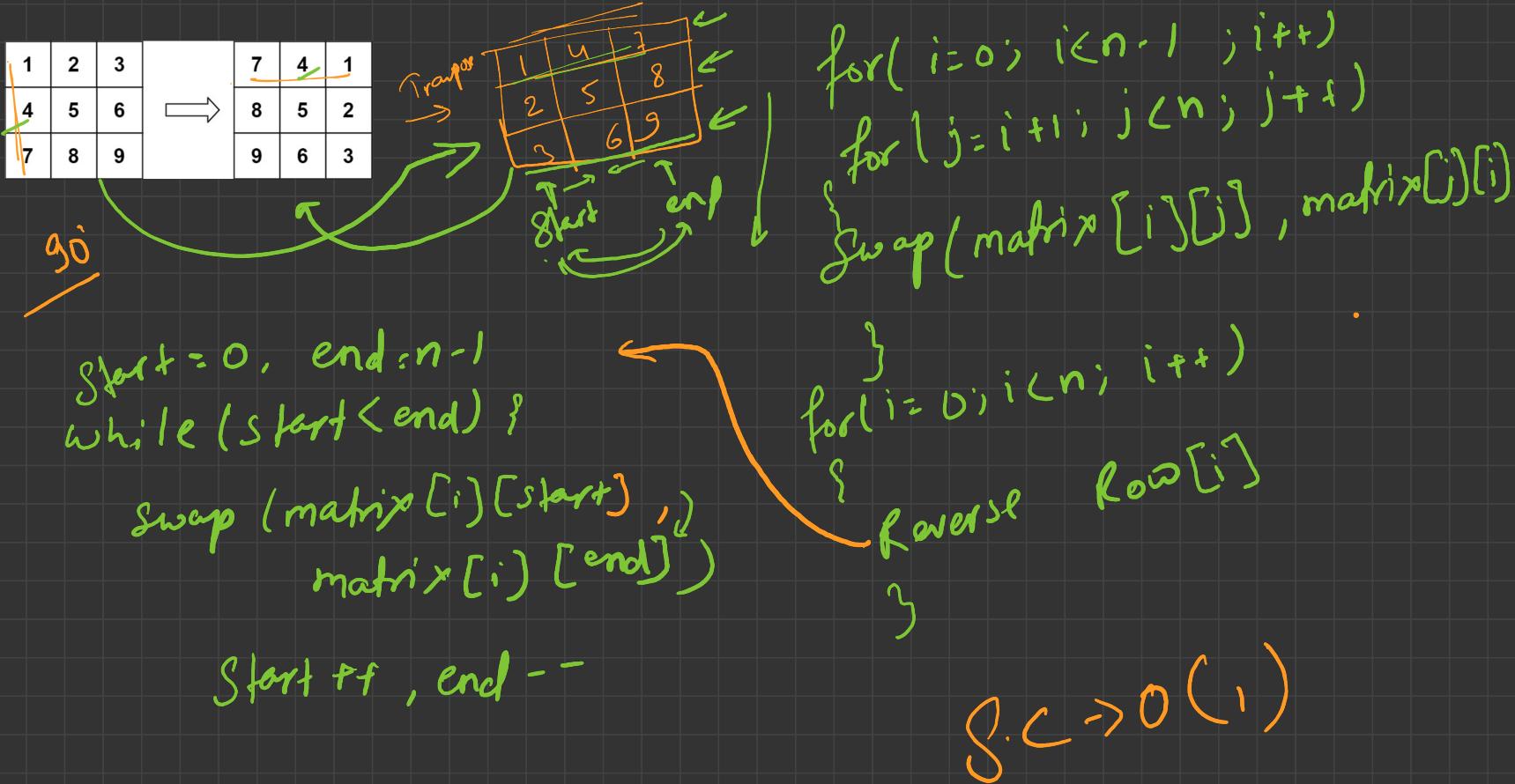
```
for ( i=0 ; i<n ; i++ ) {  
    for ( j=0 ; j<n ; j++ ) { } } T.C O(n^2)
```

$$\text{ans}[j][n-1-i] = \text{matrix}[i][j]$$

}
Transpose

O(1)

S.C O(n^2)
X



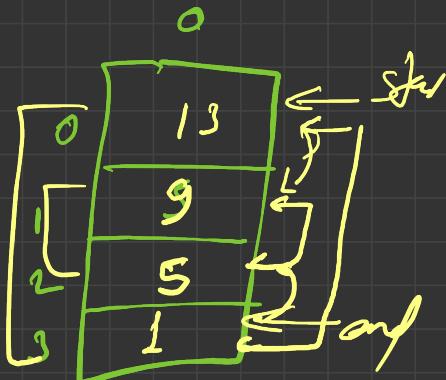
Rotate a Matrix by 180°

Input: [[1, 2, 3, 4],
[5, 6, 7, 8],
[9, 10, 11, 12],
[13, 14, 15, 16]]

Output: [[16, 15, 14, 13],
[12, 11, 10, 9],
[8, 7, 6, 5],
[4, 3, 2, 1]]

matrix

13	14	15	16
9	10	11	12
5	6	7	8
1	2	3	4



Swap $\text{matrix}[0][0]$, $\text{matrix}[3][0]$
" " $\text{matrix}[1][0]$, $\text{matrix}[2][0]$

```

for (j=0; i<n; i++) {
    start = 0    end = n - 1
    while (start < end) {
        swap(matrix[start][i], matrix[end][j]);
        }
    start++, end--;
}
.
.
.

```

$n^2 \{$
 $n^2 + n^2$
 $2n^2 \}$
 $O(n^2)$

```
for( i=0 ; i < n ; i++ ) {  
    start = 0    end = n - 1  
    }  
    while ( start < end ) {  
        swap ( matrix[i][start] , matrix[i][end] )  
        start ++ , end --  
    }  
}
```

Rotate by 90 degree anticlock

↙ ↘

270°

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16



90 + 90 + 90°

Tran

Tr

9	8	12	15
3	7	11	14
2	6	10	13
1	5	9	16

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix} \xrightarrow{90^\circ} \begin{matrix} 7 & 4 & 1 \\ 8 & 5 & 2 \\ 9 & 6 & 3 \end{matrix} \xrightarrow{90^\circ} \begin{matrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{matrix}$$

270°

180 + 90°

123
456
789

369
258
147

→

90

289

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16



4	8	12	16
3	7	11	15
2	6	10	14
1	5	9	13

Row - Reverse

↓
Trans

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

Rotate Matrix K^m time

10⁵⁰

10^{1.4}

~2

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

K^{1.4}

↳ 1 time = 90°

2 " = 180°

3 " = 270°

4 " = 360° = 0° = original

1 ↲ 3 " = 90°

2 ↲ 6 " = 180°

3 ↲ 7 " = 270°

0 ↲ 8 " = 360° = 0°

rotate (matrix)

{

}

int main() {

int K

cin >> K ;

K = 3, 2, 1, 0

K = K % 4 ;

while (K) {

rotate (matrix) ;

K-- ;

y

