Print row wise with condition

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 2 & 3 & 4 \\ 1 & 5 & 6 & 7 & 8 \\ 2 & 9 & 10 & 11 & 12 \\ 3 & 13 & 14 & 15 & 16 \\ 4 & 17 & 18 & 19 & 20 \end{array}$$

4

ans

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 2 & 3 & 4 \\ 1 & 8 & 7 & 6 & 5 \\ 2 & 9 & 10 & 11 & 12 \\ 3 & 16 & 15 & 14 & 13 \\ 4 & 17 & 18 & 19 & 20 \end{array}$$

col = arr[0].length

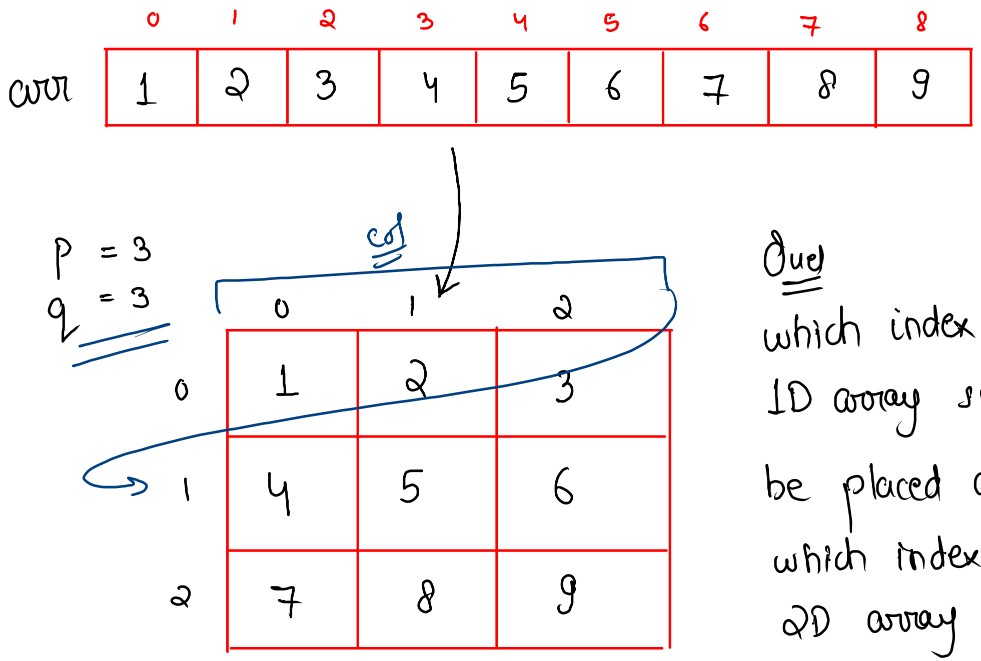
approach

↳ reverse all odd indexed rows

code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int m = scn.nextInt();
    int n = scn.nextInt();
    int[][] arr = new int[m][n];
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] = scn.nextInt();
        }
    }

    printRowwise(arr);

    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            System.out.print(arr[i][j] + " ");
        }
        System.out.println();
    }
}

public static void printRowwise(int[][] arr) {
    for (int i = 0; i < arr.length; i++) {
        if ( i % 2 != 0 ) {
            int si = 0;
            int ei = arr[0].length - 1;
            while ( si < ei ) {
                int temp = arr[i][si];
                arr[i][si] = arr[i][ei];
                arr[i][ei] = temp;

                si++;
                ei--;
            }
        }
    }
}
```

Convert 1-D Array to 2-D Array



Que

which index of 1D array should be placed at which index of 2D array

index of 1D array (idx)	index of 2D array	
	row	col
0 $\rightarrow$	(0 ,	0)
1 $\rightarrow$	(0 ,	1)
2 $\rightarrow$	(0 ,	2)
3 $\rightarrow$	(1 ,	0)
4 $\rightarrow$	(1 ,	1)
5 $\rightarrow$	(1 ,	2)
6 $\rightarrow$	(2 ,	0)
7 $\rightarrow$	(2 ,	1)
8 $\rightarrow$	(2 ,	2)

$p = 3, \quad q = 3$

idx		row	col
0	$\longrightarrow$	$0/3 = 0$,	$0 \% 3 = 0$
1	$\longrightarrow$	$1/3 = 0$,	$1 \% 3 = 1$
2	$\longrightarrow$	$2/3 = 0$,	$2 \% 3 = 2$
3	$\longrightarrow$	$3/3 = 1$,	$3 \% 3 = 0$
4	$\longrightarrow$	$4/3 = 1$,	$4 \% 3 = 1$
5	$\longrightarrow$	$5/3 = 1$,	$5 \% 3 = 2$
6	$\longrightarrow$	$6/3 = 2$,	$6 \% 3 = 0$
7	$\longrightarrow$	$7/3 = 2$,	$7 \% 3 = 1$
8	$\longrightarrow$	$8/3 = 2$,	$8 \% 3 = 2$

1D

2D

$$\boxed{\begin{aligned} row &= idx / q \\ col &= idx \% q \end{aligned}}$$

Imp

$\longleftarrow$ Imp

Code

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

    int p = scn.nextInt();
    int q = scn.nextInt();
    int[][] ans = convert1Dto2D(arr, n, p, q);
    for (int i = 0; i < p; i++) {
        for (int j = 0; j < q; j++) {
            System.out.print(ans[i][j] + " ");
        }
        System.out.println();
    }
}

public static int[][] convert1Dto2D(int[] arr, int n, int p, int q) {
    int[][] arr2d = new int[p][q];
    for (int idx = 0; idx < n; idx++) {
        int r = idx / q;
        int c = idx % q;
        arr2d[r][c] = arr[idx];
    }
    return arr2d;
}
```

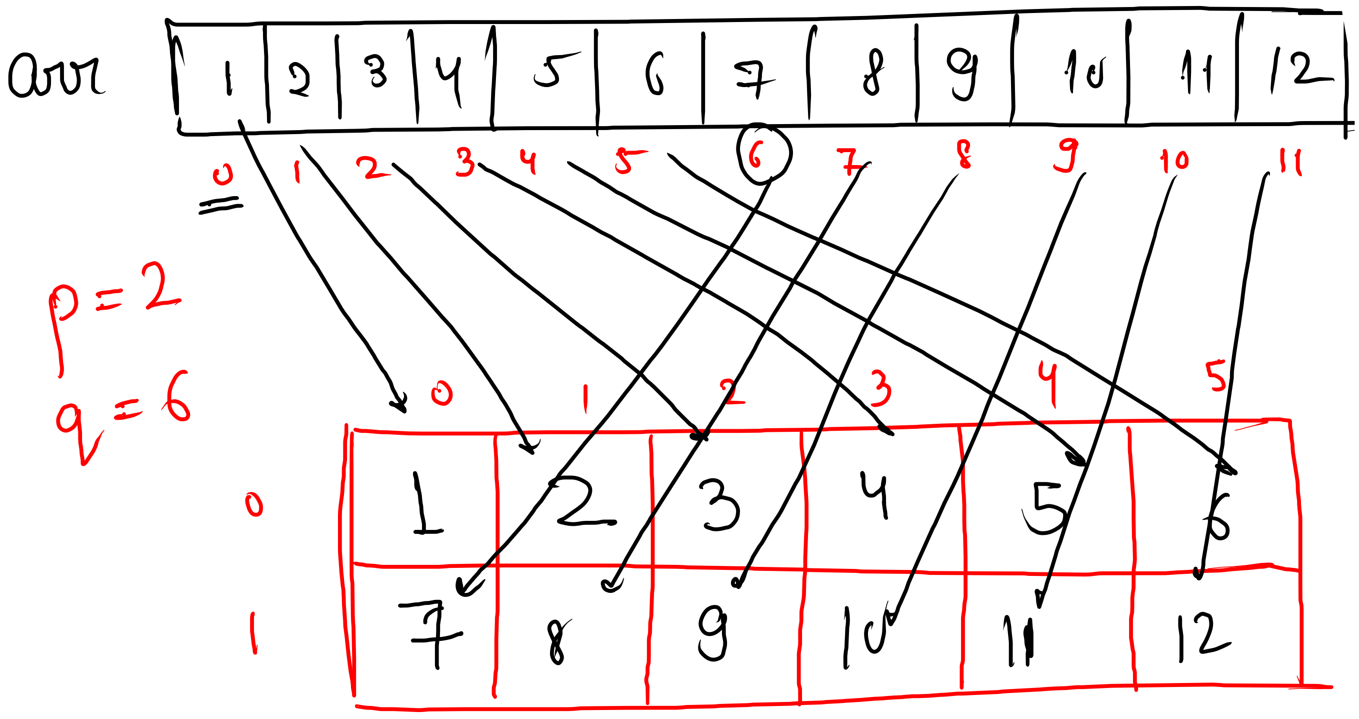
T.C

$O(\underline{N})$ ↳ size of 1D array

or

$O(p \times q)$

S.C = $O(p \times q)$



p = 3
q = 4

$$\begin{bmatrix} 3\times4 \\ 4\times3 \\ 6\times2 \\ 2\times6 \end{bmatrix}$$

$$\boxed{\begin{aligned} r &= idx / q \\ c &= \underline{idx \,\%\, q} \end{aligned}}$$

$0 \rightarrow r = 0/6 = 0$
$c = 0\%6 = 0$

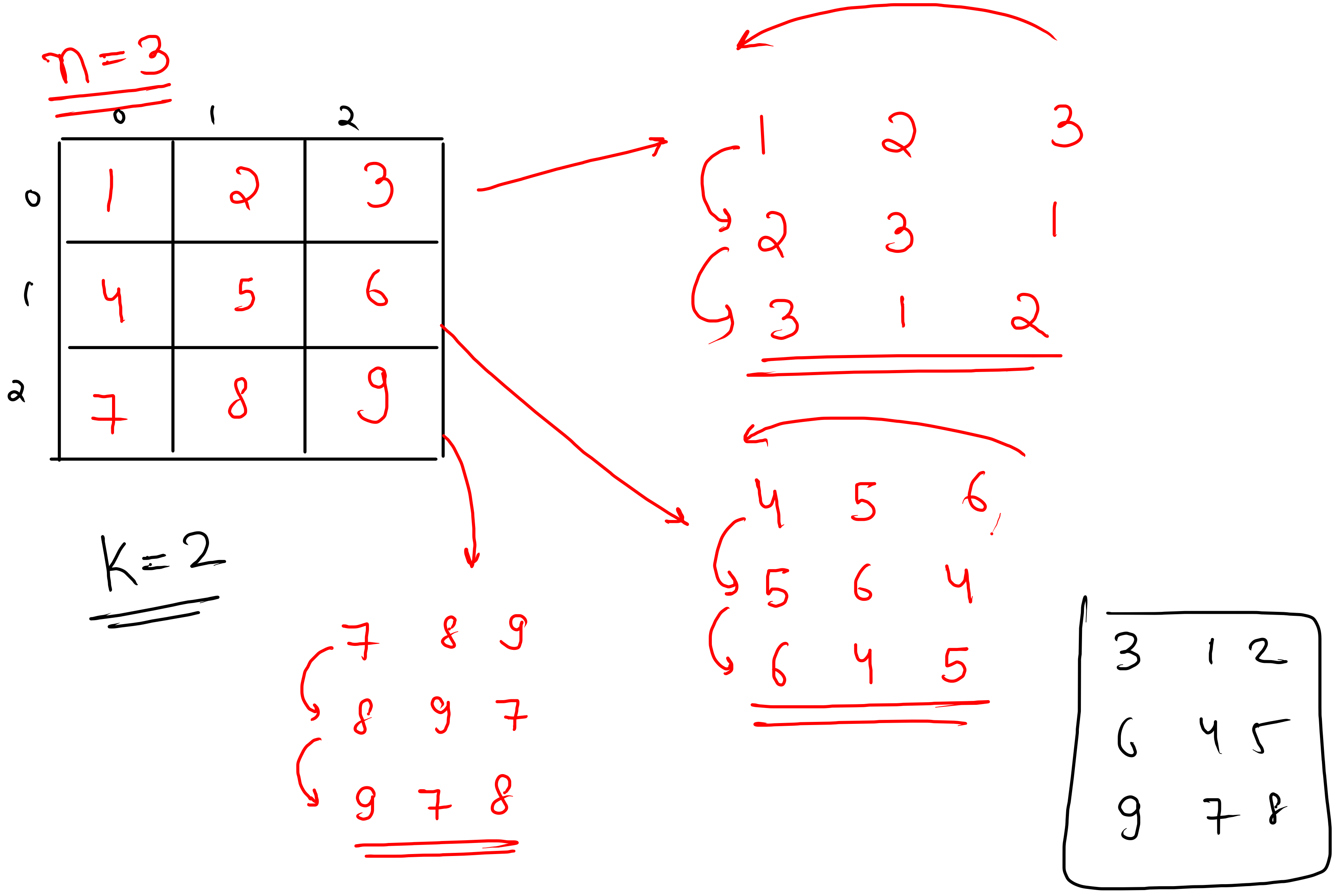
$1 \rightarrow r = 1/6 = 0$
$c = 1\%6 = 1$

$2 \rightarrow r = 2/6 = 0$
$c = 2\%6 = 2$

$6 \rightarrow r = 6/6 = 1$
$c = 6\%6 = 0$

$7 \rightarrow r = 7/6 = 1$
$c = 7\%6 = 1$

Shift Matrix Row-Wise



$$
\begin{array}{c|ccc}
 & 0 & 1 & 2 \\
\hline
0 & 1 & 2 & 3 \\
1 & 4 & 5 & 6 \\
2 & 7 & 8 & 9
\end{array}
$$

$$
\begin{array}{ccc}
1 & 2 & 3 \\
2 & 3 & 1 \\
3 & 1 & 2
\end{array}
\qquad
\begin{array}{ccc}
4 & 5 & 6 \\
5 & 6 & 4 \\
6 & 4 & 5
\end{array}
\qquad
\begin{array}{ccc}
7 & 8 & 9 \\
8 & 9 & 7 \\
9 & 7 & 8
\end{array}
$$

$$
\boxed{\begin{array}{ccc}
3 & 1 & 2 \\
6 & 4 & 5 \\
9 & 7 & 8
\end{array}}
$$

Code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[][] arr = new int[n][n];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] = scn.nextInt();
        }
    }
    int k = scn.nextInt();
    k = n - k;  // for clockwise direction
    shiftRowwise(arr, k, n);

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            System.out.print(arr[i][j] + " ");
        }
        System.out.println();
    }
}

public static void shiftRowwise(int[][] arr, int k, int n) {
    for (int i = 0; i < n; i++) {
        reverse( arr[i], n - k, n - 1 );   // reverse last k elements
        reverse( arr[i], 0, n - k - 1 );   // reverse remaining elemetns
        reverse( arr[i], 0, n - 1 );       // reverse entire array
    }
}

public static void reverse(int[] arr, int si, int ei) {
    while ( si < ei ) {
        swap(arr, si, ei);
        si++;
        ei--;
    }
}

public static void swap(int[] arr, int i, int j) {
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}
```

~~$k = n - k$~~