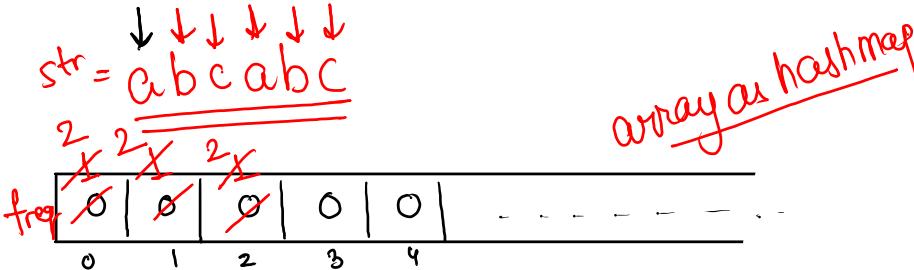


## Good String Checker



```
for  
[ char ch= str.charAt(i);  
int idx= ch- 'a';  
freq[ idx]++; ] } to create  
hashmap
```

---

```
int count = 0;
```

```
for  
[ if (f[i] != 0) {  
count = f[i];  
}  
if (f[i] != 0 & count != f[i]) {  
return false;  
}  
return true; ] } to check  
for good  
string
```

$$\begin{aligned} 'a' - 'a' &= 0 \\ 'b' - 'a' &= 1 \\ 'c' - 'a' &= 2 \\ 'a' - 'a' &= 0 \\ 'b' - 'a' &= 1 \\ 'c' - 'a' &= 2 \end{aligned}$$

## Transpose of matrix

	0	1	2	3
0	1	2 5	3 9	4 13
1	2 5	6	7 10	8 14
2	8 3	10 7	11	12 15
3	15 4	14 8	15 12	16

⇒

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

Note:- If you traverse on all the elements, then array will remain as it is.

(because all swaps will happen twice)

$$\begin{aligned}
 n=0, & \quad 0 \rightarrow 3 \\
 n=1, & \quad 1 \rightarrow 3 \\
 n=2, & \quad 2 \rightarrow 3 \\
 n=3, & \quad 3 \rightarrow 3
 \end{aligned}$$

indexes

2 → (0, 1)	3 → (0, 2)
5 → (1, 0)	9 → (2, 0)
4 → (0, 3)	7 → (1, 2)
13 → (3, 0)	10 → (2, 1)
8 → (1, 3)	12 → (2, 3)
14 → (3, 1)	15 → (3, 2)

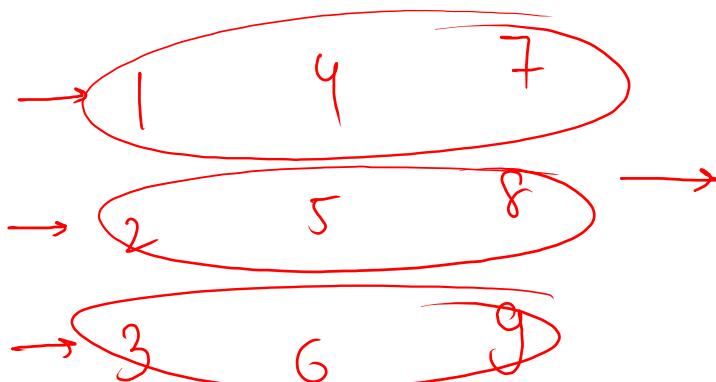
# Rotate The Matrix by 90 Degree (in clockwise direction)

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

90°

4	5	1
8	5	2
6	9	3

or  
Transpose



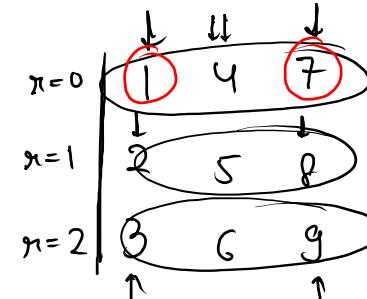
or

7	4	1
8	5	2
9	6	3

Steps :-

- = 1) take transpose of matrix
- = 2) reverse all the rows

how to reverse rows



$i=0, \boxed{7 \ 4 \ 1}$   
 $i=1, \boxed{8 \ 5 \ 2}$   
 $i=2, \boxed{9 \ 6 \ 3}$

$i=0, j=0, 2$ $j=1, 1$	$i=1, j=0, 2$ $j=1, 1$	$i=2, j=0, 2$ $j=1, 1$
---------------------------	---------------------------	---------------------------

①

```
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();
        }
    }
    rotateby90(arr, n);
}
```

```
public static void rotateby90(int[][] arr, int n) {
    // step 1
    transpose(arr, n); // 1

    // step 2
    reverseRows(arr, n); // 2

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

②

```
public static void transpose(int[][] arr, int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < i; j++) {
            int temp = arr[i][j];
            arr[i][j] = arr[j][i];
            arr[j][i] = temp;
        }
    }
}
```

```
public static void reverseRows(int[][] arr, int n) {
    for (int i = 0; i < n; i++) { // traverse for all rows
        for (int j = 0; j < n / 2; j++) {
            int temp = arr[i][j];
            arr[i][j] = arr[i][n - 1 - j];
            arr[i][n - 1 - j] = temp;
        }
    }
}
```

reverse rows

				$n=4$
$i=0$	6	5	8 4 6	
$i=1$	2	7	8 9	
$i=2$	3	10	11 12	
$i=3$	13	14	15 16	

$$1 \rightarrow (0,0), 6 \rightarrow (0,3)$$

$$\text{col} = 0, 3$$

$n=4$

$i=0,$	$(0,0), (0,3)$	$i=1,$
$i=2,$		
$i=3$		

swap  $(0,1) (0,2)$

$$j=0, 4-1-0 = 3$$

$$j=1, 4-1-1 = 2$$

reverse rows

				$n=4$
$i=0$	6	5	8 4 6	
$i=1$	2	7	8 9	
$i=2$	3	10	11 12	
$i=3$	13	14	15 16	

$$1 \rightarrow (0,0), 6 \rightarrow (0,3)$$

$$\text{col} = 0, 3$$

$n=4$

$i=0,$	$(0,0), (0,3)$	$i=1,$
$i=2,$		
$i=3$		

swap  $(0,1) (0,2)$

$$j=0, 4-1-0 = 3$$

$$j=1, 4-1-1 = 2$$

# Rotate The Matrix by 180 Degree

```
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();  
        }  
    }  
  
    rotateby90(arr, n);  
    rotateby90(arr, n);  
  
    // print  
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < n; j++) {  
            System.out.print(arr[i][j] + " ");  
        }  
        System.out.println();  
    }  
}
```

calling function  
twice

```
public static void rotateby90(int[][] arr, int n) {  
    // step 1  
    transpose(arr, n);  
  
    // step 2  
    reverseRows(arr, n);  
}  
  
public static void transpose(int[][] arr, int n) {  
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < i; j++) {  
            int temp = arr[i][j];  
            arr[i][j] = arr[j][i];  
            arr[j][i] = temp;  
        }  
    }  
}  
  
public static void reverseRows(int[][] arr, int n) {  
    for (int i = 0; i < n; i++) { // travser for all rows  
        for (int j = 0; j < n / 2; j++) {  
            int temp = arr[i][j];  
            arr[i][j] = arr[i][n - 1 - j];  
            arr[i][n - 1 - j] = temp;  
        }  
    }  
}
```

# Convert 1-D Array to 2-D Array

```
9  
1 2 3 4 5 6 7 8 9  
3 3
```

=

$$\underline{\underline{n = 9}}$$

1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8

$$\text{rows} = 3$$

$$\text{col} = 3$$

1	2	3
4	5	6
7	8	9

1D  $\rightarrow$  2D

$$0 \rightarrow (0, 0)$$

$$1 \rightarrow (0, 1)$$

$$2 \rightarrow (0, 2)$$

$$3 \rightarrow (1, 0)$$

$$4 \rightarrow (1, 1)$$

$$5 \rightarrow (1, 2)$$

$$6 \rightarrow (2, 0)$$

$$7 \rightarrow (2, 1)$$

$$8 \rightarrow (2, 2)$$

## 1D → 2D

$$0 \rightarrow (0, 0)$$

$$1 \rightarrow (0, 1)$$

$$2 \rightarrow (0, 2)$$

$$3 \rightarrow (1, 0)$$

$$4 \rightarrow (1, 1)$$

$$5 \rightarrow (1, 2)$$

$$6 \rightarrow (2, 0)$$

$$7 \rightarrow (2, 1)$$

$$8 \rightarrow (2, 2)$$

row = 3, col = 3

$$\begin{matrix} 20 \\ i \\ j \end{matrix} \quad (0, 0) = 0 * 3 + 0 = 0$$

$$(0, 1) = 0 * 3 + 1 = 1$$

$$(0, 2) = 0 * 3 + 2 = 2$$

$$(1, 0) = 1 * 3 + 0 = 3$$

$$(1, 1) = 1 * 3 + 1 = 4$$

$$(1, 2) = 1 * 3 + 2 = 5$$

$$(2, 0) = 2 * 3 + 0 = 6$$

$$(2, 1) = 2 * 3 + 1 = 7$$

$$(2, 2) = 2 * 3 + 2 = 8$$

$i * \text{col} + j$

$(i * \text{col} + j)$  ~~gmp~~

~~gmp~~

Note for 2D to 1D

$i = \text{idx}/\text{col}$

$j = \text{idx \% col}$

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 row = scn.nextInt();
    int col = scn.nextInt();

    convert1dto2d(arr, n, row, col);
}

public static void convert1dto2d(int[] arr, int n, int row, int col) {
    int[][] ans = new int[row][col];
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            ans[i][j] = arr[i * col + j];
        }
    }

    // print
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            System.out.print(ans[i][j] + " ");
        }
        System.out.println();
    }
}
```

R.W

Modify The Matrix