Rotate The Matrix by 90 Degree

Problem | Submissions | Leaderboard | Discussions

Take a **square** matrix of size **n*n** as input, and **rotate** the matrix by **90 degree**.

→ Transpose

→ Interchange Row

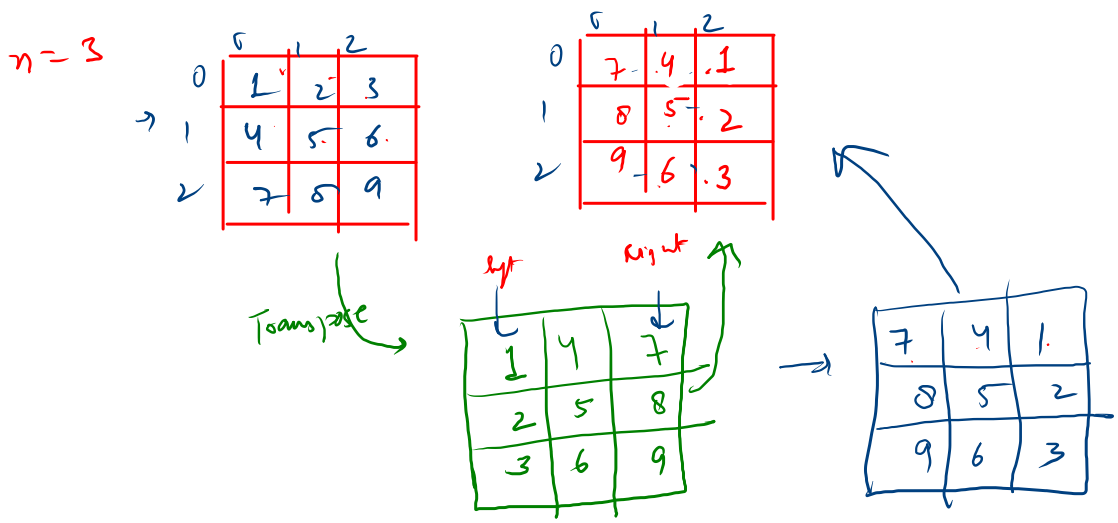
→ Rotate

Sample Input 0

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

Sample Output 0

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



```java
public static void transpose(int arr[][]){

    int n=arr.length;

    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 reverse(int arr[][],int row){
    int n=arr.length;

    int left=0;
    int right=n-1;

    while(left<right){
        int temp=arr[row][left];
        arr[row][left]=arr[row][right];
        arr[row][right]=temp;

        left++;
        right--;
    }
}
```

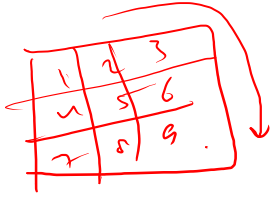
```java
public static void interchangeRowValues(int arr[][]){

    int n=arr.length;

    for(int i=0;i<n;i++){
        reverse(arr,i);
    }
}

public static void display(int arr[][]){
    int n=arr.length;

    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 rotateBy90(int arr[][]){

    transpose(arr);
    interchangeRowValues(arr);
    display(arr);
}
```

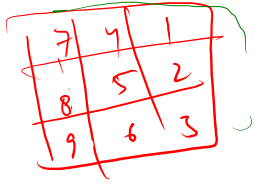
transpose

1 row → 1 col

2 row - 2 col

3 row - 3 col





1 5 →

⇒ transpose + Interchange Row values

Rotate The Matrix by 180 Degree

Problem | Submissions | Leaderboard | Discussions

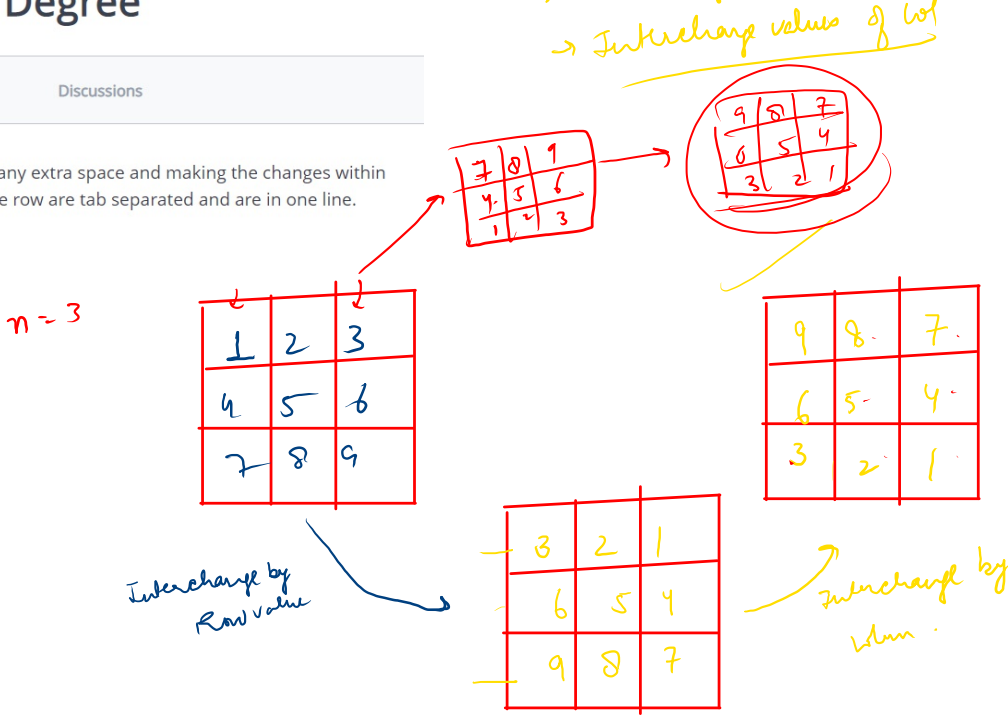
Given a n*n matrix, rotate it by 180 degrees, without taking any extra space and making the changes within the matrix. Print the final matrix such that all elements of the row are tab separated and are in one line.

Sample Input 0

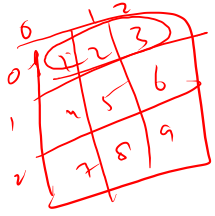
```
3
1 2 3
4 5 6
7 8 9
```

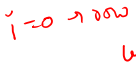
Sample Output 0

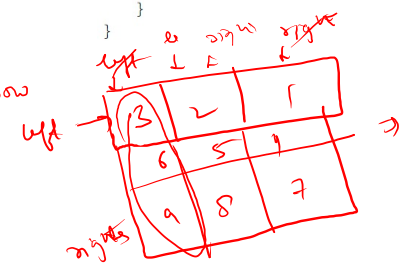
```
9 8 7
6 5 4
3 2 1
```



```java
public static void reverseByRow(int arr[][],int row){
    int n=arr.length;

    int left=0;
    int right=n-1;

    while(left<right){
        int temp=arr[row][left];
        arr[row][left]=arr[row][right];
        arr[row][right]=temp;

        left++;
        right--;
    }
}

public static void interChangeByRow(int arr[][]){
    int n=arr.length;

    for(int i=0;i<n;i++){
        reverseByRow(arr,i);
    }
}
```







```java
public static void interChangeByRow(int arr[][]){
    int n=arr.length;

    for(int i=0;i<n;i++){
        reverseByRow(arr,i);
    }
}
public static void reverseByCol(int arr[][],int col){
    int n=arr.length;

    int left=0;
    int right=n-1;

    while(left<right){
        int temp=arr[left][col];
        arr[left][col]=arr[right][col];
        arr[right][col]=temp;

        left++;
        right--;
    }
}
```

```java
public static void interChangeByCol(int arr[][]){
    int n=arr.length;

    for(int j=0;j<n;j++){
        reverseByCol(arr,j);
    }
}

public static void display(int arr[][]){
    int n=arr.length;

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

Convert 1-D Array to 2-D Array

Problem | Submissions | Leaderboard | Discussions

Take an **array** of size **N** as input, representing a 1-D array.

There are many possible factors of N, for eg:- p * q = N.

Now take **p and q** as input and print the 2-D array with dimensions as **p*q**.

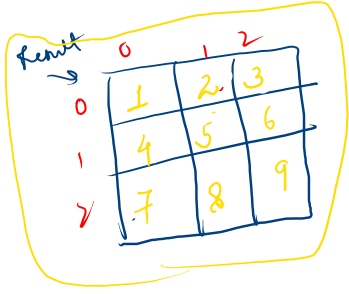
Note: It is **guaranteed** that a 2-D array will be formed.

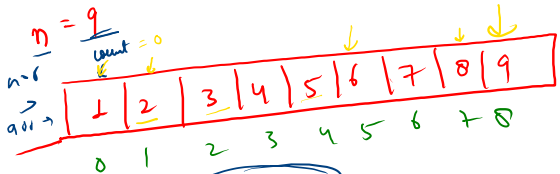
Sample Input 0

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

Sample Output 0

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





p * q = N

p = 3

q = 3

1	2	3
4	5	6
7	8	9

```
for(int i=0; i<p; i++)
{
    for(int j=0; j<q; j++)
    {
        result[i][j] = arr[count];
        count++;
    }
}
```

count = 0 1 2 3 4 5 6 7 8

i=0 j=0 1 2 3

i=1 j=0 1 2 3

i=2 j=0 1 2 3

i=3 X