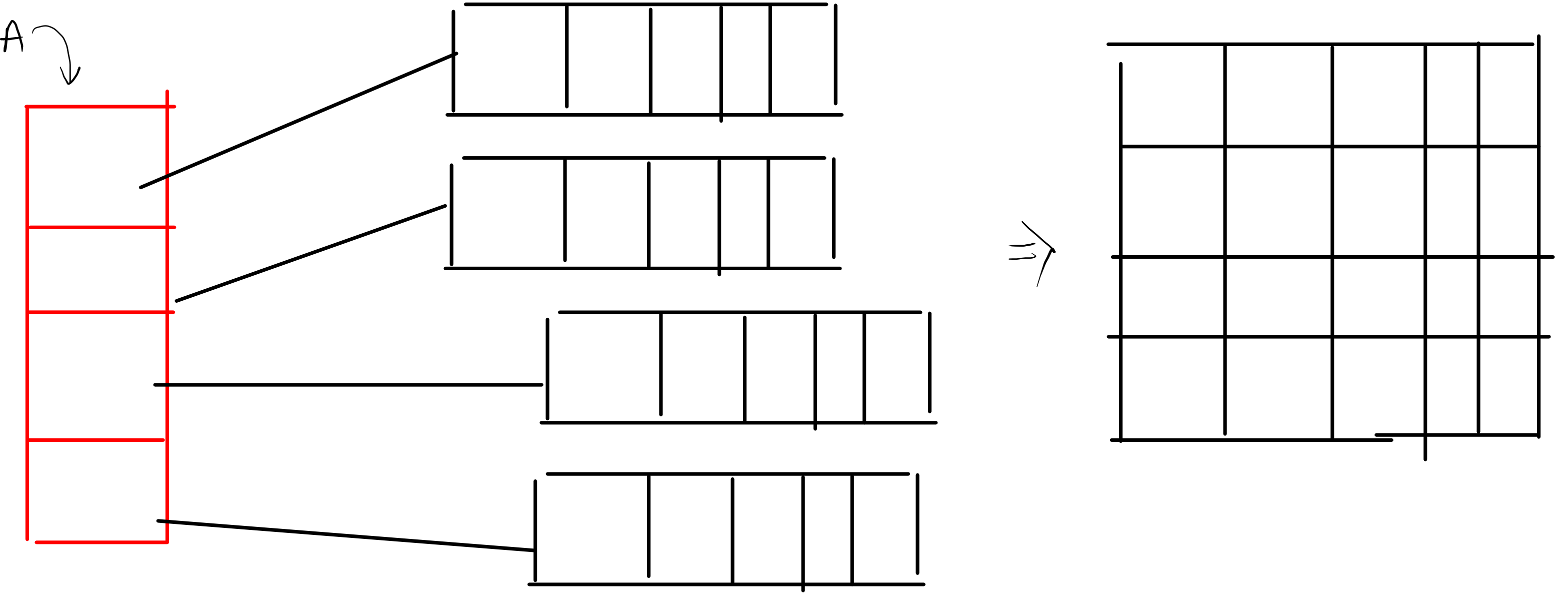
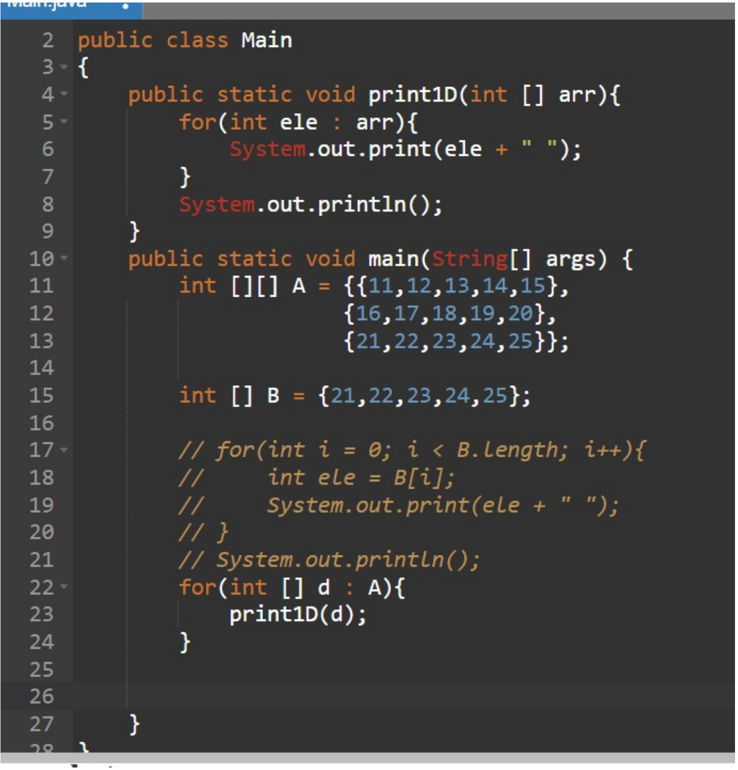

A



```java
public class Main
{
    public static void print1D(int [] arr){
        for(int ele : arr){
            System.out.print(ele + " ");
        }
        System.out.println();
    }
    public static void main(String[] args) {
        int [][] A = {{11,12,13,14,15},
                     {16,17,18,19,20},
                     {21,22,23,24,25}};

        int [] B = {21,22,23,24,25};

        // for(int i = 0; i < B.length; i++){
        //     int ele = B[i];
        //     System.out.print(ele + " ");
        // }
        // System.out.println();
        for(int [] d : A){
            print1D(d);
        }


    }
}
```

Print row wise with condition

ID → L to R

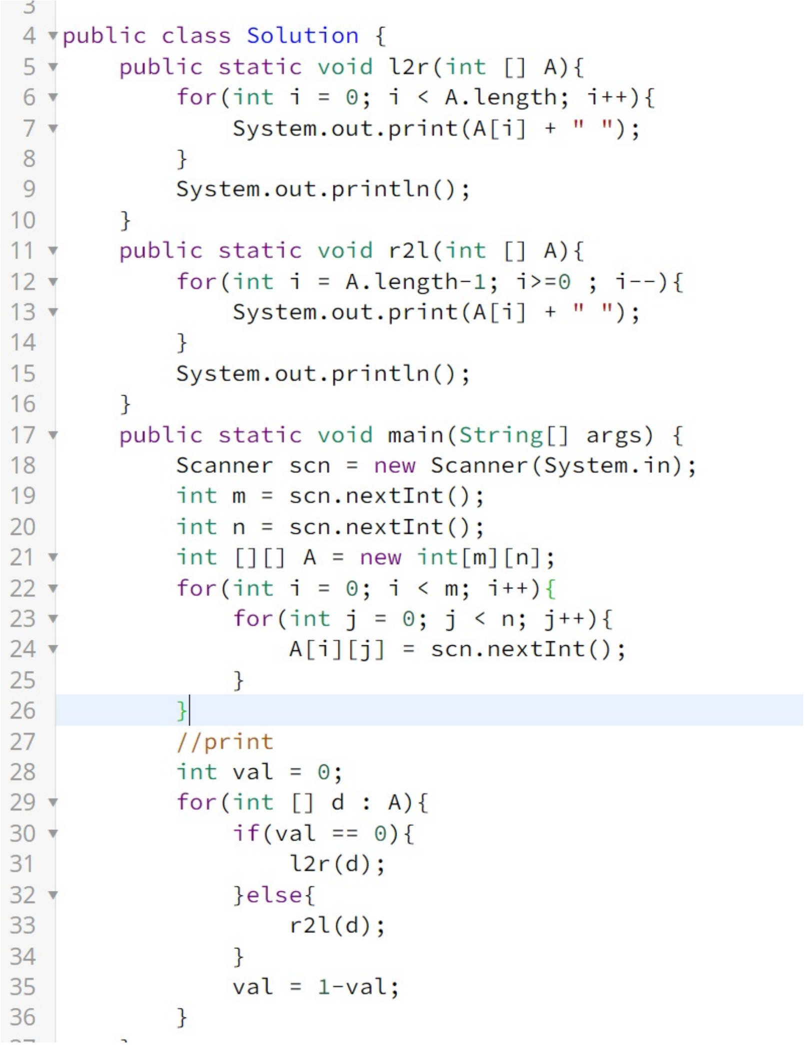
↳ R to L

Sample Input 0

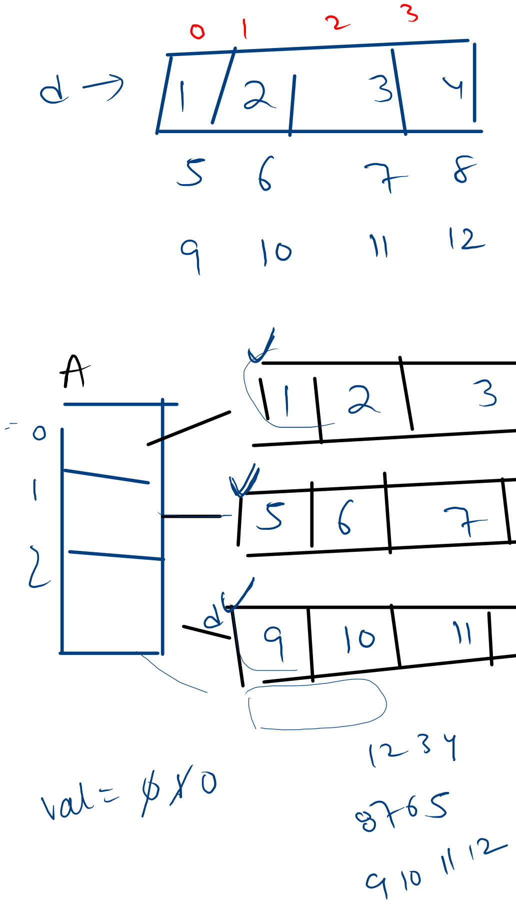
```
3
3
3 1 2
3 0 2
4 5 4
```

Sample Output 0

```
3 1 2
2 0 3
4 5 4
```



```
public class Solution {
    public static void l2r(int [] A){
        for(int i = 0; i < A.length; i++){
            System.out.print(A[i] + " ");
        }
        System.out.println();
    }
    public static void r2l(int [] A){
        for(int i = A.length-1; i>=0 ; i--){
            System.out.print(A[i] + " ");
        }
        System.out.println();
    }
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int m = scn.nextInt();
        int n = scn.nextInt();
        int [][] A = new int[m][n];
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                A[i][j] = scn.nextInt();
            }
        }
        //print
        int val = 0;
        for(int [] d : A){
            if(val == 0){
                l2r(d);
            }else{
                r2l(d);
            }
            val = 1-val;
        }
```



Convert 1-D Array to 2-D Array

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

Sample Input 1

```
6
1 2 3 4 5 6
3 2
```

Sample Output 1

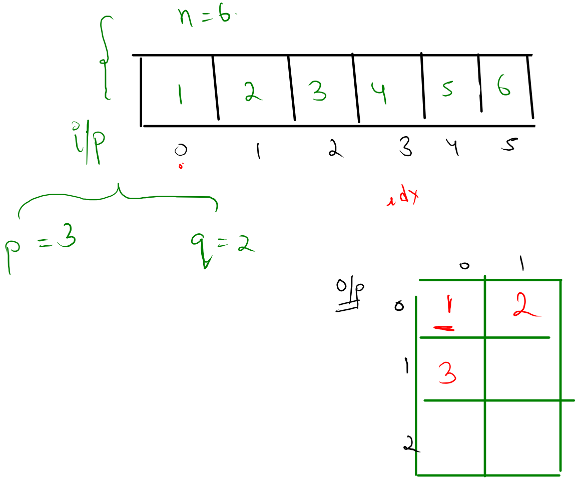
```
1 2
3 4
5 6
```

Sample Input 0

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

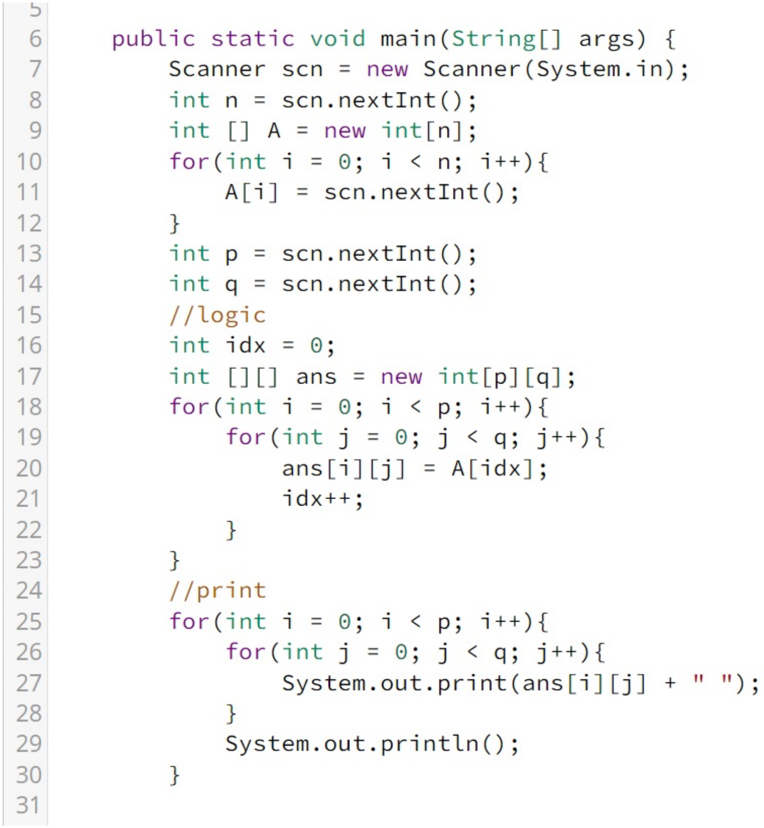
Sample Output 0

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



A[idx]

```
for ( int i=0   i<p ; i++)
{
    for (int j=0 ; j<q ; j++)
    {
        print( A[i][j] + " ");
    }
}
```



```java
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int [] A = new int[n];
        for(int i = 0; i < n; i++){
            A[i] = scn.nextInt();
        }
        int p = scn.nextInt();
        int q = scn.nextInt();
        //logic
        int idx = 0;
        int [][] ans = new int[p][q];
        for(int i = 0; i < p; i++){
            for(int j = 0; j < q; j++){
                ans[i][j] = A[idx];
                idx++;
            }
        }
        //print
        for(int i = 0; i < p; i++){
            for(int j = 0; j < q; j++){
                System.out.print(ans[i][j] + " ");
            }
            System.out.println();
        }

```

$n=6$

1	2	3	4	5	6
0	1	2	3	4	5

$p=3$

$q=2$

idx = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ 6

ans

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 1 & 2 \\ 1 & 3 & 4 \\ 2 & 5 & 6 \end{array}$$

p x q

3x2

i = ~~0~~ ~~1~~ 2

0 < 3

1 < 3 ✓

2 < 3

3 < 3

j = ~~0~~ ~~1~~ 2

0 < 2

1 < 2

2 < 2

n = 7

k = 2

3	4	5	6	7	1	2
13	14	15	16	17	11	12
22	23	24	25	26	20	21
13	14	15	16	17	11 →	12
22	23	24	25	26	20	21
3	4	5	6	7	1	2
13	14	15	16	17	9	8

3	4	5	6	7	1	2
13	14	15	16	17	11	12
22	23	24	25	26	20	21
13	14	15	16	17	11	12
22	23	24	25	26	20	21
3	4	5	6	7	1	2
13	14	15	16	17	9	8

rotate left

k=0	1	2	3	4
k=1	2	3	4	1
k=2	3	4	1	2
k=3	4	1	2	3
k=4	1	2	3	4
k=5	2	3	4	1

n unique rotate

5 % 4 = 1

k=2

2 % 4 = 2

R = 5

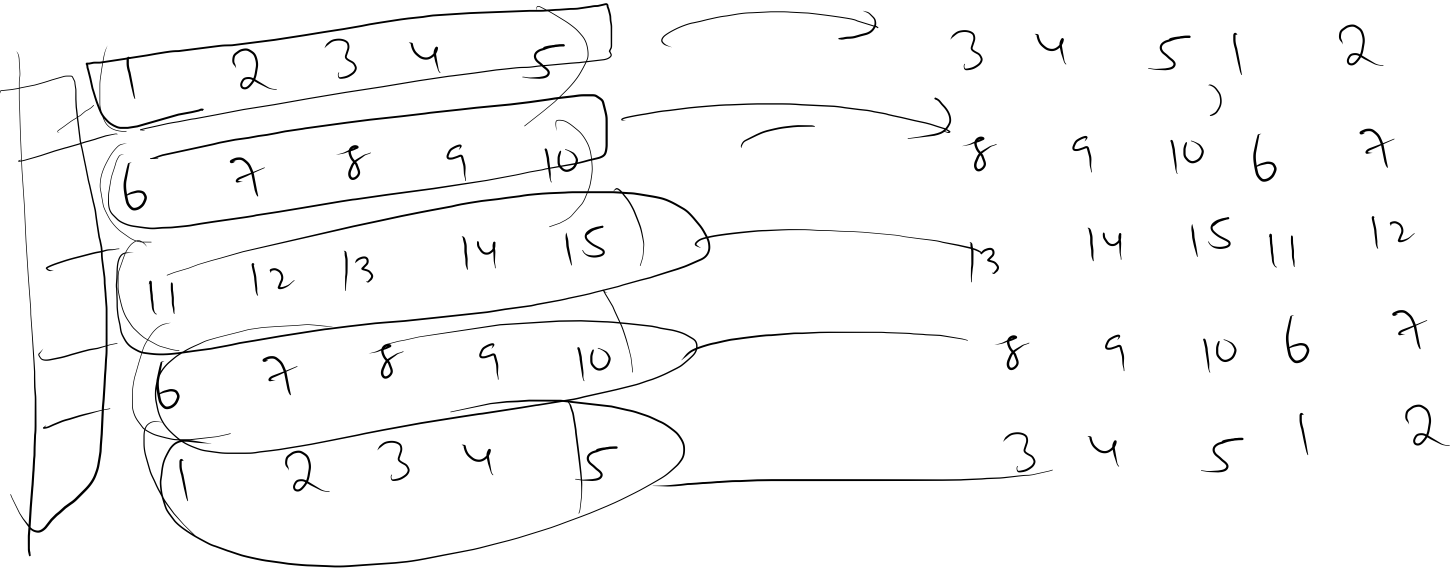
k = 1

k = 13.

k = k % n

n=5   k=2

✓



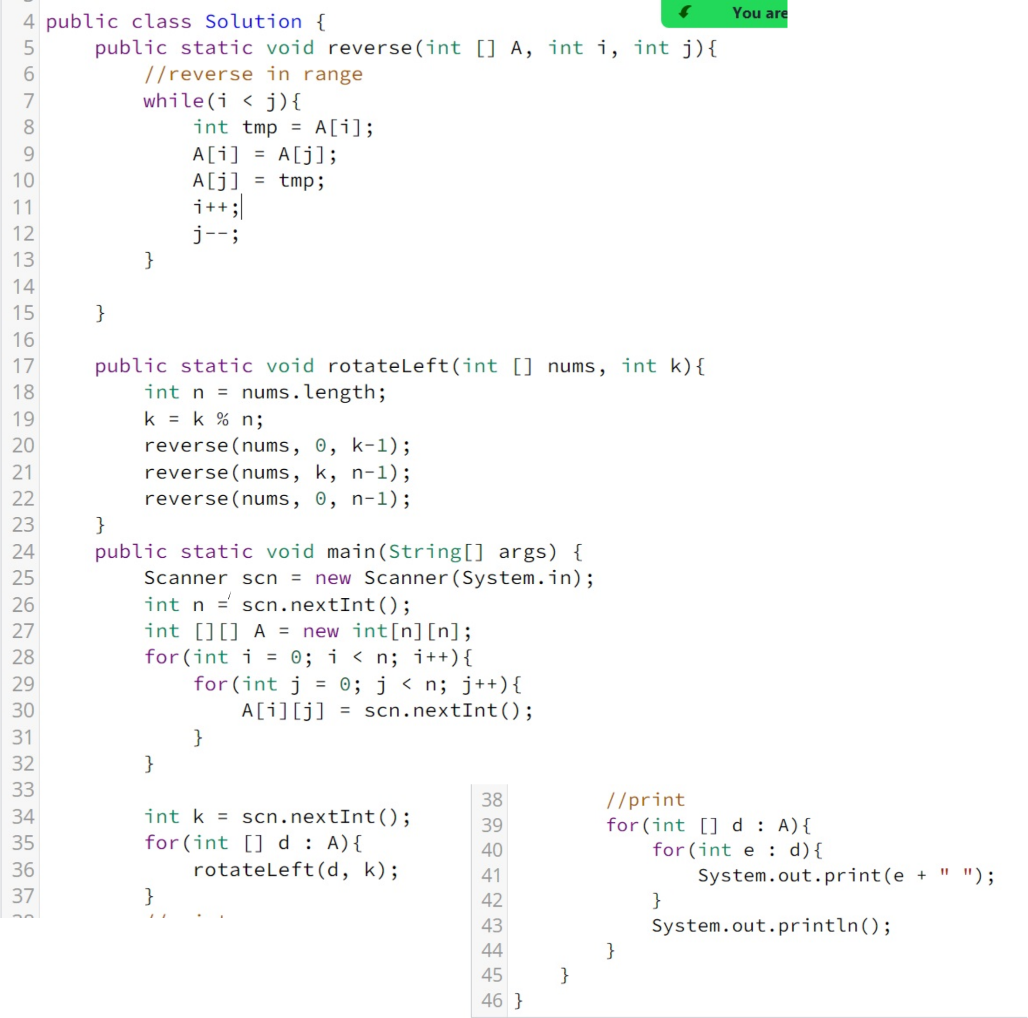
1 2 3 4 5 → 3 4 5 1 2

6 7 8 9 10 → 8 9 10 6 7

11 12 13 14 15 → 13 14 15 11 12

6 7 8 9 10 → 8 9 10 6 7

1 2 3 4 5 → 3 4 5 1 2



```java
public class Solution {
    public static void reverse(int [] A, int i, int j){
        //reverse in range
        while(i < j){
            int tmp = A[i];
            A[i] = A[j];
            A[j] = tmp;
            i++;
            j--;
        }

    }

    public static void rotateLeft(int [] nums, int k){
        int n = nums.length;
        k = k % n;
        reverse(nums, 0, k-1);
        reverse(nums, k, n-1);
        reverse(nums, 0, n-1);
    }
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int [][] A = new int[n][n];
        for(int i = 0; i < n; i++){
            for(int j = 0; j < n; j++){
                A[i][j] = scn.nextInt();
            }
        }

        int k = scn.nextInt();
        for(int [] d : A){
            rotateLeft(d, k);
        }
        //print
        for(int [] d : A){
            for(int e : d){
                System.out.print(e + " ");
            }
            System.out.println();
        }
    }
}
```

Modify The Matrix

Once upon a time, there was a company that was developing a system to track the inventory levels of different products in different **warehouses**. They had a boolean matrix Mat of size **M X N**, where each cell represented the availability of a product in a specific warehouse. If the value of a cell was **true (or 1)**, it meant that the **product** was available in that **warehouse**.

The company wanted to modify the matrix in such a way that if a cell, **Mat[i][j]**, was **true**, then all cells in the **ith row** and **jth column** of the matrix would also be set to true. This would ensure that if a **product** was available in a particular **warehouse**, all the products in that **row** and **column** would also be considered available.

Can you write a program thet modify the matrix such that if a matrix cell **Mat[i][j]** is **1 (or true)** then make all the cells of **ith row** and **jth column** as **1**.

Sample Input 0

```
3
4
1 0 0 1
0 0 1 0
0 0 0 0
```

Sample Output 0

```
1 1 1 1
1 1 1 1
1 0 1 1
```

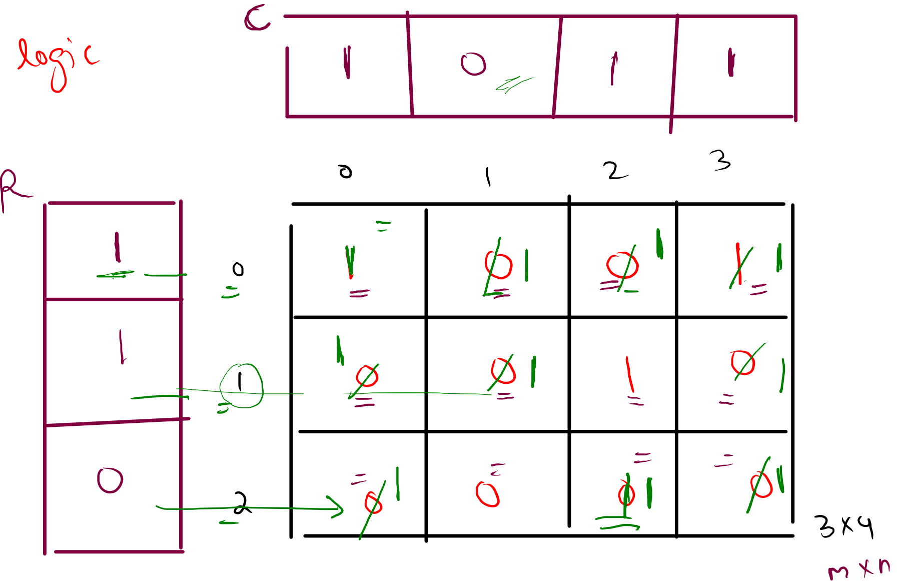
$$
\begin{array}{c|cccc}
 & 0 & 1 & 2 & 3 \\
\hline
0 & 1 = (0,0) & \cancel{0}\ 1 & \cancel{0}\ 1 & 1 = (0,3) \\
1 & \cancel{0}\ 1 & \cancel{0}\ 1 & 1 = (1,2) & \cancel{0}\ 1 \\
2 & \cancel{0}\ 1 & 0 & \cancel{0}\ 1 & \cancel{0}\ 1
\end{array}
$$

1111

1111

1011

logic



'2' 1D - Array.

1 → rows → R

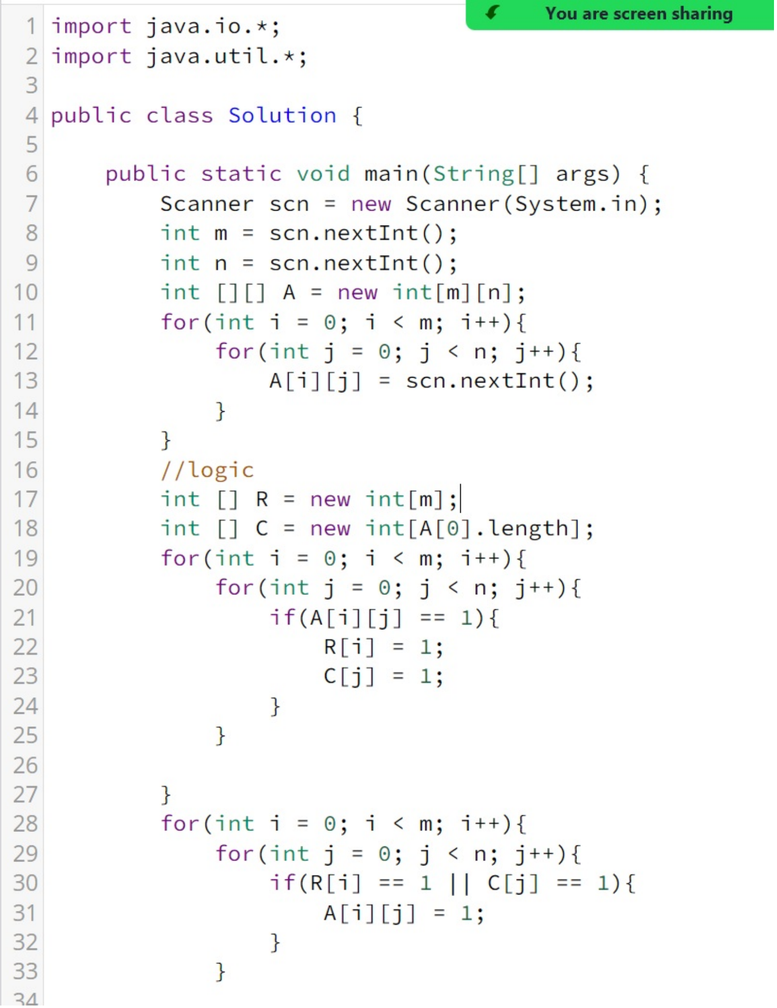
1 → cols. → C

i, j

0,0
0, 1
0, 2

```
if (A[i][j] == 1)
    R[i] = 1
    C[j] = 1
```

```
if (R[i] == 1 || C[j] == 1)
    A[i][j] = 1
```

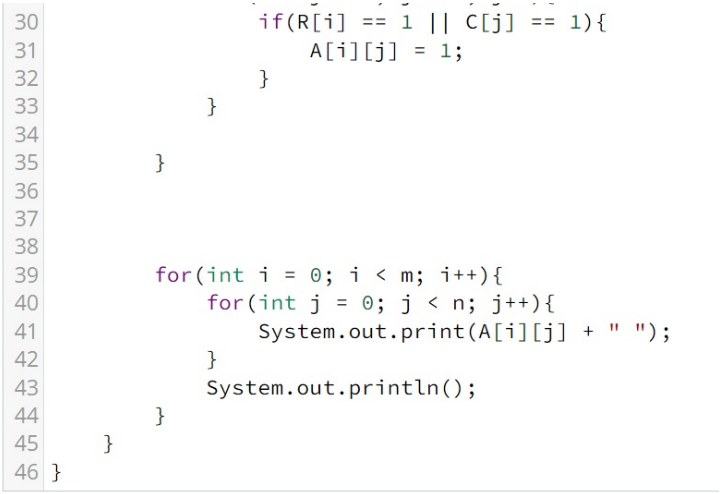


```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int m = scn.nextInt();
        int n = scn.nextInt();
        int [][] A = new int[m][n];
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                A[i][j] = scn.nextInt();
            }
        }
        //logic
        int [] R = new int[m];
        int [] C = new int[A[0].length];
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                if(A[i][j] == 1){
                    R[i] = 1;
                    C[j] = 1;
                }
            }

        }
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                if(R[i] == 1 || C[j] == 1){
                    A[i][j] = 1;
                }
            }
```



```java
                if(R[i] == 1 || C[j] == 1){
                    A[i][j] = 1;
                }
            }

        }



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