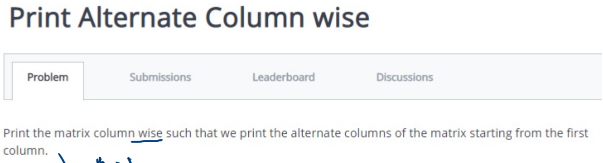
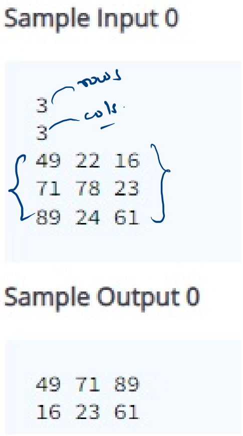


Print Alternate Column wise

Print the matrix column wise such that we print the alternate columns of the matrix starting from the first column.

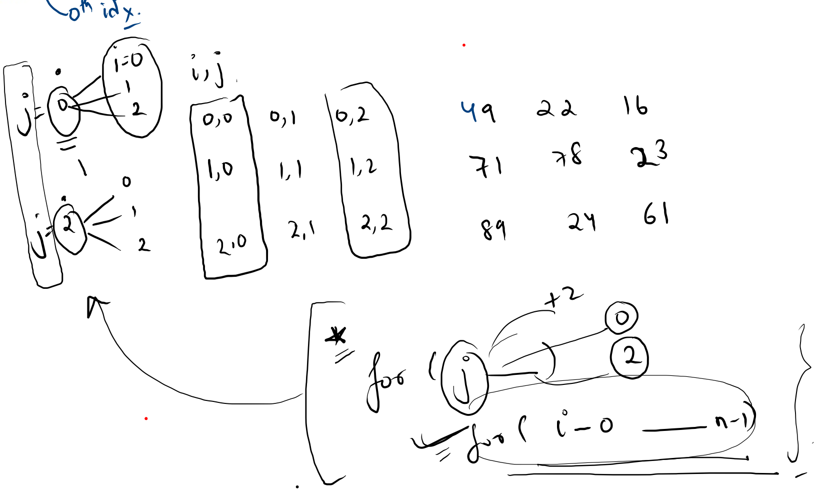


Sample Input 0

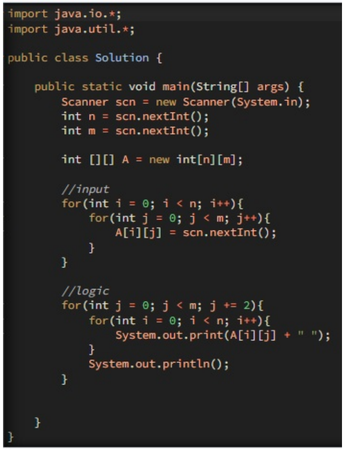
```
3
3
49 22 16
71 78 23
89 24 61
```

Sample Output 0

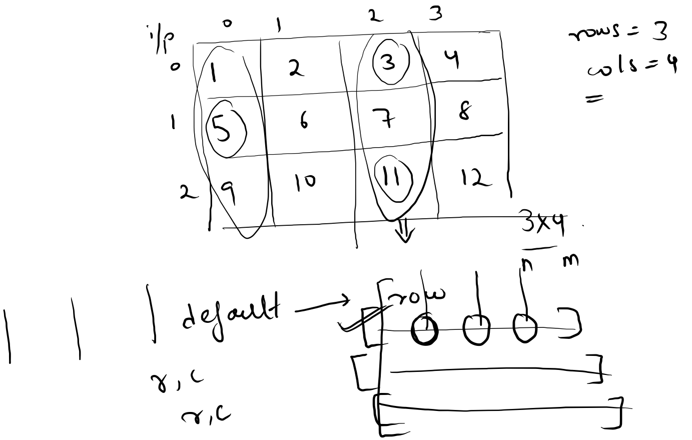
```
49 71 89
16 23 61
```



49	22	16
71	78	23
89	24	61

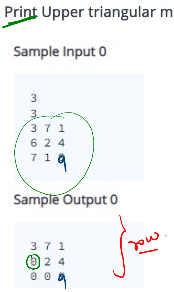


```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int m = scn.nextInt();

        int [][] A = new int[n][m];

        //input
        for(int i = 0; i < n; i++){
            for(int j = 0; j < m; j++){
                A[i][j] = scn.nextInt();
            }
        }

        //logic
        for(int j = 0; j < m; j += 2){
            for(int i = 0; i < n; i++){
                System.out.print(A[i][j] + " ");
            }
            System.out.println();
        }


    }
}
```



	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12

rows = 3
cols = 4



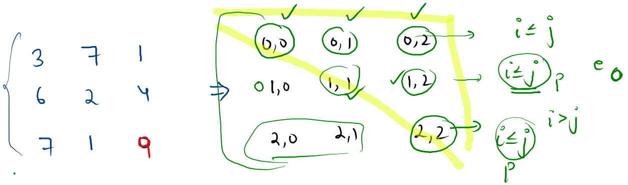
Print Upper triangular matrix 1

Sample Input 0

```
3
3
3 7 1
6 2 4
7 1 0
```

Sample Output 0

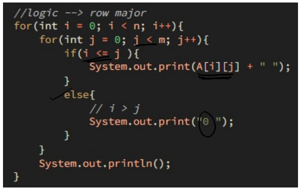
```
3 7 1
0 2 4
0 0 0
```



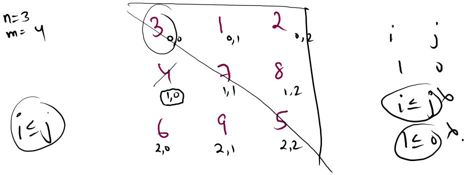
i>j

```
row
for (i=0 ⟶ n) }
   { for (j=0 ⟶ n) }
      {
        if (i≤j)
           pr
      }
}
```

else print 0



```
//logic --> row major
for(int i = 0; i < n; i++){
    for(int j = 0; j < m; j++){
        if(i <= j ){
            System.out.print(A[i][j] + " ");
        }
        else{
            // i > j
            System.out.print("0 ");
        }
    }
    System.out.println();
}
```

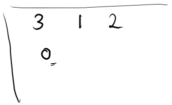


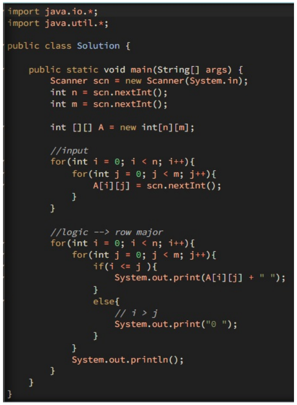
i=1 j=0 0<3 i≤j 1≤0 j=1

1 ≤ 0 1≤0 i>j

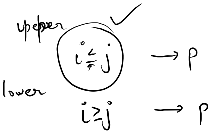
i=0 0<3 j=0 0<3 j=1 1<3 j=2 2<3

0≤0 3<3 0≤1 ✓ 0 ≤ 2 ✓ j=3

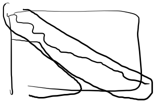


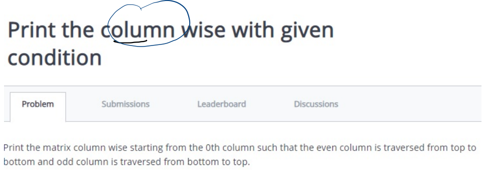


```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int m = scn.nextInt();

        int [][] A = new int[n][m];

        //input
        for(int i = 0; i < n; i++){
            for(int j = 0; j < m; j++){
                A[i][j] = scn.nextInt();
            }
        }

        //logic --> row major
        for(int i = 0; i < n; i++){
            for(int j = 0; j < m; j++){
                if(i <= j ){
                    System.out.print(A[i][j] + " ");
                }
                else{
                    // i > j
                    System.out.print("0 ");
                }
            }
            System.out.println();
        }
    }
}
```



i<j

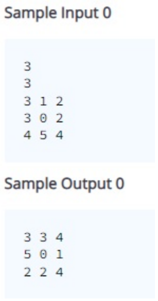




Print the column wise with given condition

Problem | Submissions | Leaderboard | Discussions

Print the matrix column wise starting from the 0th column such that the even column is traversed from top to bottom and odd column is traversed from bottom to top.



Sample Input 0

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

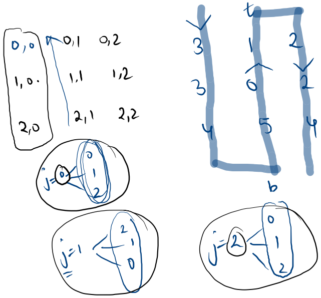
Sample Output 0

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

9:23

9:27-28

logic



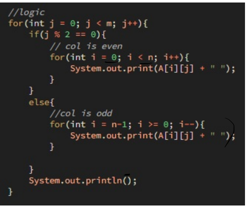
j=0 i=0 n

n 0

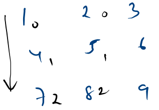
col → even → print t2B

↳ odd → " B2t

code ?



```
//logic
for(int j = 0; j < m; j++){
    if(j % 2 == 0){
        // col is even
        for(int i = 0; i < n; i++){
            System.out.print(A[i][j] + " ");
        }
    }
    else{
        //col is odd
        for(int i = n-1; i >= 0; i--){
            System.out.print(A[i][j] + " ");
        }

    }
    System.out.println();
}
```



2<3

j=0

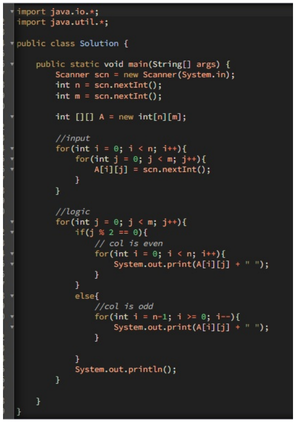
0<3

n=m =3

1 4 7

8 5 2

3 6 9



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

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int m = scn.nextInt();

        int [][] A = new int[n][m];

        //input
        for(int i = 0; i < n; i++){
            for(int j = 0; j < m; j++){
                A[i][j] = scn.nextInt();
            }
        }

        //logic
        for(int j = 0; j < m; j++){
            if(j % 2 == 0){
                // col is even
                for(int i = 0; i < n; i++){
                    System.out.print(A[i][j] + " ");
                }
            }
            else{
                //col is odd
                for(int i = n-1; i >= 0; i--){
                    System.out.print(A[i][j] + " ");
                }

            }
            System.out.println();
        }

    }
}
```

Compare Two Matrices

You have to take **two matrices** as input.

First Matrix of size **m1*n1**.

Second Matrix of size **m2*n2**.

Compare the two matrices and print **"Same"** if both the matrices are same else print **"Not Same"**.

m1 → row of 1st mat m2 → rows of 2nd

n1 → col of 1st mat n2 → cols of 2nd

$$\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \end{array} \longleftrightarrow \begin{array}{cc} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{array}$$

Not Same.

$m_1 \times n_1$ $\quad$ $m2 \times n2$

A $\quad$ 2×3 $\quad$ B $\quad$ 3×2

if $\left[\; m1 \neq m2 \;\; \text{or} \;\; n1 \neq n2 \right.$ ✓

$$\begin{array}{c|cccc} & & 0,1 & 0,2 & 0,3 \\ \hline 0,0 & 1 & 2 & 3 & 7 \\ & 4 & 5 & 6 & 2 \\ & 7 & 8 & 9 & 1 \end{array} \qquad \begin{array}{cccc} & 0,1 & 0,2 & 0,3 \\ 1 & 2 & 3 & 17 \\ 4 & 5 & 6 & 2 \\ 7 & 8 & 9 & 1 \end{array}$$

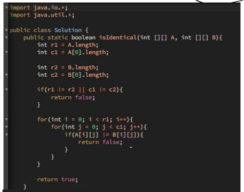
3×4 $\quad$ 3×4

$m1 \; n1$ $\quad$ $m2 \; n2$

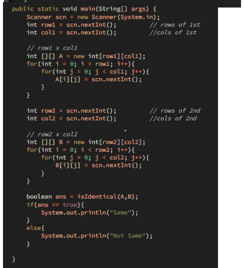
Not Same.

$m_1 = m_2$ ✓

$n_1 = n2$



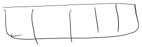
```
import java.io.*;
import java.util.*;

public class Solution {
    public static boolean isIdentical(int [][] A, int [][] B){
        int r1 = A.length;
        int c1 = A[0].length;

        int r2 = B.length;
        int c2 = B[0].length;

        if(r1 != r2 || c1 != c2){
            return false;
        }

        for(int i = 0; i < r1; i++){
            for(int j = 0; j < c1; j++){
                if(A[i][j] != B[i][j]){
                    return false;
                }
            }
        }

        return true;
    }
```



```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int row1 = scn.nextInt();          // rows of 1st
    int col1 = scn.nextInt();          //cols of 1st

    // row1 x col1
    int [][] A = new int[row1][col1];
    for(int i = 0; i < row1; i++){
        for(int j = 0; j < col1; j++){
            A[i][j] = scn.nextInt();
        }
    }

    int row2 = scn.nextInt();          // rows of 2nd
    int col2 = scn.nextInt();          //cols of 2nd

    // row2 x col2
    int [][] B = new int[row2][col2];
    for(int i = 0; i < row2; i++){
        for(int j = 0; j < col2; j++){
            B[i][j] = scn.nextInt();
        }
    }

    boolean ans = isIdentical(A,B);
    if(ans == true){
        System.out.println("Same");
    }
    else{
        System.out.println("Not Same");
    }

}
```

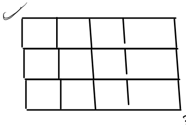
A. length

A[0]. length.

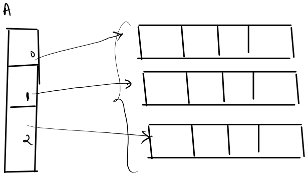


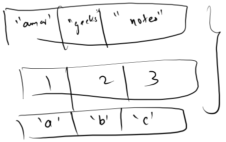
A.length → 5.

2D.

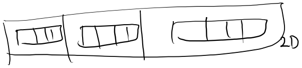


combination of 1D.

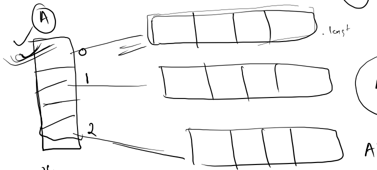




2D
↓
1D which can hold one 1D array at each cell



2D A(0). length



A[0]. ?

A[0]

3X4

A.length = 3 = rows.

A[0].length = 4 = cols.

A[1].length = 4 =

A[2].length

A[4]. length. Ion