Sophia is a talented data scientist working at a leading research institute. One day, she is tasked with analyzing a **matrix** of size **m * n** that contains crucial data for an upcoming project. As she delves into the **matrix**, Sophia realizes that the **alternate columns** hold key insights that could unlock the data's secrets.

your task is to **print** the **alternate columns** of the **matrix** starting from the **first column**.

Input Format

First line contains integers **m** and **n** representing the size of **matrix.**

Second line contains **m * n** integers representing elements of a **matrix.**

matrix

Constraints

```
1 <= m,n <=1000

-1000<=mat[i][j]<=1000
```

Output Format

Print Alternate columns of matrix.

Sample Input 0

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

Alternate Columns

Sample Output 0

```
1 3
4 6
7 9
```

Submitted Code

Language: Java 15

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

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int row = sc.nextInt();
        int col =sc.nextInt();

        int [][] mat = new int[row][col];
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
                mat[i][j]=sc.nextInt();
            }
        }
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j+=2){
                System.out.print(mat[i][j]+" ");
            }
            System.out.println();
        }
    }
}
```

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

Input Format

1. First line contains, m and n depicting the size of first matrix.

2. m*n Integer values, depicting all the elements of matrix.

Constraints

- 1 <= m1 and n1 <=1000
- -1000<=mat[i][j]<=1000

Output Format

Print the matrix column wise alternate

Sample Input 0

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

Sample Output 0

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

Explanation 0

we will print column 1 and 3 becuase we skip 2. And then print them as rows.

Language: Java 15

1st approach

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

public class Solution {

    public static void main(String[] args) {
         Scanner sc = new Scanner(System.in);
        int row = sc.nextInt();
        int col =sc.nextInt();

        int [][] mat = new int[row][col];
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
                mat[i][j]=sc.nextInt();
            }
        }

        int trans [][] = new int [col][row];
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
                trans[j][i]= mat[i][j];
            }
        }

        for(int i=0;i<col;i+=2){
            for(int j=0;j<row;j++){
                System.out.print(trans[i][j]+" ");
            }
            System.out.println();
        }
    }
}
```

Submitted Code

Language: Java 15

2nd approach

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

public class Solution {

    public static void main(String[] args) {
         Scanner sc = new Scanner(System.in);
        int row = sc.nextInt();
        int col =sc.nextInt();

        int [][] mat = new int[row][col];
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
                mat[i][j]=sc.nextInt();
            }
        }
        for(int i=0;i<col;i+=2){
            for(int j=0;j<row;j++){
                System.out.print(mat[j][i]+" ");
            }
            System.out.println();
        }
    }
}
```

John is a software engineer working for a leading tech company. One day, he is given a task to analyze a complex **matrix** of size **m** and **n** that contains crucial data for an upcoming project. As he dives into the **matrix**, John realizes that the **lower triangle** holds important insights that could unlock the data's secrets.

Help John and **print** the **lower triangular matrix.**

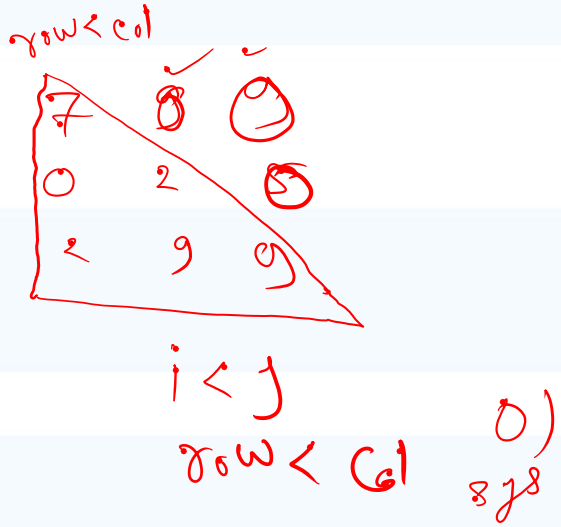
Input Format

First line contains integers **m** and **n** depicting the size of **matrix.**

Second line contains **m * n** Integer values, depicting all the elements of **matrix.**

Constraints

```
1<=m and n<=1000
-1000<=mat[i][j]<=1000
```



Output Format

Print the Lower triangular matrix

Sample Input 0

```
3
3
7 9 9
0 2 5
2 9 9
```

Sample Output 0

```
7 0 0
0 2 0
2 9 9
```

Submitted Code

Language: Java 15

```
1  import java.io.*;
2  import java.util.*;
3
4  public class Solution {
5
6      public static void main(String[] args) {
7           Scanner sc = new Scanner(System.in);
8          int row = sc.nextInt();
9          int col =sc.nextInt();
10
11         int [][] mat = new int[row][col];
12         for(int i=0;i<row;i++){
13             for(int j=0;j<col;j++){
14                 mat[i][j]=sc.nextInt();
15             }
16         }
17
18         for(int i=0;i<row;i++){
19             for(int j=0;j<col;j++){
20                 if(i<j)System.out.print(0+" ");
21                 else System.out.print(mat[i][j]+" ");
22             }System.out.println();
23         }
24     }
25
26 }
```