

IRC\_JAVA\_COD\_ARRAYS2D

Test Summary

- No. of Sections: 1
- No. of Questions: 15
- Total Duration: 180 min

Section 1 - Coding

Section Summary

- No. of Questions: 15
- Duration: 180 min

Additional Instructions:

None

Q1. Write a program to find the normal of a matrix.

Input Format

The first line of the input consists of the value of n (The number of rows and the number of columns).  
The next input is the matrix.

Output Format

The output prints the normal of the given matrix.

Sample Input

```
5
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
```

Sample Output

```
16
```

Sample Input

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

Sample Output

```
38
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2. Write a program to implement matrix multiplication.

Input Format

**Note: Square matrices only.**

The first line of the input is the value of n - order of matrices.  
Next input is the array elements.

Output Format

The output prints the product of two matrices.

Sample Input

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

Sample Output

```
33 28 35
75 70 83
47 42 51
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q3. Write a program to subtract two matrices.

Input Format

The first line of the input consists of n.

Next input is the two n\*n matrices.

Output Format

The output prints the result (Matrix 1 - Matrix 2).

Sample Input

```
4
19 28 27 26
22 31 43 59
10 22 42 58
```

Sample Output

```
6 3 11 -41
-7 14 15 54
-15 -11 30 15
1 72 26 8
```

Sample Input

```
2
5 -9
8 2
4 7
```

Sample Output

```
1 -7
9 -1
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q4. Write a program to add two matrices.

Input Format

Input to get number of rows and columns separated by single space, followed by values of r\*c matrix as shown in the sample input.

Output Format

Output the resultant matrix after addition as shown in sample test cases.

Constraints

Values(Integer type).

Sample Input

```
3 3
5 6 7
7 8 9
0 7 8
```

Sample Output

```
6 6 8
8 9 10
9 7 8
```

Sample Input

```
2 2
5 -9
8 2
4 7
```

Sample Output

```
9 -11
7 5
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q5. Write a program to obtain a matrix and find the sum of its diagonal elements.  
Note: Only square matrix.

Input Format

The input consists of the number of rows and columns separated by a space.  
The next input is the matrix.

Output Format

The output prints the sum of diagonal elements.

Sample Input

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

Sample Output

```
15
```

Sample Input

```
4 4
12 23 45 56
78 89 98 87
65 54 32 21
```

Sample Output

```
191
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q6. Write a program to obtain a matrix and find the sum of the elements in the lower triangular matrix(i.e., the elements on the diagonal and the lower elements).

Note: Only square matrix

Input Format

The first line of the input consists of the number of rows and columns.  
The next input is the matrix.

Output Format

The output prints the sum of the upper triangular matrix.

Sample Input

```
3 3
6 5 4
1 2 5
7 0 7
```

Sample Output

```
32
```

Sample Input

```
3 3
12 23 45
56 78 89
05 51 20
```

Sample Output

```
312
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q7. Raja was arranging gift boxes in his store. He decided to arrange the gift boxes in an order so that the number of gift boxes in each row is equal to the row number number. Each gift box is numbered in ascending order. He imagined how the arrangement would be.  
Write a Java program to print such pattern.

Input Format

Input consists of one integer describing the number of rows.

Output Format

Output consists of the described pattern.

Sample Input

```
5
```

Sample Output

```
[1]
[2, 3]
[4, 5, 6]
[7, 8, 9, 10]
[11, 12, 13, 14, 15]
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q8. **Snake Pattern:**  
Write a program to print the matrix in a zig\_zag form.

Input Format

Input to get the integer R and C (i.e,rows and columns) in the first line separated by a single space,then get the values for R\*C matrix in the following lines.

Output Format

The output displays the matrix in snake pattern.

Sample Input

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

Sample Output

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

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q9. Write a program to obtain a matrix and find the sum of each row and each column.

Input Format

The first line of the input consists of the value of the number of rows and the number of columns.  
The next input is the matrix.

Output Format

The output prints the sum of each row and each column.  
Refer sample input and output for formatting specifications.

Sample Input

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

Sample Output

```
Sum of the row 0 = 10
Sum of the row 1 = 26
Sum of the row 2 = 42
Sum of the row 3 = 58
```

Sample Input

```
3 3
98 87 65
54 32 21
45 56 59
```

Sample Output

```
Sum of the row 0 = 250
Sum of the row 1 = 107
Sum of the row 2 = 159
Sum of the column 0 = 197
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q10. Write a program to obtain a matrix and find the sum of the elements in the upper triangular matrix(i.e., the elements on the diagonal and the upper elements).  
Note: Only square matrix

Input Format

The first line of the input consists of the number of rows and columns.  
The next input is the matrix.

Output Format

The output prints the sum of the upper triangular matrix.

Sample Input

```
3 3
6 5 4
1 2 5
7 0 7
```

Sample Output

```
29
```

Sample Input

```
3 3
12 23 45
56 78 89
05 51 20
```

Sample Output

```
267
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q11. **Collisions of Events**  
Lucarnos Film Festival is an annual film festival and is also known for being a prestigious platform for art house films. This year at the Lucarnos Film festival there are many movies to be screened, each of different genre ranging from drama movies to comedy ones and teen movies to horror ones. The festival is a long-running event this time as the organizers are planning to screen only one movie per day. The organizers have populated their schedule in the form of a matrix where 'i' is the day number and 'j' is the movie number. Eij is the movie preference dates.  
You are given a matrix E of N rows and M columns where Eij is 1 if the i-th movie is to be screened on j-th day, otherwise it will be 0. Note that it is not necessary that if a movie x will be screened on day y, then day y should screen only movie x.  
You know that if there are two different movies x and y, which are to be screened on the same day z, and then there will be a collision. Can you calculate the number of different collisions at this movie festival? Note that order of movies in the collision doesn't matter.

Input Format

The first line of the input contains two space separated integers N, M denoting the number of movies and days, respectively.  
Each of the following N lines contain M characters, each of them is either '0' or '1'.

Output Format

Output a single line containing an integer corresponding to the number of collisions at the film festival.  
Refer sample input and output for formatting specifications.

Sample Input

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

Sample Output

```
4
```

Sample Input

```
2 2
1 0
```

Sample Output

```
0
```

01

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q12.       Johnsy wants to create a matrix in which the elements are formed differently. The elements are formed by adding the values of their index positions. Write a program that obtains the order of the matrices and creates a matrix by adding the values of their index positions.

Input Format

The input line consists of the order of the matrices m and n separated by a space.

Output Format

The output prints the matrix elements in matrix format. The elements are formed by adding the values of their index positions.  
Give a tab space between the elements.  
Refer sample input and output for formatting specifications.

Sample Input

3 3

Sample Output

012

123

234

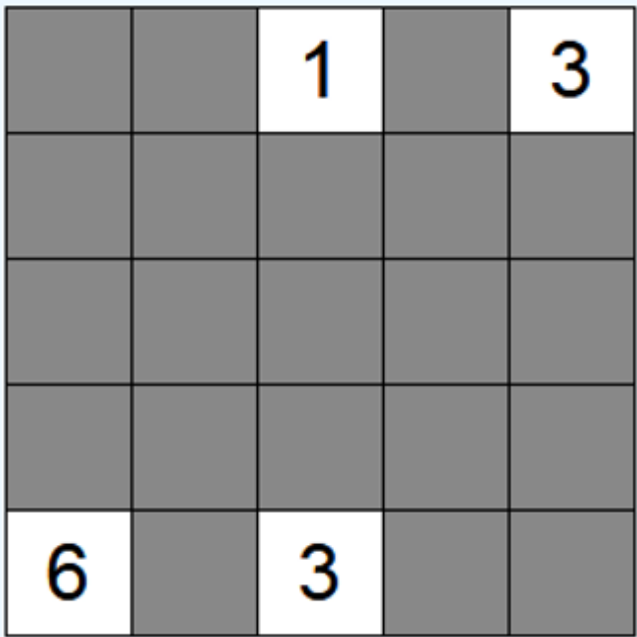
Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q13.       **Valid Initial Configuration**  
Nurikabe logical game (sometimes called Islands in the Stream) is a binary determination puzzle. The puzzle is played on a typically rectangular grid of cells, some of which contain numbers. You must decide for each cell if it is white or black (by clicking on them) according to the following rules:

- All of the black cells must be connected.
- Each numbered cell must be part of a white island of connected white cells.
- Each island must have the same number of white cells as the number it contains (including the numbered cell).
- Two islands may not be connected.
- There cannot be any 2x2 blocks of black cells.

Unnumbered cells start out grey and cycle through white and black when clicked. Initially numbered cells are white in color.

**Problem Statement:**  
Write a program to check whether the given board configuration is a valid initial configuration. Below figure is the sample valid initial configuration.



Input Format

First line of the input is an integer N that gives the number of rows and columns of the grid.  
Next N lines will have the board configuration with N\*N cells. Assume that the maximum number in a cell can be 10. Grey colored cells are represented by the integer 20 in the matrix representation of the input configuration.

Output Format

Output "Yes" (without quotes) if the given configuration is a valid initial configuration. Print "No" otherwise (without quotes).  
Refer sample input and output for formatting specifications.

Sample Input

5  
20 20 1 20 3  
20 20 20 20 20  
20 20 20 20 20

Sample Output

Yes

Sample Input

Sample Output

5 20 20 1 20 3 20 20 20 20 20 20 20 12 20 20	No
---	----

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q14. **Mid Aged**  
The Pan Am 73 flight from Bombay to New York en route Karachi and Frankfurt was hijacked by a few Palestinian terrorists at the Karachi International Airport. The senior flight purser Neerja Banhot withered her fear and helped evacuating the passengers on board.



Neerja very well knew that she would not be able to evacuate all passengers dodging the hijackers. So she wanted to hand over the responsibility of evacuating in the senior citizens(above 60 years of age) and children(below 18 years of age) in the flight to the mid-aged passengers seated in the diagonals.

Given n the number of rows of seats and the number of seats in a row and the ages of passengers in each seat can you find the number of mid-aged passengers seated in the main diagonals.

**Input Format**

The first line of input consists of an integer n, corresponding to the number of rows of seats and the number of seats in the aircraft. The next n lines of input consist of n integers that correspond to the ages of passengers.

**Output Format**

The output consists of an integer corresponding to the number of mid-aged passengers seated in the diagonals.

Sample Input	Sample Output
3 21 3 44 78 25 19 50 22 6	2

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q15. Seetha, a maths teacher explained about Matrix addition, subtraction and multiplication in her class. She assigned different task to her students.

She asked Ankit to add two matrix, Banu to subtract two matrix and Janu to multiply two matrix. Ankit, Banu and Janu approached Karthick to complete their task.

Karthick is ready to help all his friends with single program. So he asked his friends to give a square matrix only. Help Karthick to write the program.

**Input Format**

Matrix size in first line -N(Only one value)  
NxN elements in next rows for first matrix  
NxN elements in next rows for second matrix

Refer Sample Input

**Output Format**

Display result of addition , subtraction and multiplication as shown in sample output

Constraints

Only Square Matrix

Sample Input

```
3

1 2 3
1 2 2
```

Sample Output

```
Sum
4 4 4
4 4 4
4 4 4
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Answer Key & Solution

Section 1 - Coding

Q1

Test Case

Input

```
5
1 1 1 1 1
2 2 2 2 2
2 2 2 2 2
```

Output

```
16
```

Weightage - 25

Input

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

Output

```
38
```

Weightage - 25

Input

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

Output

```
16
```

Weightage - 25

Input

```
2
1 2
1 2
```

Output

```
3
```

Weightage - 25

Sample Input

```
5
1 1 1 1 1
2 2 2 2 2
2 2 2 2 2
```

Sample Output

```
16
```

Sample Input

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

Sample Output

```
38
```

Solution

```
import java.util.Scanner;
class normal {
static int MAX = 100;
static int findNormal(int mat[][], int n)
{
    int sum = 0;
    for (int i=0; i<n; i++)
        for (int j=0; j<n; j++)
```



```
        sum += mat[i][j]*mat[i][j];
    return (int)Math.sqrt(sum);
}

public static void main (String[] args) {
    int i,j,n;
    Scanner sc = new Scanner(System.in);
    n = sc.nextInt();
    int [][] mat = new int[n][n];
    for(i=0;i<n;i++) {
        for(j=0;j<n;j++) {
            mat[i][j] = sc.nextInt();
        }
    }
    System.out.println (findNormal(mat, n));
}
```

Q2

Test Case

Input

```
3
10 20 30
42 51 53
71 85 96
```

Output

```
3900 3140 2840
7887 6892 6802
13893 12010 11710
```

Weightage - 25

Input

```
4
12 23 45 56
14 25 36 35
17 58 69 15
```

Output

```
3489 6745 7085 8873
2822 5682 6196 7490
5539 11776 13456 15784
8112 17185 20188 21011
```

Weightage - 25

Input

```
3
12 45 78
23 56 89
25 36 14
```

Output

```
5424 8112 9267
7250 10345 11621
4469 5053 5141
```

Weightage - 25

Input

```
4
10 20 30 40
50 60 70 80
10 20 30 40
```

Output

```
3740 4840 5940 7040
9020 11880 14740 17600
3740 4840 5940 7040
9020 11880 14740 17600
```

Weightage - 25

Sample Input

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

Sample Output

```
33 28 35
75 70 83
47 42 51
```

Solution

```
import java.io.*;
import java.util.*;
class Main {
    public static void main(String [] args) {
        int n,i,j,k;
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        int mat1[][] = new int[n][n];
        int mat2[][] = new int[n][n];
        int m[][] = new int[n][n];
        for(i=0;i<n;i++) {
            for(j=0;j<n;j++) {
                mat1[i][j] = sc.nextInt();
            }
        }
        for(i=0;i<n;i++) {
            for(j=0;j<n;j++) {
                mat2[i][j] = sc.nextInt();
            }
        }
        for(i=0;i<n;i++) {
            for(j=0;j<n;j++) {
                m[i][j] =0;
                for(k=0;k<n;k++) {
                    m[i][j] += mat1[i][k]*mat2[k][j];
                }
            }
        }
        for(i=0;i<n;i++) {
            for(k=0;k<n;k++) {
                System.out.print(m[i][k]+" ");
            }
            System.out.println();
        }
    }
}
```

Q3 **Test Case**

**Input**

3  
5 6 7  
8 7 9  
2 2 1

**Output**

4 4 6  
2 4 5  
-6 -5 1

**Weightage - 20**

**Input**

5  
6 7 8 9 9  
19 26 45 32 67  
15 17 18 21 22

**Output**

-22 -22 -57 -69 0  
-34 11 29 15 49  
-11 -11 -3 -8 13  
11 25 20 82 1

**Weightage - 20**

**Input**

2  
5 6  
7 8  
9 9

**Output**

-4 6  
-3 -4

Weightage - 20

Input

Output

```
6
309 989 900 675 786 897
200 234 561 678 453 222
224 200 212 100 104 766
```

```
234 901 710 588 721 874
166 192 496 600 410 191
303 172 -112 -100 -108 666
4 20 10 688 22 682
```

Weightage - 20

Input

Output

```
4
10 14 16 17
18 17 19 13
22 21 25 26
```

```
-20 -20 -16 -19
-15 -24 -9 -7
1 -3 -1 -1
2 2 224 24
```

Weightage - 20

Sample Input

Sample Output

```
4
19 28 27 26
22 31 43 59
10 22 42 58
```

```
6 3 11 -41
-7 14 15 54
-15 -11 30 15
1 22 26 8
```

Sample Input

Sample Output

```
2
5 -9
8 2
4 2
```

```
1 -7
9 -1
```

Solution

```
import java.util.Scanner;
class Main
{
    public static void main(String args[])
    {
        int n,i,j;
        Scanner scan = new Scanner(System.in);
        n = scan.nextInt();
        int mat1[][] = new int[n][n];
        int mat2[][] = new int[n][n];
        int mat3[][] = new int[n][n];
        //System.out.println("First matrix :");
        for(i=0; i<n; i++)
        {
            for(j=0; j<n; j++)
            {
                mat1[i][j] = scan.nextInt();
            }
        }

        //input matrix 2
        //System.out.println("Second matrix :");
        for(i=0; i<n; i++)
        {
            for(j=0; j<n; j++)
            {
                mat2[i][j] = scan.nextInt();
            }
        }
    }
}
```

```
for(i=0; i<n; i++)
{
    for(j=0; j<n; j++)
    {
        mat3[i][j] = mat1[i][j] - mat2[i][j];
    }
}
for(i=0; i<n; i++)
{
    for(j=0; j<n; j++)
    {
        System.out.print(mat3[i][j]+ " ");
    }
    System.out.println();
}
}
```

Q4 **Test Case**

Input

```
3 3
-2 -3 -9
-10 -9 -8
15 12 10
```

Output

```
-4 -10 -18
-18 -20 -17
-18 -12 -30
```

Weightage - 20

Input

```
2 2
5 6
7 8
0 0
```

Output

```
14 6
17 20
```

Weightage - 20

Input

```
4 4
309 989 900 675
200 234 561 678
224 200 212 100
```

Output

```
365 1067 990 740
223 279 626 699
414 287 277 123
172 864 122 120
```

Weightage - 20

Input

```
2 2
3 4
5 6
0 0
```

Output

```
12 4
7 10
```

Weightage - 20

Input

```
2 2
1 0
1 1
0 0
```

Output

```
1 0
1 2
```

Weightage - 20

Sample Input

Sample Output

3 3  
5 6 7  
7 8 9  
9 7 8

6 6 8  
8 9 10  
9 7 8

Sample Input

Sample Output

2 2  
5 -9  
8 2  
4 7

9 -11  
7 5

Solution

```
import java.util.Scanner;
class Main
{
    public static void main(String args[])
    {
        int m, n, c, d;
        Scanner in = new Scanner(System.in);
        m = in.nextInt();
        n = in.nextInt();
        int first[][] = new int[m][n];
        int second[][] = new int[m][n];
        int sum[][] = new int[m][n];
        for ( c = 0 ; c < m ; c++ )
            for ( d = 0 ; d < n ; d++ )
                first[c][d] = in.nextInt();

        for ( c = 0 ; c < m ; c++ )
            for ( d = 0 ; d < n ; d++ )
                second[c][d] = in.nextInt();

        for ( c = 0 ; c < m ; c++ )
            for ( d = 0 ; d < n ; d++ )
                sum[c][d] = first[c][d] + second[c][d];
        for ( c = 0 ; c < m ; c++ )
        {
            for ( d = 0 ; d < n ; d++ )
                System.out.print(sum[c][d]+" ");

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

Q5

Test Case

Input

Output

3 3  
1 2 3  
4 5 6  
7 8 9

15

Weightage - 25

Input

Output

4 4 12 23 45 56 78 89 98 87 65 54 32 21	191
--	-----

Weightage - 25

Input

Output

2 2 74 85 96 52	126
-----------------------	-----

Weightage - 25

Input

Output

1 1 25	25
-----------	----

Weightage - 25

Sample Input

Sample Output

3 3 1 2 3 4 5 6 7 8 9	15
--------------------------------	----

Sample Input

Sample Output

4 4 12 23 45 56 78 89 98 87 65 54 32 21	191
--	-----

Solution

```
import java.util.Scanner;
class disgonalSum
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);

        int i,j,row,col,sum=0;
        row = sc.nextInt();
        col = sc.nextInt();

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

        for(i=0;i<row;i++)
        {
            for(j=0;j<col;j++)
            {
                if(i==j)
```

```
        {
            sum = sum + mat[i][j];
        }
    }

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

Q6

Test Case

Input

3 3  
6 5 4  
1 2 5  
7 0 7

Output

32

Weightage - 25

Input

3 3  
12 23 45  
56 78 89  
05 51 20

Output

312

Weightage - 25

Input

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

Output

100

Weightage - 25

Input

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

Output

235

Weightage - 25

Sample Input

3 3  
6 5 4  
1 2 5  
7 0 7

Sample Output

32

Sample Input

3 3  
12 23 45  
56 78 89  
05 51 20

Sample Output

312

Solution

```
import java.util.Scanner;
class matrix
{
    static void sum(int mat[][], int r, int c)
    {
        int i, j;
        int lower_sum = 0;
        for (i = 0; i < r; i++)
            for (j = 0; j < c; j++)
            {
                if (j <= i)
                {
                    lower_sum += mat[i][j];
                }
            }

        System.out.print(lower_sum);
    }
    public static void main (String[] args)
    {
        int r,c,i,j;
        Scanner sc = new Scanner(System.in);
        r = sc.nextInt();
        c = sc.nextInt();
        int [][] mat = new int[r][c];
        for(i=0;i<r;i++) {
            for(j=0;j<c;j++) {
                mat[i][j] = sc.nextInt();
            }
        }
        sum(mat, r, c);
    }
}
```

Q7 **Test Case**

**Input**

23

**Output**

[1]  
[2, 3]  
[4, 5, 6]

**Weightage - 15**

**Input**

50

**Output**

[1]  
[2, 3]  
[4, 5, 6]

**Weightage - 15**

**Input**

25

**Output**

[1]  
[2, 3]  
[4, 5, 6]

**Weightage - 15**



Input

7

Output

[1]  
[2, 3]  
[4, 5, 6]  
[7, 8, 9, 10]

Weightage - 15

Input

36

Output

[1]  
[2, 3]  
[4, 5, 6]

Weightage - 15

Input

29

Output

[1]  
[2, 3]  
[4, 5, 6]

Weightage - 15

Input

1

Output

[1]

Weightage - 10

Sample Input

5

Sample Output

[1]  
[2, 3]  
[4, 5, 6]  
[7, 8, 9, 10]

Solution

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

class Main
{
    public static void main(String[] a) throws Exception
    {
        Scanner input = new Scanner(System. in);
        int N = input. nextInt();
        if (N <= 0) return;
        int array[][] = new int[N][];
        for (int ctr = 0; ctr < N; ctr++) {
            array[ctr] = new int[ctr + 1];
        }
        int val = 1;
        for (int ctr_row = 0; ctr_row < N; ctr_row++) {
            for (int ctr_col = 0; ctr_col < ctr_row + 1; ctr_col++) {
                array[ctr_row][ctr_col] = val++;
            }
        }
    }
}
```

```
    }  
    for (int ctr = 0; ctr < N; ctr++) {  
        System.out.println(Arrays.toString(array[ctr]));  
    }  
}  
}
```

Q8

Test Case

Input

```
5 3  
3 4 5  
7 8 9  
23 45 67
```

Output

```
3 4 5 9 8 7 23 45 67 0 99 88 12 15 16
```

Weightage - 20

Input

```
2 2  
34 37  
45 78
```

Output

```
34 37 78 45
```

Weightage - 20

Input

```
3 2  
5 6  
89 87  
45 67
```

Output

```
5 6 87 89 45 67
```

Weightage - 20

Input

```
4 4  
12 34 56 78  
98 76 54 90  
22 33 44 55
```

Output

```
12 34 56 78 90 54 76 98 22 33 44 55 99 88 77 66
```

Weightage - 20

Input

```
2 3  
678 908 234  
102 245 778
```

Output

```
678 908 234 778 245 102
```

Weightage - 20

Sample Input

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

Sample Output

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

Solution

```
import java.util.Scanner;
class Main
{
    static void printZigZag(int row, int col, int a[][])
    {
        int evenRow = 0;
        int oddRow = 1;

        while (evenRow < row)
        {
            for (int i = 0; i < col; i++)
            {
                System.out.print(a[evenRow][i] + " ");
            }
            evenRow = evenRow + 2;

            if(oddRow < row)
            {
                for (int i = col - 1; i >= 0; i--)
                {
                    System.out.print(a[oddRow][i] + " ");
                }
                oddRow = oddRow + 2;
            }
        }
    }

    public static void main(String[] args)
    {

        int r, c;
        int i,j;
        Scanner in=new Scanner(System.in);
        r=in.nextInt();
        c=in.nextInt();
        int mat[][]=new int[10][10];
        for(i=0;i<r;i++){
            for(j=0;j<c;j++){
                mat[i][j]=in.nextInt();
            }
        }
        printZigZag(r , c , mat);
    }
}
```

Q9 **Test Case**

**Input**

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

**Output**

Sum of the row 0 = 10  
Sum of the row 1 = 26  
Sum of the row 2 = 42  
Sum of the row 3 = 58

**Weightage - 25**

**Input**

3 3  
98 87 65  
54 32 21  
45 56 58

**Output**

Sum of the row 0 = 250  
Sum of the row 1 = 107  
Sum of the row 2 = 159  
Sum of the column 0 = 197

Weightage - 25

Input

```
2 2
20 10
40 50
```

Output

```
Sum of the row 0 = 30
Sum of the row 1 = 90
Sum of the column 0 = 60
Sum of the column 1 = 60
```

Weightage - 25

Input

```
5 5
111 222 333 444 555
666 777 888 999 1000
11 22 33 44 55
```

Output

```
Sum of the row 0 = 1665
Sum of the row 1 = 4330
Sum of the row 2 = 165
Sum of the row 3 = 120
```

Weightage - 25

Sample Input

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

Sample Output

```
Sum of the row 0 = 10
Sum of the row 1 = 26
Sum of the row 2 = 42
Sum of the row 3 = 58
```

Sample Input

```
3 3
98 87 65
54 32 21
45 56 58
```

Sample Output

```
Sum of the row 0 = 250
Sum of the row 1 = 107
Sum of the row 2 = 159
Sum of the column 0 = 197
```

Solution

```
import java.util.Scanner;
class matrix {
    static void row_sum(int arr[][],int m, int n)
    {

        int i, j, sum = 0;
        for (i = 0; i < m; ++i) {
            for (j = 0; j < n; ++j) {
                sum = sum + arr[i][j];
            }
            System.out.println("Sum of the row " + i + " = "
                               + sum);
            sum = 0;
        }
    }
    static void column_sum(int arr[][], int m,int n)
    {

        int i, j, sum = 0;
        for (i = 0; i < m; ++i) {
            for (j = 0; j < n; ++j) {
                sum = sum + arr[j][i];
            }
            System.out.println("Sum of the column " + i
                               + " = " + sum);
            sum = 0;
        }
    }
    public static void main(String[] args)
```

```
{
    int i, j,m,n;
    Scanner sc = new Scanner(System.in);
    m = sc.nextInt();
    n = sc.nextInt();
    int[][] arr = new int[m][n];
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            arr[i][j] = sc.nextInt();
    row_sum(arr,m,n);
    column_sum(arr,m,n);
}
```

Q10 **Test Case**

Input

3 3  
6 5 4  
1 2 5  
7 0 7

Output

29

Weightage - 25

Input

3 3  
12 23 45  
56 78 89  
05 51 20

Output

267

Weightage - 25

Input

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

Output

70

Weightage - 25

Input

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

Output

155

Weightage - 25

Sample Input

3 3  
6 5 4  
1 2 5  
7 0 7

Sample Output

29

Sample Input

3 3  
12 23 45  
56 78 89  
05 51 20

Sample Output

267

Solution

```
import java.util.Scanner;
class matrix
{
    static void sum(int mat[][], int r, int c)
    {
        int i, j;
        int upper_sum = 0;
        for (i = 0; i < r; i++)
            for (j = 0; j < c; j++)
            {
                if (i <= j)
                {
                    upper_sum += mat[i][j];
                }
            }

        System.out.println(upper_sum);
    }
    public static void main (String[] args)
    {
        int r,c,i,j;
        Scanner sc = new Scanner(System.in);
        r = sc.nextInt();
        c = sc.nextInt();
        int [][] mat = new int[r][c];
        for(i=0;i<r;i++) {
            for(j=0;j<c;j++) {
                mat[i][j] = sc.nextInt();
            }
        }
        sum(mat, r, c);
    }
}
```

Q11 Test Case

Input

4 3

1 1 1

1 0 0

Output

4

Weightage - 10

Input

2 2

1 0

0 1

Output

0

Weightage - 10

Input

3 4

1 0 1 0

1 1 0 0

1 0 1 0

Output

4

Weightage - 10

Input

Output

6 6	17
1 0 1 0 1 0	
0 0 1 1 0 0	
0 1 0 1 0 1	

Weightage - 15

Input

Output

8 6	36
1 0 1 0 1 0	
0 1 0 1 0 1	
1 1 0 0 1 1	

Weightage - 15

Input

Output

10 8	83
1 0 1 0 1 0 1 0	
0 1 0 1 0 1 0 1	
0 1 1 0 1 0 0 1	

Weightage - 20

Input

Output

15 8	198
1 0 1 0 1 0 1 0	
0 1 0 1 0 1 0 1	
0 1 1 0 1 0 0 1	

Weightage - 20

Sample Input

Sample Output

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

Sample Input

Sample Output

2 2	0
1 0	
0 1	

Solution

```
import java.io.*;
import java.util.*;
class collisionsOfEvents {
    public static void main(String [] args) {
        int n,m,i,j,k=0,c=0;
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        m = sc.nextInt();
        int a[][] = new int[n][m];
        for(i=0;i<n;i++)
        {
```

```
        for(j=0;j<m;j++)
        {
            a[i][j] = sc.nextInt();
        }
    }
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            if(a[j][i]==1){c++;}
        }
        if(c>2){k=k+((c-1)*(c)/2);}
        else if(c==2){k++;}
        c=0;
    }
    System.out.println(k);
}
}
```

Q12 **Test Case**

**Input**

4 4

**Output**

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

**Weightage - 10**

**Input**

5 5

**Output**

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

**Weightage - 10**

**Input**

6 6

**Output**

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

**Weightage - 15**

**Input**

7 7

**Output**

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

**Weightage - 15**

**Input**

8 8

**Output**

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



Weightage - 15

Input

99

Output

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

Weightage - 15

Input

1010

Output

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

Weightage - 20

Sample Input

33

Sample Output

0	1	2
1	2	3
2	3	4

Solution

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

class Q01Simple2DArray
{
    public static void main (String[] args) throws java.lang.Exception
    {
        Scanner input = new Scanner(System. in);
        int N = input. nextInt();
        if (N <= 0) return;
        int M = input. nextInt();
        if (M <= 0) return;
        int numbers[][] = new int[N][M];
        for (int ctr_row = 0; ctr_row < N; ctr_row++) {
            for (int ctr_col = 0; ctr_col < M; ctr_col++) {
                numbers[ctr_row][ctr_col] = ctr_row + ctr_col;
            }
        }
        for (int ctr_row = 0; ctr_row < N; ctr_row++) {
            for (int ctr_col = 0; ctr_col < M; ctr_col++) {
                System.out.print(numbers[ctr_row][ctr_col] + "\t");
            }
            System.out.println();
        }
    }
}
```

Q13

Test Case

Input

Output

5 20 20 1 20 3 20 20 20 20 20 20 20 20 20 20	Yes
---	-----

Weightage - 10

Input	Output
5 20 20 1 20 3 20 20 20 20 20 20 20 12 20 20	No

Weightage - 10

Input	Output
5 20 20 1 20 5 20 20 20 20 20	Yes

Weightage - 10

Input	Output
9 20 20 1 20 3 20 20 20 1 20 20 20 20 20 20 1 20 20	Yes

Weightage - 15

Input	Output
9 5 3 1 20 3 20 20 20 1 8 20 6 20 7 20 1 8 20 20 20 18 20 20 20 2 1 20	No

Weightage - 15

Input	Output
12 20 20 1 20 3 20 20 20 1 20 1 20 20 20 20 20 20 20 1 20 20 20 20 1 20 20 20 20 20 20 2 1 20 20 20 20	Yes

Weightage - 20

Input	Output
12 20 20 1 20 3 20 20 20 1 20 1 20 20 20 20 20 20 20 1 20 20 20 20 1 20 20 20 20 20 20 2 1 20 20 20 20	No

Weightage - 20

Sample Input	Sample Output
5 20 20 1 20 3 20 20 20 20 20 20 20 20 20 20	Yes

Sample Input

Sample Output

5  
20 20 1 20 3  
20 20 20 20 20  
20 20 12 20 20

No

Solution

```
import java.io.*;
import java.util.*;
class validInitialConfiguration {
    public static void main(String [] args) {
        int i,j,n,s=1;
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        int a[][] = new int[n][n];
        for(i=0;i<n;i++) {
            for(j=0;j<n;j++) {
                a[i][j] = sc.nextInt();
            }
        }
        for(i=0;i<n;i++)
        {
            for(j=0;j<n;j++)
            {
                if((a[i][j]!=20)&&(a[i][j]>10)) {
                    s=0;
                }
            }
        }
        if(s == 0) {
            System.out.println("No");
        }
        else {
            System.out.println("Yes");
        }
    }
}
```

Q14

Test Case

Input

Output

3  
21 3 4  
78 25 19  
50 22 6

2

Weightage - 10

Input

Output

4  
12 25 45 56  
36 58 45 56  
22 10 25 58

3

Weightage - 10

Input

Output

5 10 52 63 54 85 20 12 25 25 36 54 56 52 52 51	2
---	---

Weightage - 10

InputOutput

6 10 52 63 54 85 25 20 12 25 25 36 21 54 56 52 52 51 52	3
--	---

Weightage - 15

InputOutput

7 10 52 63 54 85 25 25 20 12 25 25 36 21 12 54 56 52 52 51 52 12	3
---	---

Weightage - 15

InputOutput

8 10 52 63 54 85 25 25 36 20 12 25 25 36 21 12 45 54 56 52 52 51 52 12 44	4
--	---

Weightage - 20

InputOutput

9 10 52 63 54 85 25 25 36 36 20 12 25 25 36 21 12 45 55 54 56 52 52 51 52 12 44 26	5
---	---

Weightage - 20

Sample InputSample Output

3 21 3 44 78 25 19 50 22 6	2
-------------------------------------	---

Solution

```
import java.io.*;
import java.util.*;
class MidAged {
    public static void main(String [] args) {
        int i,j,n,count=0;
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        int a[][] = new int[n][n];
        for(i=0;i<n;i++) {
            for(j=0;j<n;j++) {
                a[i][j] = sc.nextInt();
            }
        }
    }
}
```

```
        for(i=0;i<n;i++) {
            for(j=0;j<n;j++) {
                if(i==j) {
                    if(a[i][j] >=18 && a[i][j] <=60) {
                        count++;
                    }
                }
            }
        }
        System.out.println(count);
    }
}
```

Q15

Test Case

Input

```
2
2 2
2 2
```

Output

```
Sum
4 4
4 4
Difference
```

Weightage - 20

Input

```
2
0 0
0 0
```

Output

```
Sum
2 2
2 2
Difference
```

Weightage - 20

Input

```
3
100 3 59
32 3 0
20 22 08
```

Output

```
Sum
101 37 112
75 46 22
75 66 110
```

Weightage - 30

Input

```
5
1 1 1 1 1
2 2 2 2 2
2 2 2 2 2
```

Output

```
Sum
11 11 11 11 11
22 22 22 22 22
22 22 22 22 22
```

Weightage - 30

Sample Input

```
3

1 2 3
1 2 2
```

Sample Output

```
Sum
4 4 4
4 4 4
4 4 4
```

Solution

```
import java.util.Scanner;
class Main
{
    public static void main(String args[])
```

```

{
    Scanner in = new Scanner(System.in);
    int m, n, c, d, sum1, first[10][10], second[10][10], sum[10][10], diff[10][10], multiply[10][10];
    m=in.nextInt();
    n=m;
    for (c = 0; c < m; c++)
        for (d = 0; d < n; d++)
            first[c][d] = in.nextInt();
    for (c = 0; c < m; c++)
        for (d = 0 ; d < n; d++)
            ssecond[c][d]=in.nextInt();

    System.out.println("Sum");

    for (c = 0; c < m; c++) {
        for (d = 0 ; d < n; d++) {
            sum[c][d] = first[c][d] + second[c][d];
            System.out.println(sum[c][d]);
        }
        System.out.print("\n");
    }

    System.out.println("Difference");

    for (c = 0; c < m; c++) {
        for (d = 0 ; d < n; d++) {
            diff[c][d] = first[c][d] - second[c][d];
            System.out.println(diff[c][d]);
        }
        System.out.print("\n");
    }

    for (c = 0; c < m; c++) {
        for (d = 0; d < m; d++) {
            for (int k = 0; k < m; k++) {
                sum1 = sum1 + (first[c][k] * second[k][d]);
            }

            multiply[c][d] = sum1;
            sum1 = 0;
        }
    }
    System.out.println("Multiply");
    for (c = 0; c < m; c++) {
        for (d = 0 ; d < n; d++) {
            System.out.println(multiply[c][d]);
        }
        System.out.print("\n");
    }
}
}

```