

Lecture 4 - Two Dimensional Arrays

Number of elements

[Send Feedback](#)

How many elements are present in a two dimensional array arr[2][5]?

Options

This problem has only one correct answer

☐ 4

☐ 7

☒ 10

☐ 3

☒ Hurray! Correct Answer

2-d Array

[Send Feedback](#)

How can we access the last element in last row in a 5*5 2-D array arr?

```
int arr[][]=new int[5][5];
```

Row number and element count starts from 0.

Options

This problem has only one correct answer

☐ arr[5][5]

☒ arr[4][4]

☐ arr[4][5]

☐ arr[5][4]

☒ Hurray! Correct Answer

Predict the output

[Send Feedback](#)

What will be the output of the following code?

```
public static void main (String[] args) {  
    int arr[][]=new int[4][5];  
    for(int i=0;i<5;i++)  
    {  
        arr[i][0]=2;  
    }  
    System.out.print(arr[3][0]);  
}
```

Options

This problem has only one correct answer

☐ 0

☐ 2

☒ Error

☒ Hurray! Correct Answer

Predict the output

[Send Feedback](#)

What will be the output of the following code?

```
public static void main (String[] args) {  
    int arr[][]=new int[4][5];  
    for(int i=0;i<4;i++)  
    {  
        for(int j=0;j<5;j++)  
            arr[i][j]=i*j;  
    }  
    System.out.print(arr[3][4]);  
}
```

Options

This problem has only one correct answer

- ☒ 12
- ☐ 10
- ☐ 14
- ☐ 7
- ☒ Hurray! Correct Answer

What will be the output of the following code?

```
public static void main(String[] args)  
{  
    int[][] arr = new int [2][2];  
    for (int i = 0; i < 2; i++) {  
        for (int j = 0; j < 2; j++) {  
            System.out.print(arr[i][j] + " ");  
        }  
    }  
}
```

0 0 0 0



Correct Answer

What will be the output of the following code?

```
public static void main(String[] args)  
{  
    int[][] arr = { { 1, 2 }, { 3, 4 } };  
    System.out.println(arr[0][0]+arr[1][1]);  
}
```

5



Correct Answer

Length of 2-d array

[Send Feedback](#)

Select the correct option to get number of rows in a 2-d array(arr)?

Options

This problem has only one correct answer

- ☒ arr.length
- ☐ arr[0].length
- ☐ arr[].length
- ☐ arr[].row
- ☒ Hurray! Correct Answer

Predict the output

[Send Feedback](#)

What will be the output of the following code?

```
public static void main(String[] args)
{
    int[][] arr = { {1,2,4,5,7},{3,4,5,6,7},{5,6,7,8,9} };
    System.out.println(arr.length+arr[0].length);
}
```

Options

This problem has only one correct answer

- ☐ 5
- ☐ 6
- ☐ 7
- ☒ 8
- ☒ Hurray! Correct Answer

Row Wise Sum

[Send Feedback](#)

For a given two-dimensional integer array/list of size (N x M), find and print the sum of each of the row elements in a single line, separated by a single space.

Input Format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains two integer values, 'N' and 'M', separated by a single space. They represent the 'rows' and 'columns' respectively, for the two-dimensional array/list.

Second line onwards, the next 'N' lines or rows represent the ith row values.

Each of the ith row constitutes 'M' column values separated by a single space.

Output Format :

For each test case, print the sum of every ith row elements in a single line separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^3$

$0 \leq M \leq 10^3$

Time Limit: 1sec

Sample Input 1:

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

Sample Output 1:

```
3 7 11 15
```

Sample Input 2:

```
2
2 5
4 5 3 2 6
7 5 3 8 9
4 4
1 2 3 4
9 8 7 6
3 4 5 6
-1 1 -10 5
```

Sample Output 2:

```
20 32
10 30 18 -5
```

```
public class Solution {

    public static void rowWiseSum(int[][] mat) {
        //Your code goes here
        int rows = mat.length;
        int cols = rows>0?mat[0].length:0;

        for(int i=0; i<rows; i++){
            int sum = 0;
            for(int j=0; j<cols; j++){
                sum += mat[i][j];
            }
            System.out.print(sum+" ");
        }
    }
}
```

Return type

[Send Feedback](#)

What should be the return type of a function that returns a 2-d array?

Options

This problem has only one correct answer

☐ int

☐ int []

☒ int [][]

☒ Hurray! Correct Answer

Predict the output

[Send Feedback](#)

What will be the output of the following code?

```
public static int fun(int[][] arr2d)
{
    int sum=0;
    for(int i=0;i<arr2d.length;i++)
    {
        for(int j=0;j<arr2d[i].length;j++)
        {
            sum+=arr2d[i][j];
        }
    }
    return sum;
}
public static void main(String[] args)
{
    int[][] arr = { {0,1,2,4,5}, {3,4,5,6,7},{5,6,7,8,9} };
    System.out.println(fun(arr));
}
```



Options

This problem has only one correct answer

☐ 48

☐ 56

☐ 63

☒ 72


☒ Hurray! Correct Answer

Predict the output

[Send Feedback](#)

What will be the output of the following code?

```
public static void fun(int[][] arr2d)
{
    for(int i=0;i<arr2d.length;i++)
    {
        for(int j=0;j<arr2d[i].length;j++)
        {
            arr2d[i][j]=2*i+j;
        }
    }
}
public static void main(String[] args)
{
    int[][] arr = new int[2][2];
    fun(arr);
    for(int i=0;i<2;i++)
    {
        for(int j=0;j<2;j++)
        {
            System.out.print(arr[i][j]+" ");
        }
    }
}
```

 You have max 2 attempts to score in this question.

Options

This problem has only one correct answer

☐ 0 0 0 0

☒ 0 1 2 3

☐ 1 2 3 4

☐ 0 1 1 2

☒ Hurray! Correct Answer

Largest Row or Column

[Send Feedback](#)

For a given two-dimensional integer array/list of size ($N \times M$), you need to find out which row or column has the largest sum (sum of all the elements in a row/column) amongst all the rows and columns.

Note :

If there are more than one rows/columns with maximum sum, consider the row/column that comes first. And if i th row and j th column has the same largest sum, consider the i th row as answer.

Input Format :

The first line contains an Integer ' t ' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains two integer values, ' N ' and ' M ', separated by a single space. They represent the 'rows' and 'columns' respectively, for the two-dimensional array/list.

Second line onwards, the next ' N ' lines or rows represent the i th row values.

Each of the i th row constitutes ' M ' column values separated by a single space.

Output Format :

For each test case, If row sum is maximum, then print: "row" <row_index> <row_sum>
OR
If column sum is maximum, then print: "column" <col_index> <col_sum>
It will be printed in a single line separated by a single space between each piece of information.

Output for every test case will be printed in a separate line.

Consider :

If there doesn't exist a sum at all then print "row 0 -2147483648", where -2147483648 or -2^{31} is the smallest value for the range of Integer.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^3$

$0 \leq M \leq 10^3$

Time Limit: 1sec

Sample Input 1 :

```
1
2 2
1 1
1 1
```

Sample Output 1 :

row 0 2

Sample Input 2 :

```
2
3 3
3 6 9
1 4 7
2 8 9
4 2
1 2
90 100
3 40
-10 200
```

Sample Output 2 :

```
column 2 25
column 1 342
```

```
public class Solution {
```

```
    public static void findLargest(int mat[][]){
        //Your code goes here
```

```
        int rows = mat.length;
        int cols = rows>0?mat[0].length:0;
        int max = Integer.MIN_VALUE;
        int max1 = Integer.MIN_VALUE;
        int rowno = 0;
        int colno = 0;
```

```
        for(int i=0; i<rows; i++){
            int sum = 0;
            for(int j=0; j<cols; j++){
```



```
        sum += mat[i][j];
    }
    if(sum>max){
        max = sum;
        rowno = i;
    }
}
for(int j=0; j<cols; j++){
    int sum = 0;
    for(int i=0; i<rows; i++){
        sum += mat[i][j];
    }
    if(sum>max1){
        max1 = sum;
        colno = j;
    }
}
if(max>=max1)
    System.out.println("row "+rowno+" "+max);
else
    System.out.println("column "+colno+" "+max1);
}
}
```

Predict the output

[Send Feedback](#)

What will be the output of the following code?

```
public static void main (String[] args) {  
    int arr[][]={ {1,2,3,4} , {2,4} , {3,5,7,8,9,11} };  
    System.out.print(arr.length+arr[0].length+  
        arr[1].length+arr[2].length);  
}
```

Options

This problem has only one correct answer

☐ 12

☐ 13

☒ 15

☐ 17

☒ Hurray! Correct Answer

Predict the output

[Send Feedback](#)

What will be the output of the following code?

```
public static void main (String[] args) {  
    int arr[][]=new int [4][];  
    System.out.print(arr[2].length);  
}
```

Options

This problem has only one correct answer

☐ 0

☒ Null Pointer Exception

☐ Garbage Value

☐ An Integer address

☒ Hurray! Correct Answer

Declare a 2-d array

[Send Feedback](#)

How can we declare a 2d array with 4 rows but the number of columns in each row are not set?

Options

This problem has only one correct answer

☒ `int a[][]=new int [4][];`

☐ `int a[]=new int[4][];`

☐ `int a[][]=new int[4];`

☐ `int a[][]=new int [][4];`

☒ Hurray! Correct Answer

Total Sum on the Boundaries and Diagonals

[Send Feedback](#)

For a given two-dimensional square matrix of **size** ($N \times N$). Find the total sum of elements on both the diagonals and at all the four boundaries.

Input format:

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains a single integer value, 'N' representing the 'rows' and 'columns' for the two-dimensional square matrix.

Second line onwards, the next 'N' lines or rows represent the ith row values.

Each of the ith row constitutes 'N' column values separated by a single space.

Output format:

For each test case, print the single integer denoting the sum.

Output for every test case will be printed in a separate line.

Constraints:

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^3$

Time Limit: 1sec

Sample input 1:

1

3

1 2 3

4 5 6

7 8 9

Sample Output 1:

45

Explanation for Sample Output 1:

The boundary elements are 1, 2, 3, 6, 9, 8, 7 and 4.

The first-diagonal elements are 1, 5 and 9.

The second-diagonal elements are 3, 5 and 7.

We just need to add all these numbers making sure that no number is added twice. For example, '1' is both a boundary element and a first-diagonal element similarly, '5' contributes to both the diagonals but they won't be added twice.

Hence, we add up, $[1 + 2 + 3 + 6 + 9 + 8 + 7 + 4 + 5]$ to give 45 as the output.

Sample input 2:

2

5

1 2 3 4 5

6 7 8 9 10

11 12 13 14 15

16 17 18 19 20

21 22 23 24 25

4

1 2 3 10

4 5 6 11

7 8 9 12

13 14 15 16

Sample Output 2:

273

136

```
public class Solution {

    public static void totalSum(int[][] mat) {
        //Your code goes here(
        int rows = mat.length;
        int cols = rows>0?mat[0].length:0;
        int sum = 0;

        for(int i=0; i<rows; i++){
            for(int j=0; j<cols; j++){
                if(i==0 || j==0 || i==rows-1 || j==cols-1 || i==j ||
i+j==rows-1){
                    sum += mat[i][j];
                }
            }
        }
        System.out.println(sum);
    }

}
```

Print Like a Wave

Send Feedback

For a given two-dimensional integer array/list of size ($N \times M$), print the array/list in a sine wave order, i.e, print the first column top to bottom, next column bottom to top and so on.

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains two integer values, 'N' and 'M', separated by a single space. They represent the 'rows' and 'columns' respectively, for the two-dimensional array/list.

Second line onwards, the next 'N' lines or rows represent the ith row values.

Each of the ith row constitutes 'M' column values separated by a single space.

Output format :

For each test case, print the elements of the two-dimensional array/list in the sine wave order in a single line, separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^3$

$0 \leq M \leq 10^3$

Time Limit: 1sec

```

Sample Input 1:
1
3 4
1 2 3 4
5 6 7 8
9 10 11 12
Sample Output 1:
1 5 9 10 6 2 3 7 11 12 8 4
Sample Input 2:
2
5 3
1 2 3
4 5 6
7 8 9
10 11 12
13 14 15
3 3
10 20 30
40 50 60
70 80 90
Sample Output 2:
1 4 7 10 13 14 11 8 5 2 3 6 9 12 15
10 40 70 80 50 20 30 60 90

public class Solution {

    public static void wavePrint(int mat[][]){
        //Your code goes here
        int rows = mat.length;
        int cols = rows>0?mat[0].length:0;

        for(int j=0; j<cols; j++){
            if(j%2==0)
                for(int i=0; i<rows; i++)
                    System.out.print(mat[i][j]+" ");
            else
                for(int i=rows-1; i>=0; i--)
                    System.out.print(mat[i][j]+" ");
        }
    }
}

```

Print Spiral

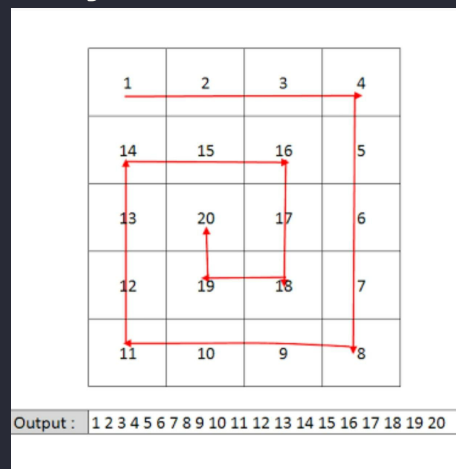
Send Feedback

For a given two-dimensional integer array/list of size ($N \times M$), print it in a spiral form. That is, you need to print in the order followed for every iteration:

- First row(left to right)
- Last column(top to bottom)
- Last row(right to left)
- First column(bottom to top)

Mind that every element will be printed only once.

Refer to the Image:



Spiral path of a matrix

Input format :

The first line contains an Integer 't' which denotes the number of test cases or queries to be run. Then the test cases follow.

First line of each test case or query contains two integer values, 'N' and 'M', separated by a single space. They represent the 'rows' and 'columns' respectively, for the two-dimensional array/list.

Second line onwards, the next 'N' lines or rows represent the ith row values.

Each of the ith row constitutes 'M' column values separated by a single space.

Output format :

For each test case, print the elements of the two-dimensional array/list in the spiral form in a single line, separated by a single space.

Output for every test case will be printed in a separate line.

Constraints :

$1 \leq t \leq 10^2$

$0 \leq N \leq 10^3$

$0 \leq M \leq 10^3$

Time Limit: 1sec

Sample Input 1:

```
1
4 4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

Sample Output 1:

```
1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10
```

Sample Input 2:

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

```
4 5 6
7 8 9
3 1
10
20
30
```

Sample Output 2:

```
1 2 3 6 9 8 7 4 5
10 20 30
```

```
public class Solution {

    public static void spiralPrint(int matrix[][]){
        //Your code goes here

        int rowstart = 0;
        int rowend = matrix.length-1;
        int colstart = 0;
        int colend=0;
        if(rowend>0)
            colend = matrix[0].length-1;
        else
            System.exit(0);

        int i;
        int j;
        int count=0;
        while(count<matrix.length*matrix[0].length){

            for(i=rowstart, j=colstart; j<=colend; j++){
                System.out.print(matrix[i][j]+" ");
                count++;
            }

            for(j=colend, i=++rowstart; i<=rowend; i++){
                System.out.print(matrix[i][j]+" ");
                count++;
            }

            for(i=rowend, j=--colend; j>=colstart; j--){
                System.out.print(matrix[i][j]+" ");
                count++;
            }

            for(j=colstart, i=--rowend; i>=rowstart; i--){
                System.out.print(matrix[i][j]+" ");
                count++;
            }
            ++colstart;
        }
    }
}
```

