|  |  |
| --- | --- |
| Sample Output |  |
| 23 98 75  10 90 33  88 88 17  34 60 45  Row#1 Sum 133  Col#2 Max: 75 |  |

1. We will create a 2D array and work with it.

1. Complete the method main to fill matrix with random integer values on the interval [10, 100).

public static void main(String[] args)

{

int numRows = (int)(Math.random() \* 5 + 3);

int numCols = (int)(Math.random() \* 6 + 2);  
 int[][] matrix = new int[numRows][numCols];

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

display(matrix);

int sum = findRowSum(matrix, 1);

System.out.println(“Sum of Row#1 = “ + sum);

int max = findMaxNum(matrix, 2);

System.out.println(“Max of Col#2 = “ + max);

}

1. Complete the method findRowSum to return the sum of all the values in row arg of matrix.

public static int findRowSum(int[][] matrix, int arg)

{

}

1. Complete the method findMaxNum to return the maximum value out of all the values in column arg of matrix.

public static int findMaxNum(int[][] matrix, int arg)

{

}

1. Complete the method contains to return true if the value arg is in the matrix and return false if the matrix does not contain the value arg.

public static boolean contains(int[][] matrix, int arg)

{

}

**True / False:** The array matrix is declared below. Answer the questions #2 through #5.

int[][] matrix = new int[3][4];

T F 2. The **matrix** must have 3 columns and 4 rows.

T F 3. The statement **matrix[1][1]=10**; would assign the value 10 in one of the cells of the array.

T F 4. In the statement **int x= matrix[0].length;** x would equal 4.

T F 5. In the statement **int y = matrix.length;** y would equal 4.

6. Once filled, the array **table** should appear as follows:

3.7 9.8 5.2

-4.2 6.2 3.1

1.8 -9.2 7.2

-4.3 -0.3 3.5

2.3 1.5 -1.1

a) Write the code that instantiates (not fill) **table** correctly.

b) The value of table[3][2] is \_\_\_\_\_\_\_\_\_\_\_

7. There is a pattern in this square matrix. What is the pattern? Write the code to   
 create any square matrix with the same pattern.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 1 | 1 | 1 | 1 |
| 3 | 2 | 1 | 1 | 1 |
| 3 | 3 | 2 | 1 | 1 |
| 3 | 3 | 3 | 2 | 1 |
| 3 | 3 | 3 | 3 | 2 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

8. Here is the left half of a black-and-white 2-D array of pixels. Write the code so that the right side is a mirror image of the left side. Algorithm: for each pixel on the left side, put that pixel in the corresponding position on the right side.

for(int r=0; r<pixels.length; r++)