Q1: Java Program to Analyze Matrix Elements

```
import java.util.Scanner;

public class MatrixAnalysis {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Take input for matrix dimensions
        System.out.print("Enter the number of rows (m): ");
        int m = scanner.nextInt();
        System.out.print("Enter the number of columns (n): ");
        int n = scanner.nextInt();

        // Initialize the matrix
        int[][] matrix = new int[m][n];

        // Counters for different types of numbers
        int positiveCount = 0;
        int negativeCount = 0;
        int oddCount = 0;
        int evenCount = 0;
        int zeroCount = 0;

        // Take input for matrix elements and analyze them
        System.out.println("Enter " + (m * n) + " integers for the matrix:");
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                matrix[i][j] = scanner.nextInt();

                // Check if the number is positive, negative, or zero
                if (matrix[i][j] > 0) {
                    positiveCount++;
                } else if (matrix[i][j] < 0) {
                    negativeCount++;
                } else {
                    zeroCount++;
                }

                // Check if the number is odd or even
                if (matrix[i][j] != 0 && matrix[i][j] % 2 == 0) {
                    evenCount++;
                } else if (matrix[i][j] % 2 != 0) {
                    oddCount++;
```

```
            }
        }
    }

    // Print the results
    System.out.println("Number of positive numbers: " + positiveCount);
    System.out.println("Number of negative numbers: " + negativeCount);
    System.out.println("Number of odd numbers: " + oddCount);
    System.out.println("Number of even numbers: " + evenCount);
    System.out.println("Number of zeros: " + zeroCount);

    scanner.close();
  }
}
```

Q2: Java Program to Print Elements Above the Secondary Diagonal

```
import java.util.Scanner;

public class AboveSecondaryDiagonal {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    // Input the size of the square matrix
    System.out.print("Enter the size of the square matrix (n): ");
    int n = scanner.nextInt();

    // Initialize the matrix
    int[][] matrix = new int[n][n];

    // Input matrix elements
    System.out.println("Enter the elements of the matrix:");
    for (int i = 0; i < n; i++) {
      for (int j = 0; j < n; j++) {
        matrix[i][j] = scanner.nextInt();
      }
    }

    // Print elements above the secondary diagonal
    System.out.println("Elements above the secondary diagonal:");
    for (int i = 0; i < n; i++) {
      for (int j = 0; j < n - i - 1; j++) {  // Elements above secondary diagonal
        System.out.print(matrix[i][j] + " ");
      }
```

```
        }

        scanner.close();
    }
}
```

Q3: Java Program to Print the Elements of Both Diagonals in a User-Inputted Square Matrix

```
import java.util.Scanner;
import java.util.HashSet;

public class PrintDiagonals {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Input the size of the square matrix
        System.out.print("Enter the size of the square matrix (n): ");
        int n = scanner.nextInt();

        // Initialize the matrix
        int[][] matrix = new int[n][n];

        // Input matrix elements
        System.out.println("Enter the elements of the matrix:");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                matrix[i][j] = scanner.nextInt();
            }
        }

        // Use a HashSet to avoid duplicates when n is odd and the center element is shared
        HashSet<Integer> diagonalElements = new HashSet<>();

        // Collect elements from the primary diagonal
        for (int i = 0; i < n; i++) {
            diagonalElements.add(matrix[i][i]);
        }

        // Collect elements from the secondary diagonal
        for (int i = 0; i < n; i++) {
            diagonalElements.add(matrix[i][n - i - 1]);
        }

        // Print elements of both diagonals
```

```
        System.out.println("Elements of both diagonals:");
        for (int element : diagonalElements) {
            System.out.print(element + " ");
        }

        scanner.close();
    }
}
```

Q4: Java Program to Find the Largest Element in a 2D Array

```
import java.util.Scanner;

public class LargestElementIn2DArray {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Input the dimensions of the 2D array
        System.out.print("Enter the number of rows: ");
        int rows = scanner.nextInt();
        System.out.print("Enter the number of columns: ");
        int columns = scanner.nextInt();

        // Initialize the 2D array
        int[][] array = new int[rows][columns];

        // Input array elements
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                array[i][j] = scanner.nextInt();
            }
        }

        // Initialize maxElement with the smallest possible integer value
        int maxElement = Integer.MIN_VALUE;

        // Traverse the 2D array to find the largest element
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                if (array[i][j] > maxElement) {
                    maxElement = array[i][j];
                }
            }
```

```
        }

        // Print the largest element
        System.out.println("The largest element in the array is: " + maxElement);

        scanner.close();
    }
}
```

Q5: Write a function which accepts a 2D array of integers and its size as arguments and displays the elements of middle row and the elements of middle column. Printing can be done in any order.
[Assuming the 2D Array to be a square matrix with odd dimensions i.e. 3x3, 5x5, 7x7 etc...]

```
import java.util.ArrayList;
import java.util.HashSet;
import java.util.Scanner;

public class MiddleRowAndColumn {

    // Function to display the middle row and middle column elements
    public static void printMiddleRowAndColumn(int[][] matrix, int size) {
        int middle = size / 2;  // Calculate the index of the middle row/column

        // Use a HashSet to avoid duplicate elements
        HashSet<Integer> elements = new HashSet<>();

        // Add elements of the middle row
        for (int j = 0; j < size; j++) {
            elements.add(matrix[middle][j]);
        }

        // Add elements of the middle column
        for (int i = 0; i < size; i++) {
            elements.add(matrix[i][middle]);
        }

        // Print all elements in one line
        System.out.println("Elements of the middle row and column:");
        for (int element : elements) {
            System.out.print(element + " ");
        }
    }
```

```
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Input the size of the square matrix
        System.out.print("Enter the size of the square matrix (odd number): ");
        int n = scanner.nextInt();

        // Ensure the size is odd
        if (n % 2 == 0) {
            System.out.println("Please enter an odd number for the size of the matrix.");
            return;
        }

        // Initialize the matrix
        int[][] matrix = new int[n][n];

        // Input matrix elements
        System.out.println("Enter the elements of the matrix:");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                matrix[i][j] = scanner.nextInt();
            }
        }

        // Call the function to print the middle row and column
        printMiddleRowAndColumn(matrix, n);

        scanner.close();
    }
}
```