**Вариант 1**

1. Реализуйте метод **int countPositive(double[][] matrix)**, возвращающий количество положительных элементов в матрице.
2. Реализуйте метод **double[] sumOfSquaresRows(double[][] matrix)**, возвращающий массив, где каждый элемент — это сумма квадратов элементов соответствующей строки.
3. Реализуйте метод **int rowWithMaxValue(int[][] matrix)**, возвращающий номер строки, содержащей максимальный элемент матрицы (нумерация с 0).

**Вариант 2**

1. Реализуйте метод **double averageMatrix(double[][] matrix)**, возвращающий среднее арифметическое всех элементов матрицы.
2. Реализуйте метод **double[] minInColumns(double[][] matrix)**, возвращающий массив минимальных значений каждого столбца.
3. Реализуйте метод **int columnWithMinSum(double[][] matrix)**, возвращающий номер столбца с наименьшей суммой элементов.

**Вариант 3**

1. Реализуйте метод **int countZeros(int[][] matrix)**, возвращающий количество нулевых элементов в матрице.
2. Реализуйте метод **boolean[] isRowSorted(double[][] matrix)**, возвращающий массив булевых значений, где **true** означает, что строка упорядочена по возрастанию.
3. Реализуйте метод **int rowWithMostNegatives(int[][] matrix)**, возвращающий номер строки с наибольшим количеством отрицательных элементов.

**Вариант 4**

1. Реализуйте метод **double differenceMaxMin(double[][] matrix)**, возвращающий разность между максимальным и минимальным элементами матрицы.
2. Реализуйте метод **double[] productNonZeroRowElements(double[][] matrix)**, возвращающий массив произведений ненулевых элементов каждой строки (если все нули — вернуть 0 для этой строки).
3. Реализуйте метод **int columnWithFewestPositives(double[][] matrix)**, возвращающий номер столбца с наименьшим количеством положительных элементов.

**Вариант 5**

1. Реализуйте метод **int countFullyPositiveRows(double[][] matrix)**, возвращающий количество строк, в которых все элементы положительны.
2. Реализуйте метод **double[] geometricMeanColumns(double[][] matrix)**, возвращающий массив, где каждый элемент — это среднее геометрическое элементов соответствующего столбца.
3. Реализуйте метод **int rowWithMostDuplicates(double[][] matrix)**, возвращающий номер строки, содержащей наибольшее количество одинаковых элементов.

**Вариант 6**

1. Реализуйте метод **int countZeroRows(double[][] matrix)**, возвращающий количество строк, состоящих только из нулей.
2. Реализуйте метод **double[] sumAbsoluteValuesRows(double[][] matrix)**, возвращающий массив, где каждый элемент — это сумма абсолютных значений соответствующей строки.
3. Реализуйте метод **int columnWithMaxSpread(double[][] matrix)**, возвращающий номер столбца с наибольшей разницей между максимальным и минимальным элементами.

**Вариант 7**

1. Реализуйте метод **int countPositiveOnlyColumns(double[][] matrix)**, возвращающий количество столбцов, в которых все элементы положительны.
2. Реализуйте метод **int[] uniqueElementCountPerRow(double[][] matrix)**, возвращающий массив, где каждый элемент — это количество уникальных элементов в соответствующей строке.
3. Реализуйте метод **int rowWithSmallestAverage(double[][] matrix)**, возвращающий номер строки с наименьшим средним арифметическим.

**Вариант 8**

1. Реализуйте метод **int countRowsWithZero(double[][] matrix)**, возвращающий количество строк, в которых есть хотя бы один ноль.
2. Реализуйте метод **double[] mediansPerRow(double[][] matrix)**, возвращающий массив медиан для каждой строки матрицы.
3. Реализуйте метод **int columnWithMaxProduct(double[][] matrix)**, возвращающий номер столбца с наибольшим произведением элементов.

**Вариант 9**

1. Реализуйте метод **int countRowsWithKZeros(int[][] matrix, int k)**, возвращающий количество строк, в которых ровно **k** нулей.
2. Реализуйте метод **double[] firstMinusLastInRow(double[][] matrix)**, возвращающий массив разностей между первым и последним элементом каждой строки.
3. Реализуйте метод **int rowWithMaxSumOfNegativeAbs(int[][] matrix)**, возвращающий номер строки, в которой сумма модулей отрицательных чисел максимальна.

**Вариант 10**

1. Реализуйте метод **int countPositiveOnlyRows(double[][] matrix)**, возвращающий количество строк, в которых нет отрицательных элементов.
2. Реализуйте метод **int[] evenCountPerRow(int[][] matrix)**, возвращающий массив, где каждый элемент — это количество чётных чисел в соответствующей строке.
3. Реализуйте метод **int columnWithMaxSum(double[][] matrix)**, возвращающий номер столбца с наибольшей суммой элементов.

## 🔹 **Вариант 11**

1. Реализуйте метод **int countNegativeRows(double[][] matrix)**, возвращающий количество строк, в которых все элементы отрицательные.
2. Реализуйте метод **double[] sumOfEvenElementsPerRow(int[][] matrix)**, возвращающий массив сумм чётных чисел по строкам.
3. Реализуйте метод **int rowWithMaxMinDifference(double[][] matrix)**, возвращающий номер строки с наибольшей разницей между максимальным и минимальным элементом.

## 🔹 **Вариант 12**

1. Реализуйте метод **int countZeroColumns(double[][] matrix)**, возвращающий количество столбцов, состоящих только из нулей.
2. Реализуйте метод **double averageInOddRows(double[][] matrix)**, возвращающий среднее арифметическое всех элементов строк с нечётными номерами.
3. Реализуйте метод **int columnWithMostPositiveValues(double[][] matrix)**, возвращающий номер столбца с наибольшим количеством положительных элементов.

## 🔹 **Вариант 13**

1. Реализуйте метод **int countRowsIncreasing(double[][] matrix)**, возвращающий количество строк, упорядоченных по возрастанию.
2. Реализуйте метод **double[] averageOfMinMaxColumns(double[][] matrix)**, возвращающий массив из двух значений: среднего арифметического минимальных и максимальных элементов столбцов.
3. Реализуйте метод **int rowWithMaxSumAfterK(int[][] matrix, int k)**, возвращающий номер строки, начиная с K-го элемента которой сумма элементов максимальна.

## 🔹 **Вариант 14**

1. Реализуйте метод **int countUniqueElements(double[][] matrix)**, возвращающий общее количество уникальных элементов в матрице.
2. Реализуйте метод **boolean[] areRowsEqual(double[][] matrix)**, возвращающий массив булевых значений, где **true** означает, что соответствующая строка совпадает со следующей.
3. Реализуйте метод **int columnWithLargestAverage(double[][] matrix)**, возвращающий номер столбца с наибольшим средним арифметическим.

## 🔹 **Вариант 15**

1. Реализуйте метод **int countRowsWithSameFirstAndLast(double[][] matrix)**, возвращающий количество строк, в которых первый и последний элементы совпадают.
2. Реализуйте метод **double[] productOfNonZeroColumnElements(double[][] matrix)**, возвращающий массив произведений ненулевых элементов каждого столбца (если все нули — вернуть 0).
3. Реализуйте метод **int rowWithMinProduct(double[][] matrix)**, возвращающий номер строки с наименьшим произведением элементов.

## 🔹 **Вариант 16**

1. Реализуйте метод **int countRowsSortedDescending(double[][] matrix)**, возвращающий количество строк, упорядоченных по убыванию.
2. Реализуйте метод **double[] sumOfPositiveElementsByRow(double[][] matrix)**, возвращающий массив сумм положительных элементов в каждой строке.
3. Реализуйте метод **int columnWithMostNegatives(double[][] matrix)**, возвращающий номер столбца с наибольшим количеством отрицательных элементов.

## 🔹 **Вариант 17**

1. Реализуйте метод **int countSymmetricRows(double[][] matrix)**, возвращающий количество строк, которые являются палиндромами.
2. Реализуйте метод **double[] averageOfAbsElementsByRow(double[][] matrix)**, возвращающий массив средних арифметических абсолютных значений по строкам.
3. Реализуйте метод **int rowWithMaxMedian(double[][] matrix)**, возвращающий номер строки с наибольшей медианой.

## 🔹 **Вариант 18**

1. Реализуйте метод **int countIdenticalRows(double[][] matrix)**, возвращающий количество пар одинаковых строк.
2. Реализуйте метод **double[] averageOfPositiveElementsByColumn(double[][] matrix)**, возвращающий массив средних арифметических положительных элементов по столбцам.
3. Реализуйте метод **int columnWithMinMedian(double[][] matrix)**, возвращающий номер столбца с наименьшей медианой.

## 🔹 **Вариант 19**

1. Реализуйте метод **int countRowsWithoutZero(double[][] matrix)**, возвращающий количество строк, в которых нет ни одного нуля.
2. Реализуйте метод **double[] sumOfMinMaxInEachRow(double[][] matrix)**, возвращающий массив сумм минимального и максимального элемента в каждой строке.
3. Реализуйте метод **int rowWithMostEvenNumbers(int[][] matrix)**, возвращающий номер строки с наибольшим количеством чётных чисел.

## 🔹 **Вариант 20**

1. Реализуйте метод **int countRowsStartsWithPositive(double[][] matrix)**, возвращающий количество строк, начинающихся с положительного числа.
2. Реализуйте метод **double[] averageOfNegativeElementsByRow(double[][] matrix)**, возвращающий массив средних арифметических отрицательных элементов по строкам.
3. Реализуйте метод **int columnWithMostZeros(double[][] matrix)**, возвращающий номер столбца с наибольшим количеством нулевых элементов.

## 🔹 **Вариант 21**

1. Реализуйте метод **int countRowsWithAllEven(int[][] matrix)**, возвращающий количество строк, в которых все элементы чётные.
2. Реализуйте метод **double[] sumOfNegativeElementsByRow(double[][] matrix)**, возвращающий массив сумм отрицательных элементов по строкам.
3. Реализуйте метод **int columnWithMinSumAbs(double[][] matrix)**, возвращающий номер столбца с наименьшей суммой абсолютных значений элементов.

## 🔹 **Вариант 22**

1. Реализуйте метод **int countRowsWithAlternatingSigns(double[][] matrix)**, возвращающий количество строк, в которых знаки элементов чередуются (+ - + и т.д.).
2. Реализуйте метод **double averageOfMaxInEachColumn(double[][] matrix)**, возвращающий среднее арифметическое максимальных элементов каждого столбца.
3. Реализуйте метод **int rowWithMostZeros(double[][] matrix)**, возвращающий номер строки с наибольшим количеством нулевых элементов.

## 🔹 **Вариант 23**

1. Реализуйте метод **int countRowsDecreasing(double[][] matrix)**, возвращающий количество строк, упорядоченных по убыванию.
2. Реализуйте метод **double[] averageOfOddElementsPerRow(int[][] matrix)**, возвращающий массив средних арифметических нечётных чисел по строкам.
3. Реализуйте метод **int columnWithLargestSpread(double[][] matrix)**, возвращающий номер столбца с наибольшей разницей между максимальным и минимальным элементами.

## 🔹 **Вариант 24**

1. Реализуйте метод **int countColumnsWithSameFirstAndLast(double[][] matrix)**, возвращающий количество столбцов, где первый и последний элемент совпадают.
2. Реализуйте метод **double[] productOfPositiveElementsByColumn(double[][] matrix)**, возвращающий массив произведений положительных элементов по столбцам (если их нет — вернуть 0).
3. Реализуйте метод **int rowWithMinAverage(double[][] matrix)**, возвращающий номер строки с наименьшим средним арифметическим.

## 🔹 **Вариант 25**

1. Реализуйте метод **int countSymmetricColumns(double[][] matrix)**, возвращающий количество столбцов, которые являются палиндромами.
2. Реализуйте метод **double[] sumOfMinMaxInEachColumn(double[][] matrix)**, возвращающий массив сумм минимального и максимального элемента в каждом столбце.
3. Реализуйте метод **int columnWithMostEvenNumbers(int[][] matrix)**, возвращающий номер столбца с наибольшим количеством чётных чисел.

## 🔹 **Вариант 26**

1. Реализуйте метод **int countRowsStartsWithZero(double[][] matrix)**, возвращающий количество строк, начинающихся с нуля.
2. Реализуйте метод **double[] averageOfPositiveElementsByRow(double[][] matrix)**, возвращающий массив средних арифметических положительных элементов по строкам.
3. Реализуйте метод **int columnWithMostUniqueValues(double[][] matrix)**, возвращающий номер столбца с наибольшим количеством уникальных элементов.

## 🔹 **Вариант 27**

1. Реализуйте метод **int countRowsWithEqualSumAndProduct(double[][] matrix)**, возвращающий количество строк, где сумма элементов равна произведению.
2. Реализуйте метод **double averageOfNegativeColumns(double[][] matrix)**, возвращающий среднее арифметическое всех элементов столбцов, содержащих только отрицательные числа.
3. Реализуйте метод **int rowWithSmallestMedian(double[][] matrix)**, возвращающий номер строки с наименьшей медианой.

## 🔹 **Вариант 28**

1. Реализуйте метод **int countRowsWithoutDuplicates(double[][] matrix)**, возвращающий количество строк, в которых нет повторяющихся элементов.
2. Реализуйте метод **double[] geometricMeanPerRow(double[][] matrix)**, возвращающий массив средних геометрических для каждой строки.
3. Реализуйте метод **int columnWithMaxNegativeSum(double[][] matrix)**, возвращающий номер столбца с наибольшей суммой отрицательных элементов.

## 🔹 **Вариант 29**

1. Реализуйте метод **int countRowsWithOnlyPrimeNumbers(int[][] matrix)**, возвращающий количество строк, состоящих только из простых чисел.
2. Реализуйте метод **double[] averageOfEvenElementsByColumn(int[][] matrix)**, возвращающий массив средних арифметических чётных чисел по столбцам.
3. Реализуйте метод **int rowWithMostDistinctElements(double[][] matrix)**, возвращающий номер строки с наибольшим количеством различных элементов.

## 🔹 **Вариант 30**

1. Реализуйте метод **int countRowsWithAllSameElements(double[][] matrix)**, возвращающий количество строк, в которых все элементы одинаковые.
2. Реализуйте метод **double[] sumOfAbsoluteValuesColumns(double[][] matrix)**, возвращающий массив сумм абсолютных значений элементов по столбцам.
3. Реализуйте метод **int columnWithMaxMedian(double[][] matrix)**, возвращающий номер столбца с наибольшей медианой.