```
import kotlin.random.Random

fun main() {
    // 1. Создание и вывод элементов
    val array1 = intArrayOf(1, 2, 3, 4, 5)
    println("Элементы массива 1: ${array1.joinToString(", ")}")

    // 2. Сумма элементов массива
    val sum = array1.sum()
    println("Сумма элементов массива: $sum")

    // 3. Максимальное и минимальное значение
    val array2 = intArrayOf(5, 3, 8, 1, 4, 9, 2, 10, 6, 7)
    val max = array2.maxOrNull()
    val min = array2.minOrNull()
    println("Максимальное значение: $max, Минимальное значение: $min")

    // 4. Сортировка массива
    val sortedArray = array2.sortedArray()
    println("Отсортированный массив: ${sortedArray.joinToString(", ")}")

    // 5. Уникальные элементы
    val array3 = intArrayOf(1, 2, 2, 3, 4, 4, 5)
    val uniqueElements = array3.toSet()
    println("Уникальные элементы: ${uniqueElements.joinToString(", ")}")

    // 6. Четные и нечетные числа
    val evenNumbers = array3.filter { it % 2 == 0 }
    val oddNumbers = array3.filter { it % 2 != 0 }
    println("Четные числа: ${evenNumbers.joinToString(", ")}")
    println("Нечетные числа: ${oddNumbers.joinToString(", ")}")
```

```
// 7. Реверс массива
val reversedArray = array3.reversedArray()
println("Реверсированный массив: ${reversedArray.joinToString(", ")}")

// 8. Поиск элемента
val searchElement = 3
val index = array3.indexOf(searchElement)
println("Индекс элемента $searchElement: $index")

// 9. Копирование массива
val copiedArray = array3.copyOf()
println("Скопированный массив: ${copiedArray.joinToString(", ")}")

// 10. Сумма четных чисел
val sumEven = array3.filter { it % 2 == 0 }.sum()
println("Сумма четных чисел: $sumEven")

// 11. Пересечение массивов
val array4 = intArrayOf(2, 3, 5, 7)
val intersection = array3.toSet().intersect(array4.toSet())
println("Пересечение массивов: ${intersection.joinToString(", ")}")

// 12. Перестановка элементов
fun swap(array: IntArray, index1: Int, index2: Int) {
    val temp = array[index1]
    array[index1] = array[index2]
    array[index2] = temp
}
swap(array3, 0, 1)
println("Массив после перестановки: ${array3.joinToString(", ")}")

// 13. Заполнение случайными числами
```

```
val randomArray = IntArray(20) { Random.nextInt(1, 101) }
println("Случайные числа: ${randomArray.joinToString(", ")}")

// 14. Числа Прокопенко
val numbersDivisibleBy3 = randomArray.filter { it % 3 == 0 }
println("Числа, делящиеся на 3: ${numbersDivisibleBy3.joinToString(", ")}")

// 15. Проверка на палиндром
val isPalindrome = array3.contentEquals(array3.reversedArray())
println("Массив является палиндромом: $isPalindrome")

// 16. Конкатенация двух массивов
val concatenatedArray = array3 + array4
println("Конкатенированный массив: ${concatenatedArray.joinToString(", ")}")

// 17. Сумма и произведение
val product = array3.reduce { acc, i -> acc * i }
println("Сумма: ${array3.sum()}, Произведение: $product")

// 18. Группировка чисел
for (i in array3.indices step 5) {
    println("Группа: ${array3.copyOfRange(i, (i + 5).coerceAtMost(array3.size)).joinToString(", ")}")
}

// 19. Слияние двух массивов
val sortedArray1 = intArrayOf(1, 3, 5)
val sortedArray2 = intArrayOf(2, 4, 6)
val mergedArray = (sortedArray1 + sortedArray2).sortedArray()
println("Слитый отсортированный массив: ${mergedArray.joinToString(", ")}")

// 20. Числовая последовательность
val arithmeticProgression = IntArray(10) { it * 2 }
```

```
    println("Арифметическая прогрессия: ${arithmeticProgression.joinToString(", ")}")

    // 21. Удаление элемента
    fun removeElement(array: IntArray, element: Int): IntArray {
        return array.filter { it != element }.toIntArray()
    }
    val newArray = removeElement(array3, 2)
    println("Массив после удаления элемента 2: ${newArray.joinToString(", ")}")

    // 22. Поиск второго максимального
    val secondMax = array3.sortedDescending().distinct().elementAtOrNull(1)
println("Второй по величине элемент: $secondMax")

    // 23. Объединение массивов
    fun mergeArrays(vararg arrays: IntArray): IntArray {
        return arrays.flatten().toIntArray()
    }
    val mergedMultiple = mergeArrays(array3, array4)
    println("Объединённый массив: ${mergedMultiple.joinToString(", ")}")

    // 24. Транспонирование матрицы
    val matrix = arrayOf(
        arrayOf(1, 2, 3),
        arrayOf(4, 5, 6),
        arrayOf(7, 8, 9)
    )
    val transposedMatrix = Array(matrix[0].size) { IntArray(matrix.size) }
    for (i in matrix.indices) {
        for (j in matrix[i].indices) {
            transposedMatrix[j][i] = matrix[i][j]
        }
    }
```

```
    println("Транспонированная матрица:")
    transposedMatrix.forEach { println(it.joinToString(", ")) }

    // 25. Линейный поиск
    fun linearSearch(array: IntArray, element: Int): Boolean {
        return array.contains(element)
    }
    println("Элемент 3 найден в массиве: ${linearSearch(array3, 3)}")

    // 26. Среднее арифметическое
    val average = array3.average()
    println("Среднее арифметическое: $average")

    // 27. Максимальная последовательность
    val maxSequence = array3.fold(mutableListOf<Pair<Int, Int>>()) { acc, i ->
        if (acc.isEmpty() || acc.last().first != i) {
            acc.add(Pair(i, 1))
        } else {
            acc[acc.lastIndex] = Pair(i, acc.last().second + 1)
        }
        acc
    }.maxByOrNull { it.second }
    println("Максимальная последовательность: ${maxSequence?.first} с длиной
${maxSequence?.second}")

    // 28. Ввод и вывод массива
    println("Введите количество элементов массива:")
    val size = readLine()!!.toInt()
    val userArray = IntArray(size)
    println("Введите элементы массива:")
    for (i in userArray.indices) {
        userArray[i] = readLine()!!.toInt()
    }
```

```
    println("Введённый массив: ${userArray.joinToString(", ")}")

    // 29. Нахождение медианы
    fun median(array: IntArray): Double {
        val sorted = array.sortedArray()
        return if (sorted.size % 2 == 0) {
            (sorted[sorted.size / 2 - 1] + sorted[sorted.size / 2]) / 2.0
        } else {
            sorted[sorted.size / 2].toDouble()
        }
    }
    println("Медиана массива: ${median(array3)}")

    // 30. Распределение по группам
    val distributionArray = IntArray(100) { it + 1 }
    for (i in distributionArray.indices step 10) {
        println("Группа: ${distributionArray.copyOfRange(i, (i +
10).coerceAtMost(distributionArray.size)).joinToString(", ")}")
    }
}
```