Задание 1: Создание и вывод элементов

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5)
    println("Элементы массива: ${numbers.joinToString(", ")}")
}
```

Задание 2: Сумма элементов массива

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5)
    val sum = numbers.sum()
    println("Сумма элементов массива: $sum")
}
```

Задание 3: Максимальное и минимальное значение

```
fun main() {
    val numbers = arrayOf(10, 5, 3, 8, 15, 2, 7, 12, 1, 6)
    val max = numbers.maxOrNull()
    val min = numbers.minOrNull()
    println("Максимальное значение: $max")
    println("Минимальное значение: $min")
}
```

Задание 4: Сортировка массива

```
fun main() {
    val numbers = arrayOf(5, 2, 9, 1, 5, 6)
    val sortedNumbers = numbers.sortedArray()
    println("Отсортированный массив: ${sortedNumbers.joinToString(", ")}")
}
```

Задание 5: Уникальные элементы

```
fun main() {
    val numbers = arrayOf(1, 2, 2, 3, 4, 4, 5)
```

```
    val uniqueNumbers = numbers.distinct()
    println("Уникальные элементы: ${uniqueNumbers.joinToString(", ")}")
}
```

Задание 6: Четные и нечетные числа

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5, 6, 7, 8)
    val evenNumbers = numbers.filter { it % 2 == 0 }
    val oddNumbers = numbers.filter { it % 2 != 0 }

    println("Четные числа: ${evenNumbers.joinToString(", ")}")
    println("Нечетные числа: ${oddNumbers.joinToString(", ")}")
}
```

Задание 7: Реверс массива

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5)
    val reversedNumbers = numbers.reversedArray()
    println("Реверсированный массив: ${reversedNumbers.joinToString(", ")}")
}
```

Задание 8: Поиск элемента

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5)
    val elementToFind = 3
    val index = numbers.indexOf(elementToFind)

    if (index != -1) {
        println("Элемент $elementToFind найден на индексе $index")
    } else {
        println("Элемент $elementToFind не найден")
    }
```

```
}
```

Задание 9: Копирование массива

```
fun main() {
    val originalArray = arrayOf(1, 2, 3)
    val copiedArray = originalArray.copyOf()

    println("Скопированный массив: ${copiedArray.joinToString(", ")}")
}
```

Задание 10: Сумма четных чисел

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5)
    val sumOfEvens = numbers.filter { it % 2 == 0 }.sum()

    println("Сумма четных чисел: $sumOfEvens")
}
```

Задание 11: Пересечение массивов

```
fun main() {
    val array1 = arrayOf(1, 2, 3, 4)
    val array2 = arrayOf(3, 4, 5, 6)

    val intersection = array1.intersect(array2.asIterable())

    println("Пересечение массивов: ${intersection.joinToString(", ")}")
}
```

Задание 12: Перестановка элементов

```
fun swapElements(array: IntArray, index1: Int, index2: Int) {
    val temp = array[index1]
    array[index1] = array[index2]
```

```
    array[index2] = temp
}

fun main() {
    val numbers = intArrayOf(1, 2, 3)
    swapElements(numbers, 0, 2)

    println("Массив после перестановки: ${numbers.joinToString(", ")}")
}
```

Задание 13: Заполнение случайными числами

```
import kotlin.random.Random

fun main() {
    val randomNumbers = IntArray(20) { Random.nextInt(1, 101) }

    println("Массив случайных чисел: ${randomNumbers.joinToString(", ")}")
}
```

Задание 14: Числа Прокопенко (делящиеся на 3)

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5, 6, 7, 8, 9)

    val divisibleByThree = numbers.filter { it % 3 == 0 }

    println("Числа делящиеся на 3: ${divisibleByThree.joinToString(", ")}")
}
```

Задание 15: Проверка на палиндром

```
fun isPalindrome(array: IntArray): Boolean {
    return array.contentEquals(array.reversedArray())
```

```
}

fun main() {
    val numbers = intArrayOf(1, 2, 3, 2, 1)

    if (isPalindrome(numbers)) {
        println("Массив является палиндромом")
    } else {
        println("Массив не является палиндромом")
    }
}
```

Задание 16: Конкатенация двух массивов

```
fun main() {
    val array1 = arrayOf(1, 2, 3)
    val array2 = arrayOf(4, 5, 6)
    val concatenatedArray = array1 + array2
    println("Конкатенированный массив: ${concatenatedArray.joinToString(", ")}")
}
```

Задание 17: Сумма и произведение

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5)
    val sum = numbers.sum()
    val product = numbers.reduce { acc, i -> acc * i }
    println("Сумма: $sum, Произведение: $product")
}
```

Задание 18: Группировка чисел

```
fun main() {
    val numbers = (1..20).toList()
    val groupedNumbers = numbers.chunked(5)
```

```
    for ((index, group) in groupedNumbers.withIndex()) {
        println("Группа ${index + 1}: ${group.joinToString(", ")}")
    }
}
```

Задание 19: Слияние двух массивов

```
fun mergeSortedArrays(array1: IntArray, array2: IntArray): IntArray {
    val mergedArray = IntArray(array1.size + array2.size)
    var i = 0
    var j = 0
    var k = 0

    while (i < array1.size && j < array2.size) {
        if (array1[i] < array2[j]) {
            mergedArray[k++] = array1[i++]
        } else {
            mergedArray[k++] = array2[j++]
        }
    }

    while (i < array1.size) {
        mergedArray[k++] = array1[i++]
    }

    while (j < array2.size) {
        mergedArray[k++] = array2[j++]
    }

    return mergedArray
}
```

```
fun main() {
    val array1 = intArrayOf(1, 3, 5)
    val array2 = intArrayOf(2, 4, 6)
    val mergedArray = mergeSortedArrays(array1, array2)

    println("Слияние отсортированных массивов: ${mergedArray.joinToString(", ")}")
}
```

Задание 20: Числовая последовательность

```
fun main() {
    val start = 1
    val difference = 2
    val count = 10
    val arithmeticSequence = IntArray(count) { start + it * difference }

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

Задание 21: Удаление элемента

```
fun removeElement(array: IntArray, elementToRemove: Int): IntArray {
    return array.filter { it != elementToRemove }.toIntArray()
}

fun main() {
    val numbers = intArrayOf(1, 2, 3, 4, 5)
    val elementToRemove = 3
    val updatedArray = removeElement(numbers, elementToRemove)

    println("Массив после удаления элемента $elementToRemove: ${updatedArray.joinToString(", ")}")
}
```

Задание 22: Поиск второго максимального

```
fun main() {
    val numbers = intArrayOf(10, 20, 4, 45, 99)
    val uniqueNumbers = numbers.distinct().sortedDescending()

    if (uniqueNumbers.size < 2) {
        println("Второго максимального элемента нет.")
    } else {
        println("Второй по величине элемент: ${uniqueNumbers[1]}")
    }
}
```

Задание 23: Объединение массивов

```
fun mergeMultipleArrays(vararg arrays: IntArray): IntArray {
    return arrays.flatten().toIntArray()
}

fun main() {
    val array1 = intArrayOf(1, 2)
    val array2 = intArrayOf(3, 4)
    val array3 = intArrayOf(5, 6)

    val mergedArray = mergeMultipleArrays(array1, array2, array3)

    println("Объединенный массив: ${mergedArray.joinToString(", ")}")
}
```

Задание 24: Транспонирование матрицы

```
fun transposeMatrix(matrix: Array<IntArray>): Array<IntArray> {
    val rows = matrix.size
    val cols = matrix[0].size
    val transposedMatrix = Array(cols) { IntArray(rows) }
```

```
    for (i in 0 until rows) {
        for (j in 0 until cols) {
            transposedMatrix[j][i] = matrix[i][j]
        }
    }

    return transposedMatrix
}

fun main() {
    val matrix = arrayOf(
        intArrayOf(1, 2, 3),
        intArrayOf(4, 5, 6),
        intArrayOf(7, 8, 9)
    )

    val transposedMatrix = transposeMatrix(matrix)

    println("Транспонированная матрица:")
    for (row in transposedMatrix) {
        println(row.joinToString(", "))
    }
}
```

Задание 25: Линейный поиск

```
fun linearSearch(array: IntArray, target: Int): Boolean {
    for (element in array) {
        if (element == target) return true
    }
    return false
}
```

```
fun main() {
    val numbers = intArrayOf(1, 2, 3, 4, 5)
    val target = 3

    if (linearSearch(numbers, target)) {
        println("Элемент $target найден в массиве.")
    } else {
        println("Элемент $target не найден в массиве.")
    }
}
```

Задание 26: Среднее арифметическое

```
fun main() {
    val numbers = doubleArrayOf(1.0, 2.0, 3.0, 4.0, 5.0)
    val average = numbers.average()

    println("Среднее арифметическое: $average")
}
```

Задание 27: Максимальная последовательность

```
fun main() {
    val numbers = intArrayOf(1, 1, 2, 3, 3, 3, 4, 4)

    var maxCount = 0
    var currentCount = 1

    for (i in 1 until numbers.size) {
        if (numbers[i] == numbers[i - 1]) {
            currentCount++
        } else {
            maxCount = maxOf(maxCount, currentCount)
            currentCount = 1
```

```
        }
    }

    maxCount = maxOf(maxCount, currentCount)

    println("Максимальная последовательность одинаковых элементов: $maxCount")
}
```

Задание 28: Ввод и вывод массива

```
fun main() {
    println("Введите количество элементов массива:")
    val n = readLine()!!.toInt()

    val numbers = IntArray(n)

    println("Введите элементы массива:")
    for (i in numbers.indices) {
        numbers[i] = readLine()!!.toInt()
    }

    println("Введенный массив: ${numbers.joinToString(", ")}")
}
```

Задание 29: Нахождение медианы

```
fun findMedian(array: IntArray): Double {
    val sortedArray = array.sorted()

    return if (sortedArray.size % 2 == 0) {
        (sortedArray[sortedArray.size / 2 - 1] + sortedArray[sortedArray.size / 2]) / 2.0
    } else {
        sortedArray[sortedArray.size / 2].toDouble()
    }
```

```
}

fun main() {
    val numbers = intArrayOf(3, 1, 4, 2, 5)

    val median = findMedian(numbers)

    println("Медиана массива: $median")
}
```

Задание 30: Распределение по группам

```
fun main() {
    val randomNumbers = IntArray(100) { (1..100).random() }

    for (i in randomNumbers.indices step 10) {
        println("Группа ${i / 10 + 1}: ${randomNumbers.slice(i until minOf(i + 10, randomNumbers.size)).joinToString(", ")}")
    }
}
```