

### Задание 1

```
fun main() {  
    val numbers = intArrayOf(1, 2, 3, 4, 5)  
  
    for (number in numbers) {  
        println(number)  
    }  
}
```

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe  
1  
2  
3  
4  
5  
  
Process finished with exit code 0
```

### Задание 2

```
fun main() {  
    val numbers = intArrayOf(1, 2, 3, 4, 5)  
    var sum = 0  
  
    for (number in numbers) {  
        sum += number  
    }  
  
    println("Сумма элементов массива: $sum")  
}
```

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe  
Сумма элементов массива: 15  
  
Process finished with exit code 0
```

### Задание 3

```

fun main() {
    val numbers = intArrayOf(23, 5, 76, 2, 45, 89, 14, 8, 31, 60)

    val max = numbers.maxOrNull()
    val min = numbers.minOrNull()

    println("Максимальное значение: $max")
    println("Минимальное значение: $min")
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-j
Максимальное значение: 89
Минимальное значение: 2

Process finished with exit code 0

```

#### Задание 4

```

fun main() {
    val numbers = intArrayOf(23, 5, 76, 2, 45, 89, 14, 8, 31, 60)

    // Сортировка пузырьком
    for (i in numbers.indices) {
        for (j in 0 until numbers.size - 1 - i) {
            if (numbers[j] > numbers[j + 1]) {
                // Обмен значений
                val temp = numbers[j]
                numbers[j] = numbers[j + 1]
                numbers[j + 1] = temp
            }
        }
    }

    println("Отсортированный массив: ${numbers.joinToString(separator: ", ")}")
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Prog
Отсортированный массив: 2, 5, 8, 14, 23, 31, 45, 60, 76, 89

Process finished with exit code 0

```

#### Задание 5

```

fun main() {
    val numbers = intArrayOf(4, 5, 4, 1, 2, 1, 3, 3, 5, 6)

    // Используем множество для получения уникальных элементов
    val uniqueNumbers = numbers.toSet()

    println("Уникальные элементы: ${uniqueNumbers.joinToString(separator: ", ")}")
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.
Уникальные элементы: 4, 5, 1, 2, 3, 6

Process finished with exit code 0

```

#### Задание 6

```

fun main() {
    val numbers = intArrayOf(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

    val evenNumbers = mutableListof<Int>()
    val oddNumbers = mutableListof<Int>()

    for (number in numbers) {
        if (number % 2 == 0) {
            evenNumbers.add(number)
        } else {
            oddNumbers.add(number)
        }
    }

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

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.
Четные числа: 2, 4, 6, 8, 10
Нечетные числа: 1, 3, 5, 7, 9

Process finished with exit code 0

```

#### Задание 7

```

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

    val reversedNumbers = numbers.reversedArray()

    println("Исходный массив: ${numbers.joinToString(separator: ", ")}")
    println("Реверсированный массив: ${reversedNumbers.joinToString(separator: ", ")}")
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe
Исходный массив: 1, 2, 3, 4, 5
Реверсированный массив: 5, 4, 3, 2, 1

Process finished with exit code 0

```

#### Задание 8

```

fun main() {
    val numbers = intArrayOf(10, 20, 30, 40, 50)
    val target = 30 // элемент для поиска

    val index = numbers.indexOf(target)

    if (index != -1) {
        println("Элемент $target найден на индексе $index.")
    } else {
        println("Элемент $target не найден в массиве.")
    }
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.
Элемент 30 найден на индексе 2.

Process finished with exit code 0

```

#### Задание 9

```

fun main() {
    val originalArray = arrayOf(1, 2, 3, 4, 5) // Исходный массив
    val newArray = originalArray.copyOf() // Копирование массива

    println("Исходный массив: ${originalArray.joinToString()}")
    println("Новый массив: ${newArray.joinToString()}")
}

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe

Исходный массив: 1, 2, 3, 4, 5

Новый массив: 1, 2, 3, 4, 5

Process finished with exit code 0

#### Задание 10

```

fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5, 6, 7, 8, 9, 10) // Исходный массив
    val sumOfEvens = numbers.filter { it % 2 == 0 }.sum() // Сумма четных чисел

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

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe

Сумма четных чисел: 30

Process finished with exit code 0

#### Задание 11

```

fun main() {
    val array1 = arrayOf(1, 2, 3, 4, 5)
    val array2 = arrayOf(4, 5, 6, 7, 8)

    val intersection = array1.toSet().intersect(array2.toSet()) // Находим пересечение

    println("Пересечение массивов: $intersection")
}

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe

Пересечение массивов: [4, 5]

Process finished with exit code 0

#### Задание 12

```
import kotlin.random.Random

fun main() {
    val size = 20
    val randomArray = IntArray(size) { Random.nextInt(from: 1, until: 101) } // Заполняем массив случайными числами от 1 до 100

    println("Случайные числа в массиве: ${randomArray.joinToString(separator: ", ")}")
}
```

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Commun
Случайные числа в массиве: 41, 32, 6, 77, 52, 71, 41, 48, 62, 81, 15, 38, 6, 64, 64, 40, 35, 90, 19, 32

Process finished with exit code 0
```

### Задание 13

```
import kotlin.random.Random

fun main() {
    val size = 20
    val randomArray = IntArray(size) { Random.nextInt(from: 1, until: 101) }

    println("Случайные числа в массиве: ${randomArray.joinToString(separator: ", ")}")

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

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

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
Случайные числа в массиве: 1, 9, 6, 42, 30, 77, 43, 43, 48, 9, 66, 23, 68, 26, 15, 79, 61, 50, 73, 53
Числа, делящиеся на 3: 9, 6, 42, 30, 48, 9, 66, 15

Process finished with exit code 0
```

### Задание 14

```

fun main() {
    val array = arrayOf(1, 2, 3, 2, 1)

    if (isPalindrome(array)) {
        println("Массив является палиндромом.")
    } else {
        println("Массив не является палиндромом.")
    }
}

fun <T> isPalindrome(array: Array<T>): Boolean {
    val length = array.size
    for (i in 0 until length / 2) {
        if (array[i] != array[length - 1 - i]) {
            return false
        }
    }
    return true
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent
Массив является палиндромом.

Process finished with exit code 0

```

#### Задание 15

```

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

    val concatenatedArray = concatenateArrays(array1, array2)

    println("Соединенный массив: ${concatenatedArray.joinToString(separator: ", ")}")
}

fun <T> concatenateArrays(array1: Array<T>, array2: Array<T>): Array<T> {
    return array1 + array2
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe
Соединенный массив: 1, 2, 3, 4, 5, 6

Process finished with exit code 0

```

#### Задание 16

```

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

    val sum = array.sum()
    val product = array.fold( initial: 1) { acc, value -> acc * value }

    println("Сумма элементов массива: $sum")
    println("Произведение элементов массива: $product")
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe
Сумма элементов массива: 15
Произведение элементов массива: 120

Process finished with exit code 0

```

Задание 17

```

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

    val sum = array.sum()
    val product = array.fold( initial: 1) { acc, value -> acc * value }

    println("Сумма элементов массива: $sum")
    println("Произведение элементов массива: $product")
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe
Сумма элементов массива: 15
Произведение элементов массива: 120

Process finished with exit code 0

```

Задание 18



```
fun main() {  
    val array = (1..25).toList() // Пример массива из 25 элементов  
  
    val grouped = array.chunked(size: 5) // Разделяем на группы по 5 элементов  
  
    for ((index, group) in grouped.withIndex()) {  
        println("Группа ${index + 1}: $group")  
    }  
}
```

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\jav  
Группа 1: [1, 2, 3, 4, 5]  
Группа 2: [6, 7, 8, 9, 10]  
Группа 3: [11, 12, 13, 14, 15]  
Группа 4: [16, 17, 18, 19, 20]  
Группа 5: [21, 22, 23, 24, 25]  
  
Process finished with exit code 0
```

Задание 19

```

fun main() {
    val array1 = intArrayOf(1, 3, 5, 7, 9) // Первый отсортированный массив
    val array2 = intArrayOf(2, 4, 6, 8, 10) // Второй отсортированный массив

    val mergedArray = mergeSortedArrays(array1, array2)

    println("Слитый отсортированный массив: ${mergedArray.joinToString(separator: ", ")}")
}

fun mergeSortedArrays(array1: IntArray, array2: IntArray): IntArray {
    val merged = IntArray(size = 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]) {
            merged[k++] = array1[i++]
        } else {
            merged[k++] = array2[j++]
        }
    }

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

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

    return merged
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Pro
Слитый отсортированный массив: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

```

Задание 20

```

fun main() {
    val start = 1 // Начальное значение
    val difference = 2 // Разность
    val size = 10 // Количество элементов прогрессии

    val arithmeticProgression = createArithmeticProgression(start, difference, size)

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

fun createArithmeticProgression(start: Int, difference: Int, size: Int): IntArray {
    val progression = IntArray(size)
    for (i in 0 until size) {
        progression[i] = start + i * difference
    }
    return progression
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent
Арифметическая прогрессия: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19

Process finished with exit code 0

```

#### Задание 21

```

fun main() {
    val originalArray = intArrayOf(1, 2, 3, 4, 5, 3, 6)
    val elementToRemove = 3

    val updatedArray = removeElement(originalArray, elementToRemove)

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

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

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent
Массив после удаления элемента 3: 1, 2, 4, 5, 6

Process finished with exit code 0

```

#### Задание 22

```

fun main() {
    val array = intArrayOf(3, 5, 7, 2, 8, 1, 8)

    val secondMax = findSecondMax(array)

    if (secondMax != null) {
        println("Второй по величине элемент: $secondMax")
    } else {
        println("В массиве недостаточно уникальных элементов.")
    }
}

fun findSecondMax(array: IntArray): Int? {
    val uniqueElements = array.distinct()

    return when {
        uniqueElements.size < 2 -> null
        else -> uniqueElements.sortedDescending().let { it[1] }
    }
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java
Второй по величине элемент: 7

Process finished with exit code 0

```

### Задание 23

```

fun main() {
    val array1 = intArrayOf(1, 2, 3)
    val array2 = intArrayOf(4, 5, 6)
    val array3 = intArrayOf(7, 8)

    val resultArray = mergeArrays(array1, array2, array3)

    println("Объединенный массив: ${resultArray.joinToString(separator: ", ")}")
}

fun mergeArrays(vararg arrays: IntArray): IntArray {
    return arrays.flatMap { it.toList() }.toIntArray()
}

```

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe
Объединенный массив: 1, 2, 3, 4, 5, 6, 7, 8

Process finished with exit code 0
```

#### Задание 24

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

    println("Исходная матрица:")
    printMatrix(matrix)

    val transposedMatrix = transposeMatrix(matrix)

    println("Транспонированная матрица:")
    printMatrix(transposedMatrix)
}

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

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

    return transposed
}

fun printMatrix(matrix: Array<IntArray>) {
    for (row in matrix) {
        println(row.joinToString(separator: " "))
    }
}
```

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe
Исходная матрица:
1 2 3
4 5 6
7 8 9
Транспонированная матрица:
1 4 7
2 5 8
3 6 9

Process finished with exit code 0
|
```

#### Задание 25

```
fun main() {
    val array = arrayOf(10, 20, 30, 40, 50)
    val target = 30

    val found = linearSearch(array, target)

    if (found) {
        println("Элемент $target найден в массиве.")
    } else {
        println("Элемент $target не найден в массиве.")
    }
}

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

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe
Элемент 30 найден в массиве.

Process finished with exit code 0
```

#### Задание 26

```
fun main() {  
    val array = arrayOf(10, 20, 30, 40, 50)  
  
    val average = calculateAverage(array)  
  
    println("Среднее арифметическое: $average")  
}  
  
fun calculateAverage(array: Array<Int>): Double {  
    if (array.isEmpty()) return 0.0  
  
    val sum = array.sum()  
    return sum.toDouble() / array.size  
}
```

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe  
Среднее арифметическое: 30.0  
  
Process finished with exit code 0
```

Задание 27

```

fun main() {
    val array = arrayOf(1, 2, 2, 3, 3, 3, 2, 2, 2, 1)

    val maxSequence = findMaxSequence(array)

    println("Максимальная последовательность одинаковых элементов: $maxSequence")
}

fun findMaxSequence(array: Array<Int>): Pair<Int, Int> {
    if (array.isEmpty()) return Pair(0, 0)

    var maxCount = 1
    var currentCount = 1
    var currentElement = array[0]

    for (i in 1 until array.size) {
        if (array[i] == array[i - 1]) {
            currentCount++
        } else {
            if (currentCount > maxCount) {
                maxCount = currentCount
                currentElement = array[i - 1]
            }
            currentCount = 1
        }
    }

    if (currentCount > maxCount) {
        maxCount = currentCount
        currentElement = array[array.size - 1]
    }

    return Pair(currentElement, maxCount)
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Program
Максимальная последовательность одинаковых элементов: (3, 3)

Process finished with exit code 0
|

```

Задание 28



```

fun main() {
    println("Введите числа через пробел:")

    // Читаем строку ввода от пользователя
    val input = readLine()

    // Преобразуем строку в массив чисел
    val numbers = input?.split(...delimiters: " ")?.mapNotNull { it.toIntOrNull() }?.toTypedArray()

    // Проверяем, если массив не пустой
    if (numbers != null && numbers.isNotEmpty()) {
        println("Вы ввели массив чисел:")
        println(numbers.joinToString(separator: ", "))
    } else {
        println("Вы не ввели корректные числа.")
    }
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\j
Введите числа через пробел:
3 5 9
Вы ввели массив чисел:
3, 5, 9

Process finished with exit code 0

```

Задание 29

```

fun main() {
    println("Введите числа через пробел:")

    // Читаем строку ввода от пользователя
    val input = readLine()

    // Преобразуем строку в массив чисел
    val numbers = input?.split(...delimiters: " ")?.mapNotNull { it.toIntOrNull() }?.toList()

    // Проверяем, если массив не пустой
    if (numbers != null && numbers.isNotEmpty()) {
        // Сортируем массив
        val sortedNumbers = numbers.sorted()
        val size = sortedNumbers.size
        val median: Double

        // Находим медиану
        median = if (size % 2 == 0) {
            // Если четное количество элементов, медиана - среднее двух средних
            (sortedNumbers[size / 2 - 1] + sortedNumbers[size / 2]) / 2.0
        } else {
            // Если нечетное количество элементов, медиана - средний элемент
            sortedNumbers[size / 2].toDouble()
        }

        println("Медиана массива: $median")
    } else {
        println("Вы не ввели корректные числа.")
    }
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\
Введите числа через пробел:
9 9 1
Медиана массива: 9.0

Process finished with exit code 0

```

Задание 30

```

import kotlin.random.Random

fun main() {
    // Создаем массив из 100 случайных целых чисел от 0 до 99
    val numbers = IntArray(size: 100) { Random.nextInt(from: 0, until: 100) }

    // Выводим оригинальный массив
    println("Оригинальный массив:")
    println(numbers.joinToString(separator: ", "))

    // Делим на 10 групп по 10 элементов
    val groups = numbers.toList().chunked(size: 10)

    // Выводим результаты
    println("\nРаспределение по группам:")
    for (i in groups.indices) {
        println("Группа ${i + 1}: ${groups[i].joinToString(separator: ", ")}")
    }
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaag
Оригинальный массив:
80, 82, 38, 14, 31, 61, 50, 77, 37, 79, 3, 1, 21, 96, 7, 32

Распределение по группам:
Группа 1: 80, 82, 38, 14, 31, 61, 50, 77, 37, 79
Группа 2: 3, 1, 21, 96, 7, 32, 99, 13, 25, 48
Группа 3: 97, 69, 73, 79, 83, 38, 88, 60, 14, 47
Группа 4: 25, 31, 26, 7, 39, 32, 53, 64, 7, 12
Группа 5: 28, 31, 17, 65, 81, 32, 86, 7, 81, 44
Группа 6: 52, 27, 76, 90, 41, 30, 85, 9, 7, 11
Группа 7: 29, 28, 32, 17, 5, 56, 77, 99, 69, 22
Группа 8: 31, 89, 51, 49, 15, 37, 13, 21, 22, 61
Группа 9: 89, 69, 11, 75, 32, 8, 92, 15, 28, 71
Группа 10: 52, 2, 27, 81, 14, 7, 41, 51, 54, 69

Process finished with exit code 0

```