

Массивы

1. Создание и вывод элементов

```
fun main() {  
    val array = listOf(1, 2, 3, 4, 5)  
    println("Элементы массива: $array")  
}
```

Элементы массива: [1, 2, 3, 4, 5]

Process finished with exit code 0

2. Сумма элементов массива

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

Сумма элементов массива: 15

3. Максимальное и минимальное значение

```
fun main() {  
    val array = listOf(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)  
    println("Максимальное значение: ${array.maxOrNull()}")  
    println("Минимальное значение: ${array.minOrNull()}")  
}
```

Максимальное значение: 10

Минимальное значение: 1

4. Сортировка массива

```
fun main() {  
    val array = arrayOf(1, 5, 2, 3, 4)  
    val sortedArray = array.sorted()  
    println("Отсортированный массив: $sortedArray")  
}
```

Отсортированный массив: [1, 2, 3, 4, 5]

5. Уникальные элементы

```
fun main() {  
    val array = listOf(1, 1, 2, 2, 3, 4, 4, 5)  
    val uniqueElements = array.distinct()  
    println("Уникальные элементы: $uniqueElements")  
}
```

Уникальные элементы: [1, 2, 3, 4, 5]

6. Четные и нечетные числа

```
fun main() {  
    val list2 = listOf(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)  
    val evenNumbers = list2.filter { it % 2 == 0 }  
    val oddNumbers = list2.filter { it % 2 != 0 }  
    println("Четные числа: $evenNumbers")  
    println("Нечетные числа: $oddNumbers")  
}
```

```
Четные числа: [2, 4, 6, 8, 10]
Нечетные числа: [1, 3, 5, 7, 9]
```

7. Реверс массива

```
fun main() {
    val array = listOf(1, 2, 3, 4, 5)
    val reversedArray = array.reversed()
    println("Реверсированный массив: $reversedArray")
}
```

Реверсированный массив: [5, 4, 3, 2, 1]

8. Поиск элемента

```
fun main() {
    val element = 4
    val array = listOf(1, 2, 3, 4, 5)
    val index = array.indexOf(element)
    println("Индекс элемента $element: $index")
}
```

Индекс элемента 4: 3

10. Сумма четных чисел

```
fun main() {
    val originalArray = arrayOf(10, 20, 30, 40, 50)
    val copiedArray = originalArray.copyOf()
    println("Исходный массив: ${originalArray.joinToString()}")
    println("Скопированный массив: ${copiedArray.joinToString()}")
}
```

Исходный массив: 10, 20, 30, 40, 50
Скопированный массив: 10, 20, 30, 40, 50

11. Пересечение массивов

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
    val sumOfEvenNumbers = numbers.filter { it % 2 == 0 }.sum()
    println("Сумма четных чисел: $sumOfEvenNumbers")
}
```

Сумма четных чисел: 6

12. Перестановка элементов

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5)
    val index1 = 1
    val index2 = 3
    val temp = numbers[index1]
    numbers[index1] = numbers[index2]
    numbers[index2] = temp
    println("Массив после перестановки: ${numbers.joinToString()}")
}
```

Массив после перестановки: 1, 4, 3, 2, 5

13. Заполнение случайными числами

```
fun main() {
    val randomNumbers = Array(20) { (1..100).random() }
    println("Массив случайных чисел: ${randomNumbers.joinToString()}")
}
```

Массив случайных чисел: 52, 32, 83, 21, 48, 56, 90, 14, 48, 90, 14, 71, 44, 82, 79, 5, 26, 100, 10, 75

14. Числа, делящиеся на 3

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
    val divisibleBy3 = numbers.filter { it % 3 == 0 }
}
```

```
println("Числа, делящиеся на 3: ${divisibleBy3.joinToString()}")
}
```

Числа, делящиеся на 3: 3, 6, 9

15. Проверка на палиндром

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 2, 1)
    var isPalindrome = true
    for (i in 0 until numbers.size / 2) {
        if (numbers[i] != numbers[numbers.size - 1 - i]) {
            isPalindrome = false
            break
        }
    }
    println("Массив является палиндромом: $isPalindrome")
}
```

Массив является палиндромом: true

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()}")
}
```

Конкатенированный массив: 1, 2, 3, 4, 5, 6

17. Сумма и произведение

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

Сумма: 15
Произведение: 120

18. Группировка чисел

```
fun main() {
    val numbers = Array(20) { it + 1 }
    val grouped = numbers.toList().chunked(5)
    println("Группировка по 5 элементов:")
    grouped.forEachIndexed { index, group ->
        println("Группа ${index + 1}: ${group.joinToString()}")
    }
}
```

Группировка по 5 элементов:
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

19. Слияние двух массивов

```
fun main() {
    val array1 = arrayOf(1, 2, 3)
```

```

val array2 = arrayOf(4, 5, 6)
val mergedArray = array1 + array2
println("Слияние двух массивов: ${mergedArray.joinToString()}")
}
Слияние двух массивов: 1, 2, 3, 4, 5, 6

```

20. Числовая последовательность (арифметическая прогрессия)

```

fun main() {
    val sequence = Array(10) { it * 2 }
    println("Арифметическая прогрессия: ${sequence.joinToString()}")
}
Арифметическая прогрессия: 0, 2, 4, 6, 8, 10, 12, 14, 16, 18

```

21: Удаление элемента

```

fun main() {
    val array = arrayOf(1, 2, 3, 4, 5)
    val elementToRemove = 3
    val newArray = array.filter { it != elementToRemove }
    .toTypedArray()
    println("Массив после удаления элемента $elementToRemove: ${newArray.joinToString()}")
}
Массив после удаления элемента 3: 1, 2, 4, 5

```

22: Поиск второго максимального элемента

```

fun main() {
    val array = arrayOf(30, 40, 50, 60, 70)
    val secondMax = array.sortedDescending()[1]
    println("Второй максимальный элемент: $secondMax")
}
Второй максимальный элемент: 60

```

23: Объединение массивов

```

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

```

24: Транспонирование матрицы

```

fun main() {
    val matrix = arrayOf(
        arrayOf(1, 2, 3),
        arrayOf(4, 5, 6),
        arrayOf(7, 8, 9)
    )
    val transposedMatrix = matrix[0].indices.map { col -> matrix.map { it[col] } }
    println("Транспонированная матрица: $transposedMatrix")
}

```

```
Транспонированная матрица: [[1, 4, 7], [2, 5, 8], [3, 6, 9]]
```

25: Линейный поиск

```
fun main() {  
    val array = arrayOf(10, 20, 30, 40, 50)  
    val element = 20  
    val isFound = element in array  
    println("Элемент $element найден: $isFound")  
}
```

```
Элемент 20 найден: true
```

26: Среднее арифметическое

```
fun main() {  
    val array = arrayOf(10, 20, 30, 40, 50)  
    val average = array.average()  
    println("Среднее арифметическое: $average")  
}
```

```
Среднее арифметическое: 30.0
```

27: Максимальная последовательность одинаковых элементов

```
fun main() {  
    val array = arrayOf(1, 1, 2, 2, 2, 3, 3, 3, 3)  
    val maxSequence = array.groupBy { it }.maxByOrNull { it.value.size }?.key  
    println("Максимальная последовательность: $maxSequence")  
}
```

```
Максимальная последовательность: 3
```

28: Ввод и вывод массива

```
fun main() {  
    println("Введите массив чисел через пробел:")  
    val inputArray = readLine()?.split(" ")?.map { it.toInt() }  
    ?.toTypedArray() ?: emptyArray()  
    println("Введенный массив: ${inputArray.joinToString()}")  
}
```

```
Введите массив чисел через пробел:
```

```
1 3 4 2 5
```

```
Введенный массив: 1, 3, 4, 2, 5
```

29: Нахождение медианы

```
fun main() {  
    val array = arrayOf(1, 3, 2, 5, 4).sortedArray()  
    val median = if (array.size % 2 == 0) {  
        (array[array.size / 2 - 1] + array[array.size / 2]) / 2.0  
    } else {  
        array[array.size / 2].toDouble()  
    }  
    println("Медиана: $median")  
}
```

```
C:\Users\USER\.jdk
```

```
Медиана: 3.0
```

30: Распределение по группам

```
fun main() {  
    val array = Array(100) { it + 1 }
```

```
val grouped = array.toList().chunked(10)
println("Группировка по 10 элементов:")
grouped.forEachIndexed { index, group ->
    println("Группа ${index + 1}: ${group.joinToString()}")
}
}
```

Группировка по 10 элементов:

Группа 1: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Группа 2: 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
Группа 3: 21, 22, 23, 24, 25, 26, 27, 28, 29, 30
Группа 4: 31, 32, 33, 34, 35, 36, 37, 38, 39, 40
Группа 5: 41, 42, 43, 44, 45, 46, 47, 48, 49, 50
Группа 6: 51, 52, 53, 54, 55, 56, 57, 58, 59, 60
Группа 7: 61, 62, 63, 64, 65, 66, 67, 68, 69, 70
Группа 8: 71, 72, 73, 74, 75, 76, 77, 78, 79, 80
Группа 9: 81, 82, 83, 84, 85, 86, 87, 88, 89, 90
Группа 10: 91, 92, 93, 94, 95, 96, 97, 98, 99, 100