


```
fun main() { new *
    val array = arrayOf(1, 2, 3, 4, 5)
    println("Массив: ${array.joinToString()}")
}
```

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

```
fun main() { new *
    val array = arrayOf(10, 20, 5, 30, 15)
    println("Максимальное значение: ${array.maxOrNull()}")
    println("Минимальное значение: ${array.minOrNull()}")
}
```

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

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

```
fun main() { new *
    val array = arrayOf(1, 2, 3, 4, 5, 6)
    val even = array.filter { it % 2 == 0 }
    val odd = array.filter { it % 2 != 0 }
    println("Четные: ${even.joinToString()}")
    println("Нечетные: ${odd.joinToString()}")
}
```

```
fun main() { new *
    val array = arrayOf(1, 2, 3, 4, 5)
    val reversedArray = array.reversedArray()
    println("Реверс массива: ${reversedArray.joinToString()}")
}
```

```
fun main() { new *
    val array = arrayOf(10, 20, 30, 40, 50)
    val element = 30
    val index = array.indexOf(element)
    println("Индекс элемента $element: $index")
}
```

```
fun main() { new *
    val originalArray = arrayOf(1, 2, 3, 4, 5)
    val copiedArray = originalArray.copyOf()
    println("Скопированный массив: ${copiedArray.joinToString()}")
}
```

```
fun main() { new *
    val array = arrayOf(1, 2, 3, 4, 5, 6)
    val sumEven = array.filter { it % 2 == 0 }.sum()
    println("Сумма четных чисел: $sumEven")
}
```

```
fun main() { new *
    val array1 = arrayOf(1, 2, 3, 4, 5)
    val array2 = arrayOf(4, 5, 6, 7, 8)
    val intersection = array1.intersect(array2.toList())
    println("Пересечение массивов: ${intersection.joinToString()}")
}
```

```
fun main() { new *
    val array = arrayOf(1, 2, 3, 4, 5)
    array[1] = array[2].also { array[2] = array[1] }
    println("Массив после перестановки: ${array.joinToString()}")
}
```

```
import kotlin.random.Random
```

```
fun main() { new *
    val array = IntArray( size: 20 ) { Random.nextInt( from: 1, until: 101 ) }
    println("Массив случайных чисел: ${array.joinToString()}")
}
```

```
fun main() { new *
    val array = arrayOf(3, 5, 6, 9, 10, 12)
    val divisibleBy3 = array.filter { it % 3 == 0 }
    println("Числа, делящиеся на 3: ${divisibleBy3.joinToString()}")
}
```

```
fun main() { new *
    val array = arrayOf(1, 2, 3, 2, 1)
    val isPalindrome = array.contentEquals(array.reversedArray())
    println("Массив является палиндромом: $isPalindrome")
}
```

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

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

```
fun main() { new *
    val array = (1..25).toList().toTypedArray()
    val groups = array.toList().chunked { size: 5 }
    groups.forEachIndexed { index, group ->
        println("Группа ${index + 1}: ${group.joinToString()}")
    }
}
```

```
fun main() { new *
    val array1 = arrayOf(1, 3, 5)
    val array2 = arrayOf(2, 4, 6)
    val mergedArray = (array1 + array2).sortedArray()
    println("Слияние массивов: ${mergedArray.joinToString()}")
}
```

```
fun main() { new *
    val array = IntArray { size: 10 } { it * 2 } // Арифметическая прогрессия с шагом 2
    println("Числовая последовательность: ${array.joinToString()}")
}
```

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

```
fun main() { new *
    val array = arrayOf(10, 20, 5, 30, 15)
    val secondMax = array.sortedDescending()[1]
    println("Второй максимальный элемент: $secondMax")
}
```



```
fun main() { new *
    val array1 = arrayOf(1, 2, 3)
    val array2 = arrayOf(4, 5, 6)
    val array3 = arrayOf(7, 8, 9)
    val mergedArray = array1 + array2 + array3
    println("Объединенный массив: ${mergedArray.joinToString()}")
}
```

```
fun main() { new *
    val matrix = arrayOf(
        arrayOf(1, 2, 3),
        arrayOf(4, 5, 6),
        arrayOf(7, 8, 9)
    )
    val transposed = Array(matrix[0].size) { i -> Array(matrix.size) { j -> matrix[j][i] } }
    println("Транспонированная матрица:")
    transposed.forEach { row -> println(row.joinToString()) }
}
```

```
fun main() { new *
    val array = arrayOf(1, 2, 3, 4, 5)
    val element = 3
    val isFound = array.contains(element)
    println("Элемент $element найден: $isFound")
}
```

```
fun main() { new *
    val array = arrayOf(1, 2, 3, 4, 5)
    val average = array.average()
    println("Среднее арифметическое: $average")
}
```

```
fun main() { new *
    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")
}
```

```
fun main() { new *
    println("Введите элементы массива через пробел:")
    val input = readLine()!!
    val array = input.split( ...delimiters: " ").map { it.toInt() }.toTypedArray()
    println("Ваш массив: ${array.joinToString()}")
}
```

```
fun main() { new *  
    val array = arrayOf(1, 3, 2, 5, 4)  
    val sortedArray = array.sorted()  
    val median = sortedArray[sortedArray.size / 2]  
    println("Медиана: $median")  
}
```

```
fun main() { new *  
    val array = (1..100).toList().toTypedArray()  
    val groups = array.toList().chunked( size: 10)  
    groups.forEachIndexed { index, group ->  
        println("Група ${index + 1}: ${group.joinToString()}")  
    }  
}
```