

// 1. Создание и вывод элементов

```
fun main() {  
    val array1 = arrayOf(1, 2, 3, 4, 5)  
    println("1. Массив: ${array1.joinToString()}")  
}
```

// 2. Сумма элементов массива

```
fun main() {  
    val array1 = arrayOf(1, 2, 3, 4, 5)  
    val sum = array1.sum()  
    println("2. Сумма элементов: $sum")  
}
```

// 3. Максимальное и минимальное значение

```
fun main() {  
    val array2 = arrayOf(10, 20, 5, 30, 15, 25, 40, 35, 50, 45)  
    val max = array2.maxOrNull()  
    val min = array2.minOrNull()  
    println("3. Максимальное значение: $max, Минимальное значение: $min")  
}
```

// 4. Сортировка массива

```
fun main() {  
    val array2 = arrayOf(10, 20, 5, 30, 15, 25, 40, 35, 50, 45)  
    val sortedArray = array2.sortedArray()  
    println("4. Отсортированный массив: ${sortedArray.joinToString()}")  
}
```

// 5. Уникальные элементы

```
fun main() {  
    val array3 = arrayOf(1, 2, 2, 3, 4, 4, 5)  
    val uniqueElements = array3.distinct()
```

```

        println("5. Уникальные элементы: ${uniqueElements.joinToString()}")
    }

// 6. Четные и нечетные числа
fun main() {
    val array3 = arrayOf(1, 2, 2, 3, 4, 4, 5)
    val evenNumbers = array3.filter { it % 2 == 0 }.toTypedArray()
    val oddNumbers = array3.filter { it % 2 != 0 }.toTypedArray()

    println("6. Четные числа: ${evenNumbers.joinToString()}, Нечетные числа: ${oddNumbers.joinToString()}")
}

// 7. Реверс массива
fun main() {
    val array3 = arrayOf(1, 2, 2, 3, 4, 4, 5)
    val reversedArray = array3.reversedArray()
    println("7. Реверсированный массив: ${reversedArray.joinToString()}")
}

// 8. Поиск элемента
fun main() {
    val array3 = arrayOf(1, 2, 2, 3, 4, 4, 5)
    val searchElement = 3
    val index = array3.indexOf(searchElement)
    println("8. Индекс элемента $searchElement: $index")
}

// 9. Копирование массива
fun main() {
    val array3 = arrayOf(1, 2, 2, 3, 4, 4, 5)
    val copiedArray = array3.copyOf()
    println("9. Скопированный массив: ${copiedArray.joinToString()}")
}

```

// 10. Сумма четных чисел

```
fun main() {  
    val array3 = arrayOf(1, 2, 2, 3, 4, 4, 5)  
    val sumEven = array3.filter { it % 2 == 0 }.sum()  
    println("10. Сумма четных чисел: $sumEven")  
}
```

// 11. Пересечение массивов

```
fun main() {  
    val array3 = arrayOf(1, 2, 2, 3, 4, 4, 5)  
    val array4 = arrayOf(3, 4, 5, 6, 7)  
    val intersection = array3.intersect(array4.toSet())  
    println("11. Пересечение массивов: ${intersection.joinToString()}")  
}
```

// 12. Перестановка элементов

```
fun main() {  
    val array5 = arrayOf(1, 2, 3, 4, 5)  
    array5[0] = array5[1].also { array5[1] = array5[0] }  
    println("12. Массив после перестановки: ${array5.joinToString()}")  
}
```

// 13. Заполнение случайными числами

```
fun main() {  
    val randomArray = IntArray(20) { (1..100).random() }  
    println("13. Массив случайных чисел: ${randomArray.joinToString()}")  
}
```

// 14. Числа Прокопенко (делящиеся на 3)

```
fun main() {  
    val randomArray = IntArray(20) { (1..100).random() }
```

```
val divisibleBy3 = randomArray.filter { it % 3 == 0 }  
println("14. Числа, делящиеся на 3: ${divisibleBy3.joinToString()}")  
}
```

// 15. Проверка на палиндром

```
fun main() {  
    val array3 = arrayOf(1, 2, 2, 3, 4, 4, 5)  
    val isPalindrome = array3.contentEquals(array3.reversedArray())  
    println("15. Массив является палиндромом: $isPalindrome")  
}
```

// 16. Конкатенация двух массивов

```
fun main() {  
    val array3 = arrayOf(1, 2, 2, 3, 4, 4, 5)  
    val array4 = arrayOf(3, 4, 5, 6, 7)  
    val concatenatedArray = array3 + array4  
    println("16. Конкатенированный массив: ${concatenatedArray.joinToString()}")  
}
```

// 17. Сумма и произведение

```
fun main() {  
    val array3 = arrayOf(1, 2, 2, 3, 4, 4, 5)  
    val sum = array3.sum()  
    val product = array3.fold(1) { acc, i -> acc * i }  
    println("17. Сумма: $sum, Произведение: $product")  
}
```

// 18. Группировка чисел

```
fun main() {  
    val randomArray = IntArray(20) { (1..100).random() }  
    val grouped = randomArray.toList().chunked(5)  
    println("18. Группировка по 5 элементов: $grouped")  
}
```

```
}
```

```
// 19. Слияние двух отсортированных массивов
```

```
fun main() {  
    val sortedArray1 = arrayOf(1, 3, 5, 7)  
    val sortedArray2 = arrayOf(2, 4, 6, 8)  
    val mergedArray = (sortedArray1 + sortedArray2).sortedArray()  
    println("19. Слияние двух отсортированных массивов: ${mergedArray.joinToString()}")  
}
```

```
// 20. Числовая последовательность (арифметическая прогрессия)
```

```
fun main() {  
    val arithmeticArray = IntArray(10) { it * 2 + 1 }  
    println("20. Арифметическая прогрессия: ${arithmeticArray.joinToString()}")  
}
```

```
// 21. Удаление элемента
```

```
fun main() {  
    val array6 = arrayOf(1, 2, 3, 4, 5).toMutableList()  
    array6.remove(3)  
    println("21. Массив после удаления элемента 3: ${array6.joinToString()}")  
}
```

```
// 22. Поиск второго максимального
```

```
fun main() {  
    val array2 = arrayOf(10, 20, 5, 30, 15, 25, 40, 35, 50, 45)  
    val secondMax = array2.sortedArrayDescending()[1]  
    println("22. Второй максимальный элемент: $secondMax")  
}
```

```
// 23. Объединение массивов
```

```
fun main() {
```

```

val array3 = arrayOf(1, 2, 2, 3, 4, 4, 5)
val array4 = arrayOf(3, 4, 5, 6, 7)
val combinedArray = arrayOf(array3, array4).flatten().toTypedArray()
println("23. Объединенный массив: ${combinedArray.joinToString()}")
}

// 24. Транспонирование матрицы
fun main() {
    val matrix = arrayOf(arrayOf(1, 2, 3), arrayOf(4, 5, 6))
    val transposedMatrix = matrix[0].indices.map { i -> matrix.map { it[i] }.toTypedArray() }.toTypedArray()
    println("24. Транспонированная матрица: ${transposedMatrix.joinToString { it.joinToString() }}")
}

// 25. Линейный поиск
fun main() {
    val array3 = arrayOf(1, 2, 2, 3, 4, 4, 5)
    val found = array3.contains(5)
    println("25. Элемент 5 найден: $found")
}

// 26. Среднее арифметическое
fun main() {
    val array3 = arrayOf(1, 2, 2, 3, 4, 4, 5)
    val average = array3.average()
    println("26. Среднее арифметическое: $average")
}

// 27. Максимальная последовательность
fun main() {
    val array7 = arrayOf(1, 1, 1, 2, 2, 3, 3, 3, 3, 4)
    val maxSequence = array7.fold(Pair(0, 0)) { (maxCount, currentCount), i ->
        if (i == 3) Pair(maxOf(maxCount, currentCount + 1), currentCount + 1) else Pair(maxCount, 0)
    }
}

```

```

    }.first

    println("27. Максимальная последовательность: $maxSequence")
}

// 28. Ввод и вывод массива
fun main() {
    println("28. Введите массив чисел через пробел:")
    val userArray = readLine()?.split(" ")??.map { it.toInt() }?.toTypedArray() ?: arrayOf()
    println("Введенный массив: ${userArray.joinToString()}")
}

// 29. Нахождение медианы
fun main() {
    val sortedArray = arrayOf(1, 2, 3, 4, 5).sortedArray()
    val median = if (sortedArray.size % 2 == 0) (sortedArray[sortedArray.size / 2 - 1] +
sortedArray[sortedArray.size / 2]) / 2.0 else sortedArray[sortedArray.size / 2]
    println("29. Медиана: $median")
}

// 30. Распределение по группам
fun main() {
    val bigArray = IntArray(100) { it + 1 }
    val groups = bigArray.toList().chunked(10)
    println("30. Группы по 10 элементов: $groups")
}

```