

## Лабораторная работа №6

Харахардин В. ИС233

The screenshot shows the IntelliJ IDEA IDE with a Kotlin file named `Main.kt`. The code performs several operations on arrays: generating a random array, calculating the sum of elements, finding the maximum and minimum values, sorting the array, finding unique elements, partitioning into even and odd numbers, and reversing the array. The Run console shows the output of these operations.

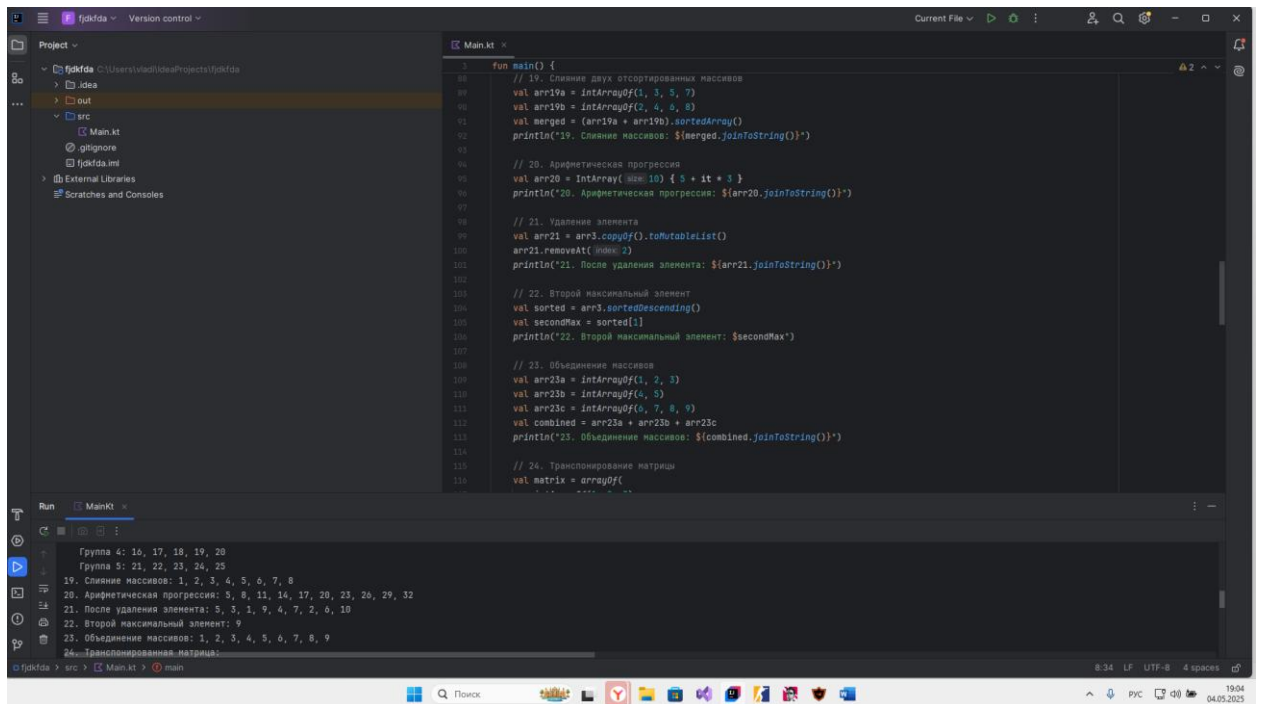
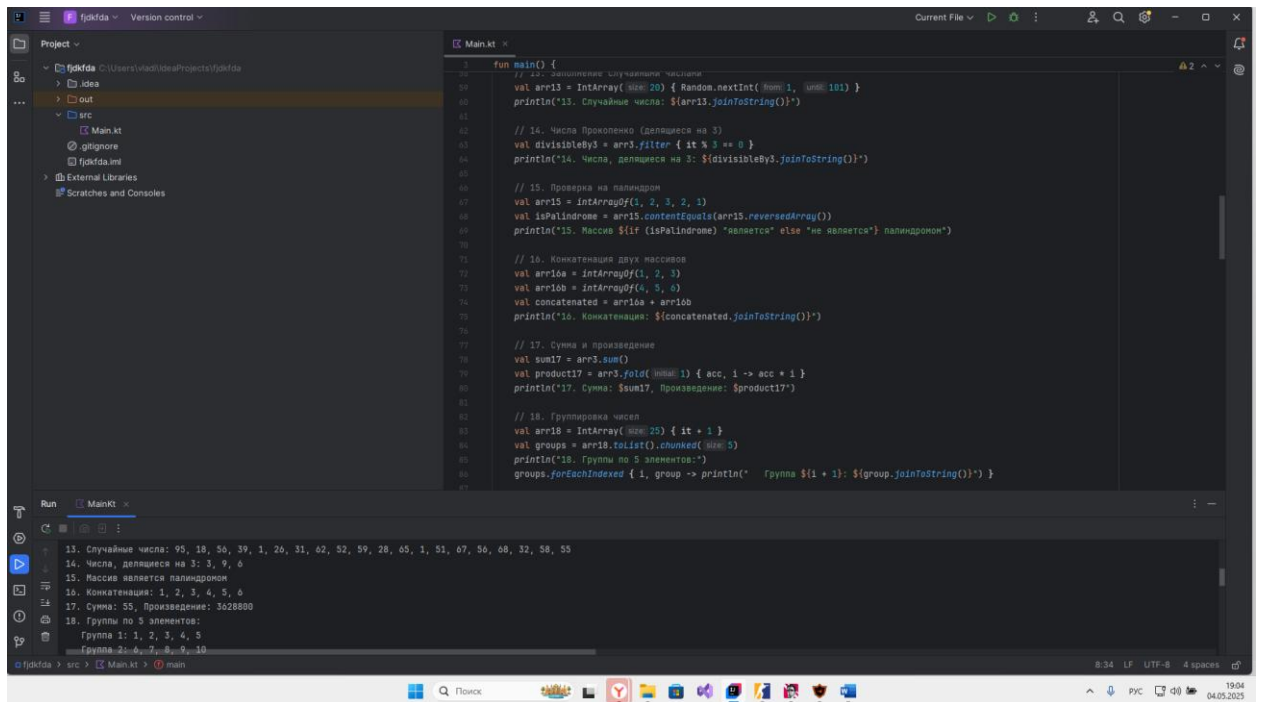
```
1 import kotlin.random.Random
2
3 fun main() {
4     // 1. Создание и вывод элементов
5     val arr1 = IntArrayOf(5, 1, 8, 1, 9)
6     println("1. Массив: ${arr1.joinToString()}")
7
8     // 2. Сумма элементов массива
9     val sum = arr1.sum()
10    println("2. Сумма элементов: $sum")
11
12    // 3. Максимальное и минимальное значение
13    val arr3 = IntArrayOf(5, 3, 8, 1, 9, 4, 7, 2, 6, 10)
14    println("3. Максимальное: ${arr3.maxOrNull()}, Минимальное: ${arr3.minOrNull()}")
15
16    // 4. Сортировка массива
17    val arr4 = arr3.copyOf()
18    arr4.sort()
19    println("4. Отсортированный массив: ${arr4.joinToString()}")
20
21    // 5. Уникальные элементы
22    val arr5 = IntArrayOf(1, 2, 2, 3, 4, 4, 5)
23    val unique = arr5.distinct()
24    println("5. Уникальные элементы: ${unique.joinToString()}")
25
26    // 6. Четные и нечетные числа
27    val (even, odd) = arr3.partition { it % 2 == 0 }
28    println("6. Четные: ${even.joinToString()}, Нечетные: ${odd.joinToString()}")
29
30    // 7. Реверс массива
```

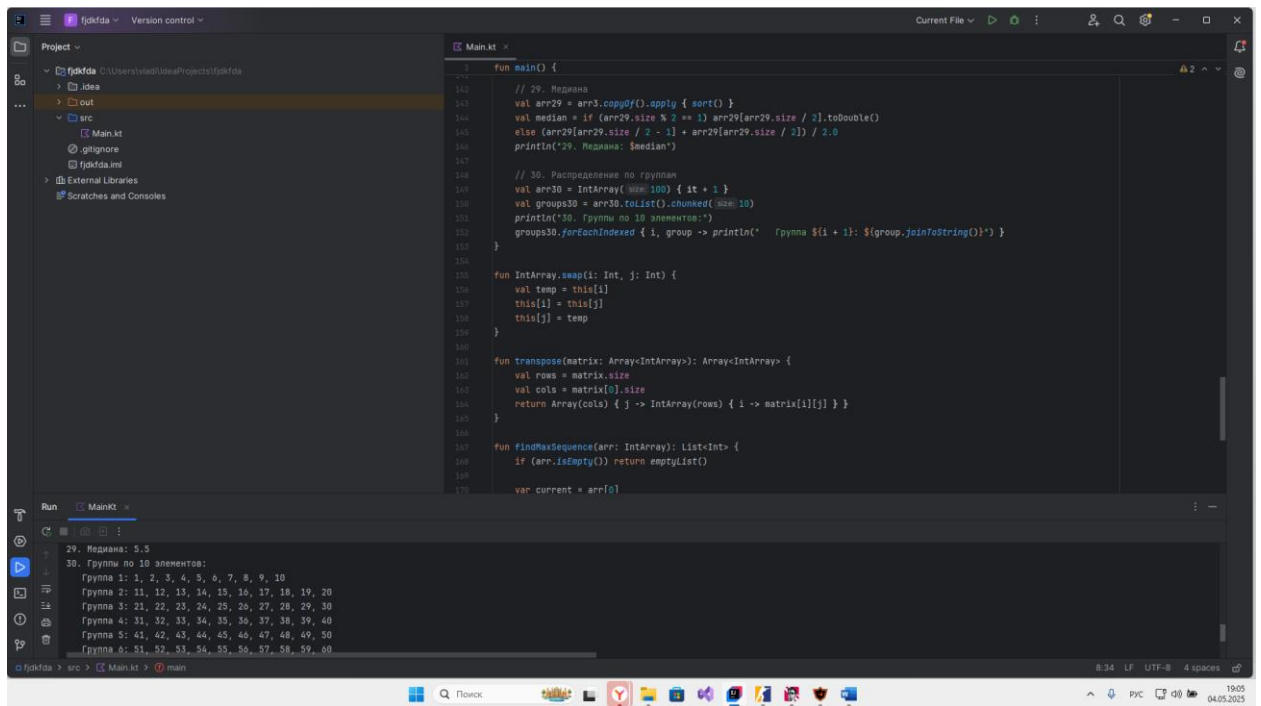
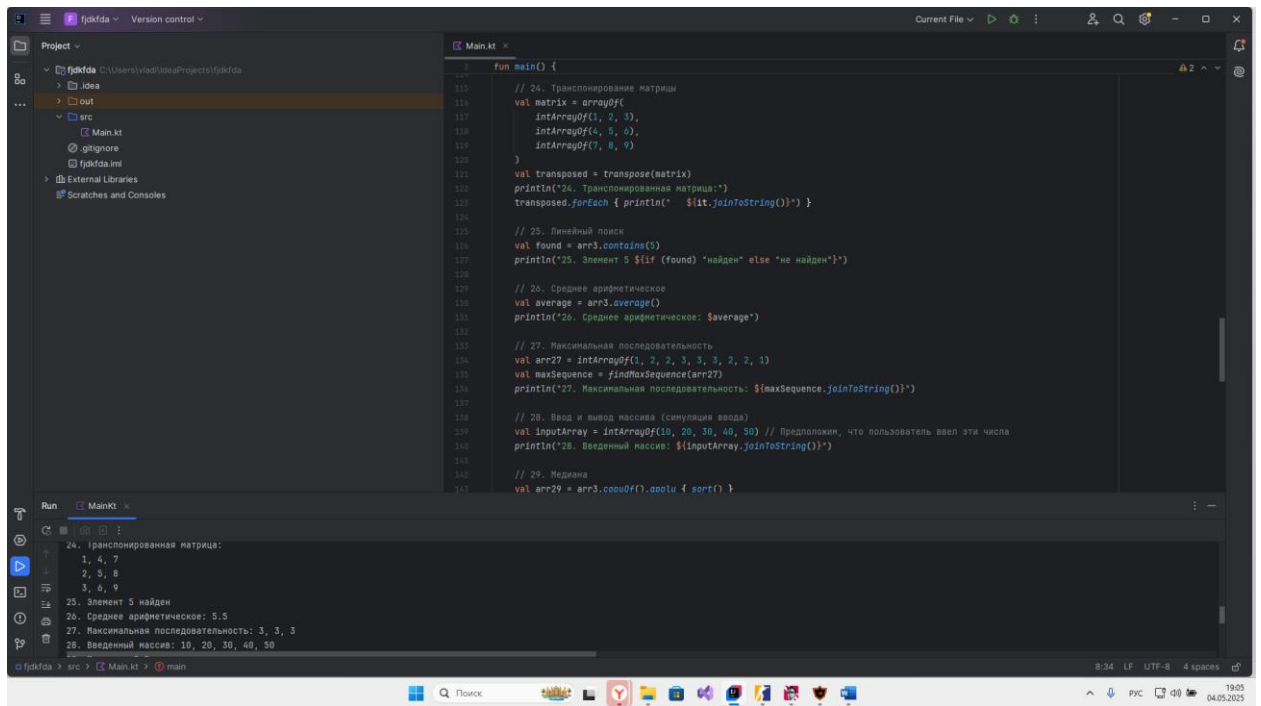
Run console output:

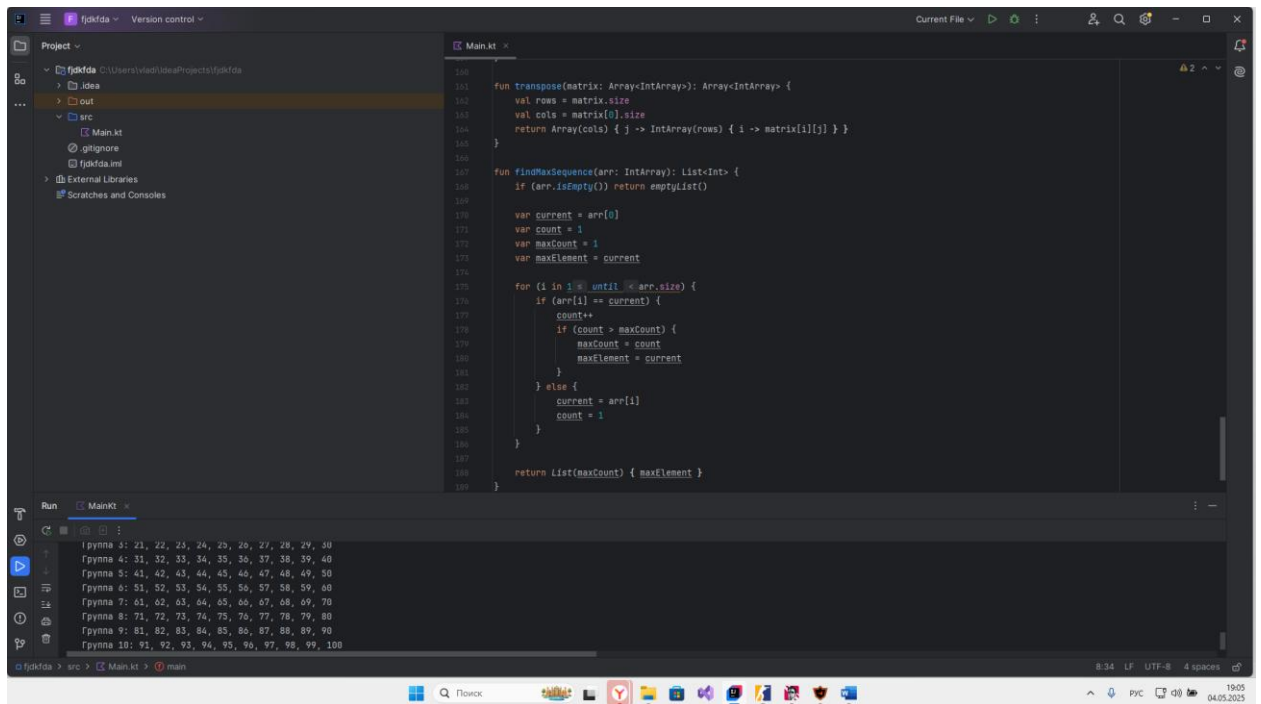
```
1. Массив: 5, 3, 8, 1, 9
2. Сумма элементов: 26
3. Максимальное: 10, Минимальное: 1
4. Отсортированный массив: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
5. Уникальные элементы: 1, 2, 3, 4, 5
6. Четные: 8, 4, 2, 6, 10, Нечетные: 5, 3, 1, 9, 7
7. Реверс массива: 10, 6, 2, 7, 4, 9, 1, 8, 3, 5
```

The screenshot shows the IntelliJ IDEA IDE with the same Kotlin file `Main.kt`. The code continues with reversing the array, finding the index of an element, copying the array, calculating the sum of even numbers, finding the intersection of two arrays, and performing a swap. The Run console shows the output of these operations.

```
7. Реверс массива: 10, 6, 2, 7, 4, 9, 1, 8, 3, 5
8. Индекс элемента 8: 2
9. Скопированный массив: 5, 3, 8, 1, 9, 4, 7, 2, 6, 10
10. Сумма четных чисел: 30
11. Пересечение массивов: 4, 5
12. После перестановки: 3, 5, 8, 1, 9, 4, 7, 2, 6, 10
13. Случайные числа: 95, 18, 56, 39, 1, 26, 33, 62, 52, 59, 28, 65, 1, 51, 67, 56, 68, 32, 58, 55
14. Числа, делящиеся на 3: 3, 9, 6
```







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

```
fun main() {
```

```
    // 1. Создание и вывод элементов
```

```
    val arr1 = intArrayOf(5, 3, 8, 1, 9)
```

```
    println("1. Массив: ${arr1.joinToString()})")
```

```
    // 2. Сумма элементов массива
```

```
    val sum = arr1.sum()
```

```
    println("2. Сумма элементов: $sum")
```

```
    // 3. Максимальное и минимальное значение
```

```
    val arr3 = intArrayOf(5, 3, 8, 1, 9, 4, 7, 2, 6, 10)
```

```
    println("3. Максимальное: ${arr3.maxOrNull()}, Минимальное: ${arr3.minOrNull()})")
```

```
    // 4. Сортировка массива
```

```
    val arr4 = arr3.copyOf()
```

```
    arr4.sort()
```

```
    println("4. Отсортированный массив: ${arr4.joinToString()})")
```

```
    // 5. Уникальные элементы
```

```
    val arr5 = intArrayOf(1, 2, 2, 3, 4, 4, 5)
```

```
    val unique = arr5.distinct()
```

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

```
// 6. Четные и нечетные числа
val (even, odd) = arr3.partition { it % 2 == 0 }
println("6. Четные: ${even.joinToString()}, Нечетные: ${odd.joinToString()}")
```

```
// 7. Реверс массива
val arr7 = arr3.copyOf()
arr7.reverse()
println("7. Реверс массива: ${arr7.joinToString()}")
```

```
// 8. Поиск элемента
val index = arr3.indexOf(8)
println("8. Индекс элемента 8: $index")
```

```
// 9. Копирование массива
val arr9 = arr3.copyOf()
println("9. Скопированный массив: ${arr9.joinToString()}")
```

```
// 10. Сумма четных чисел
val sumEven = arr3.filter { it % 2 == 0 }.sum()
println("10. Сумма четных чисел: $sumEven")
```

```
// 11. Пересечение массивов
val arr11a = intArrayOf(1, 2, 3, 4, 5)
val arr11b = intArrayOf(4, 5, 6, 7, 8)
val intersect = arr11a.intersect(arr11b.toList())
println("11. Пересечение массивов: ${intersect.joinToString()}")
```

```
// 12. Перестановка элементов
val arr12 = arr3.copyOf()
arr12.swap(0, 1)
println("12. После перестановки: ${arr12.joinToString()}")
```

```
// 13. Заполнение случайными числами
val arr13 = IntArray(20) { Random.nextInt(1, 101) }
println("13. Случайные числа: ${arr13.joinToString()}")
```

```
// 14. Числа Прокопенко (делящиеся на 3)
val divisibleBy3 = arr3.filter { it % 3 == 0 }
println("14. Числа, делящиеся на 3: ${divisibleBy3.joinToString()}")
```

```
// 15. Проверка на палиндром
val arr15 = intArrayOf(1, 2, 3, 2, 1)
val isPalindrome = arr15.contentEquals(arr15.reversedArray())
```

```
println("15. Массив ${if (isPalindrome) "является" else "не является"} палиндромом")
```

```
// 16. Конкатенация двух массивов
```

```
val arr16a = intArrayOf(1, 2, 3)
```

```
val arr16b = intArrayOf(4, 5, 6)
```

```
val concatenated = arr16a + arr16b
```

```
println("16. Конкатенация: ${concatenated.joinToString()}")
```

```
// 17. Сумма и произведение
```

```
val sum17 = arr3.sum()
```

```
val product17 = arr3.fold(1) { acc, i -> acc * i }
```

```
println("17. Сумма: $sum17, Произведение: $product17")
```

```
// 18. Группировка чисел
```

```
val arr18 = IntArray(25) { it + 1 }
```

```
val groups = arr18.toList().chunked(5)
```

```
println("18. Группы по 5 элементов:")
```

```
groups.forEachIndexed { i, group -> println(" Группа ${i + 1}: ${group.joinToString()}") }
```

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

```
val arr19a = intArrayOf(1, 3, 5, 7)
```

```
val arr19b = intArrayOf(2, 4, 6, 8)
```

```
val merged = (arr19a + arr19b).sortedArray()
```

```
println("19. Слияние массивов: ${merged.joinToString()}")
```

```
// 20. Арифметическая прогрессия
```

```
val arr20 = IntArray(10) { 5 + it * 3 }
```

```
println("20. Арифметическая прогрессия: ${arr20.joinToString()}")
```

```
// 21. Удаление элемента
```

```
val arr21 = arr3.copyOf().toMutableList()
```

```
arr21.removeAt(2)
```

```
println("21. После удаления элемента: ${arr21.joinToString()}")
```

```
// 22. Второй максимальный элемент
```

```
val sorted = arr3.sortedDescending()
```

```
val secondMax = sorted[1]
```

```
println("22. Второй максимальный элемент: $secondMax")
```

```
// 23. Объединение массивов
```

```
val arr23a = intArrayOf(1, 2, 3)
```

```
val arr23b = intArrayOf(4, 5)
```

```
val arr23c = intArrayOf(6, 7, 8, 9)
```

```

val combined = arr23a + arr23b + arr23c
println("23. Объединение массивов: ${combined.joinToString()}")

// 24. Транспонирование матрицы
val matrix = arrayOf(
    intArrayOf(1, 2, 3),
    intArrayOf(4, 5, 6),
    intArrayOf(7, 8, 9)
)
val transposed = transpose(matrix)
println("24. Транспонированная матрица:")
transposed.forEach { println("  ${it.joinToString()}") }

// 25. Линейный поиск
val found = arr3.contains(5)
println("25. Элемент 5 ${if (found) "найден" else "не найден"}")

// 26. Среднее арифметическое
val average = arr3.average()
println("26. Среднее арифметическое: $average")

// 27. Максимальная последовательность
val arr27 = intArrayOf(1, 2, 2, 3, 3, 3, 2, 2, 1)
val maxSequence = findMaxSequence(arr27)
println("27. Максимальная последовательность: ${maxSequence.joinToString()}")

// 28. Ввод и вывод массива (симуляция ввода)
val inputArray = intArrayOf(10, 20, 30, 40, 50) // Предположим, что пользователь ввел эти
числа
println("28. Введенный массив: ${inputArray.joinToString()}")

// 29. Медиана
val arr29 = arr3.copyOf().apply { sort() }
val median = if (arr29.size % 2 == 1) arr29[arr29.size / 2].toDouble()
else (arr29[arr29.size / 2 - 1] + arr29[arr29.size / 2]) / 2.0
println("29. Медиана: $median")

// 30. Распределение по группам
val arr30 = IntArray(100) { it + 1 }
val groups30 = arr30.toList().chunked(10)
println("30. Группы по 10 элементов:")
groups30.forEachIndexed { i, group -> println("  Группа ${i + 1}: ${group.joinToString()}") }
}

```

```

fun IntArray.swap(i: Int, j: Int) {
    val temp = this[i]
    this[i] = this[j]
    this[j] = temp
}

fun transpose(matrix: Array<IntArray>): Array<IntArray> {
    val rows = matrix.size
    val cols = matrix[0].size
    return Array(cols) { j -> IntArray(rows) { i -> matrix[i][j] } }
}

fun findMaxSequence(arr: IntArray): List<Int> {
    if (arr.isEmpty()) return emptyList()

    var current = arr[0]
    var count = 1
    var maxCount = 1
    var maxElement = current

    for (i in 1 until arr.size) {
        if (arr[i] == current) {
            count++
            if (count > maxCount) {
                maxCount = count
                maxElement = current
            }
        } else {
            current = arr[i]
            count = 1
        }
    }

    return List(maxCount) { maxElement }
}

```