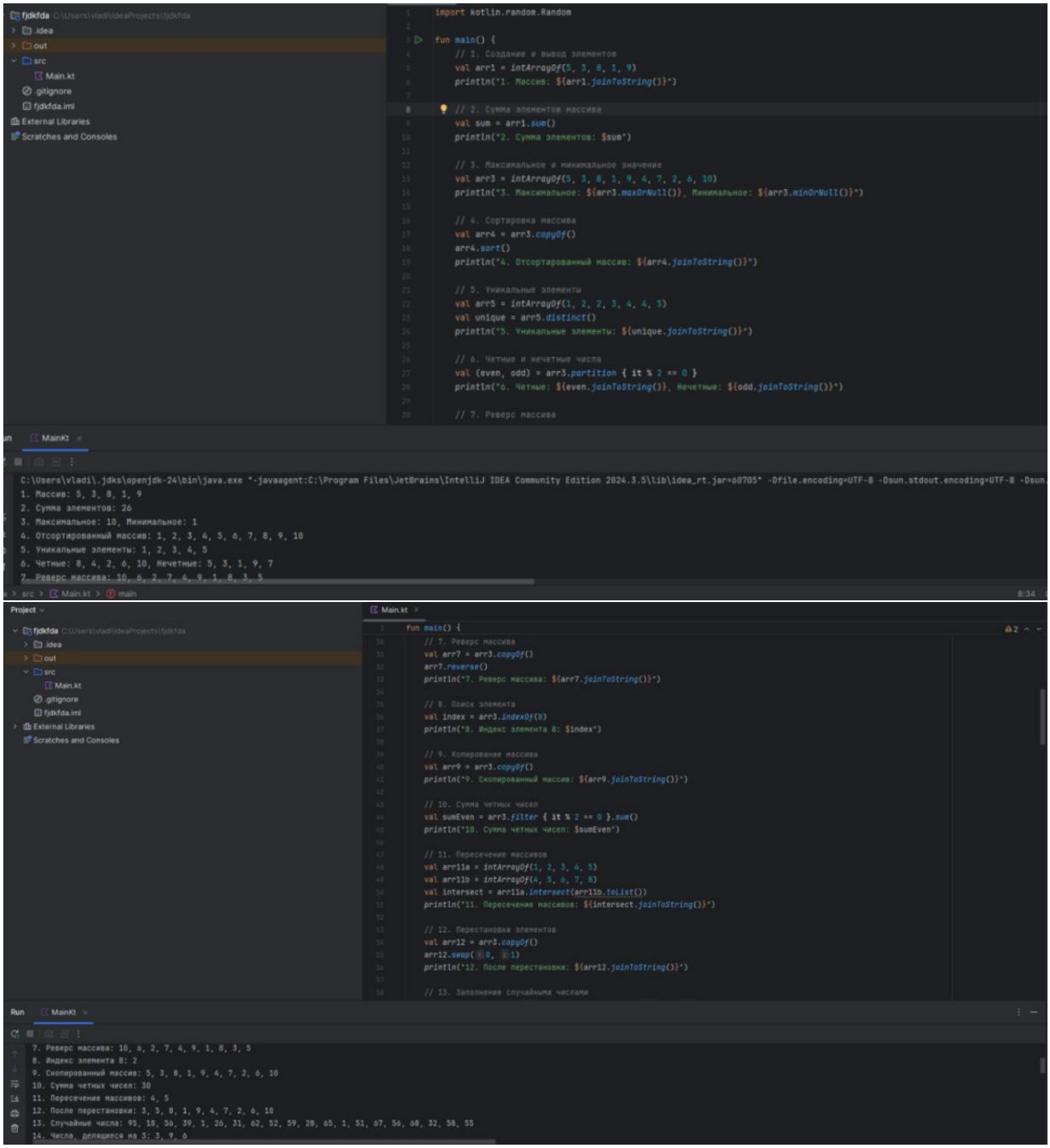
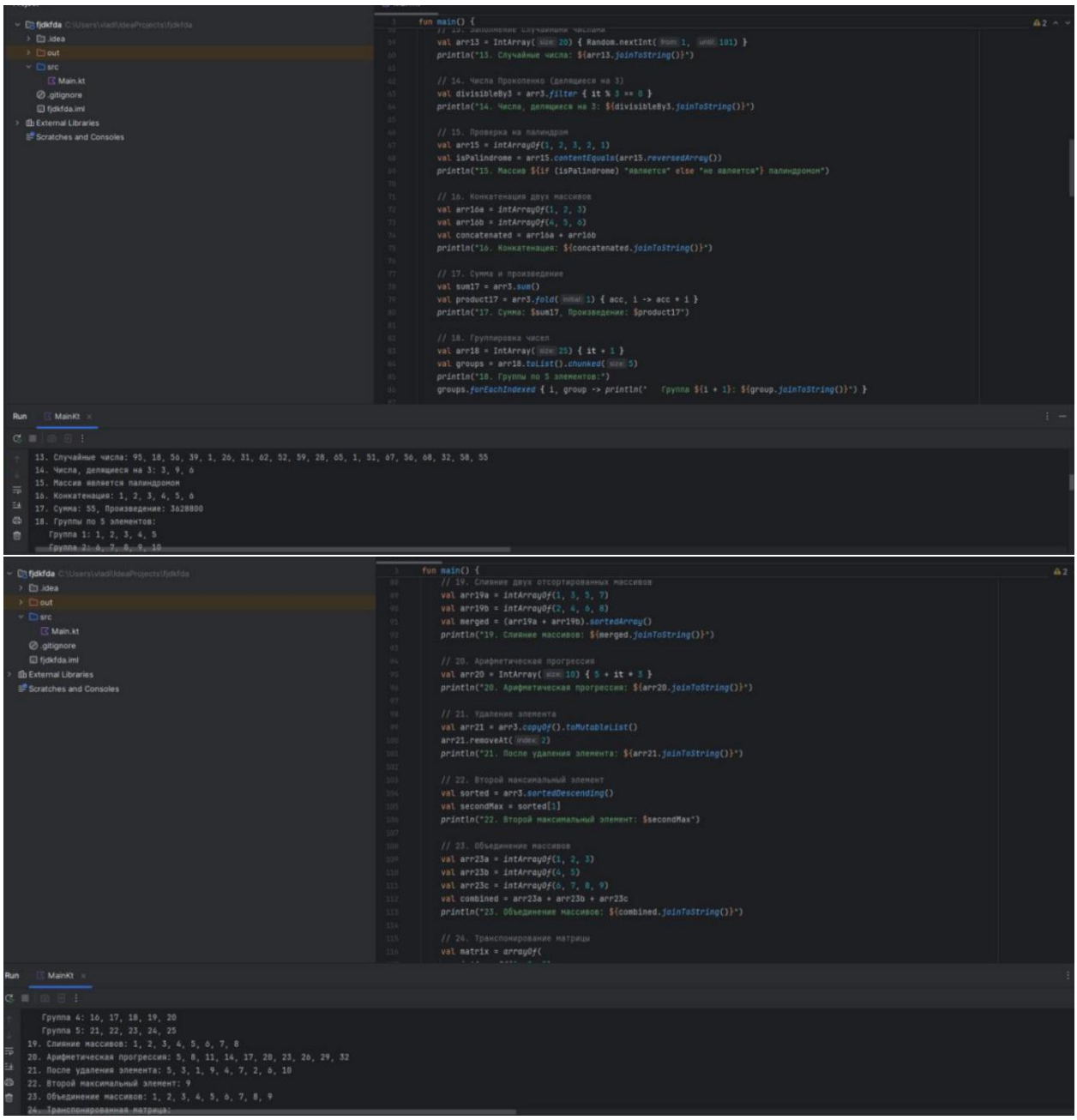
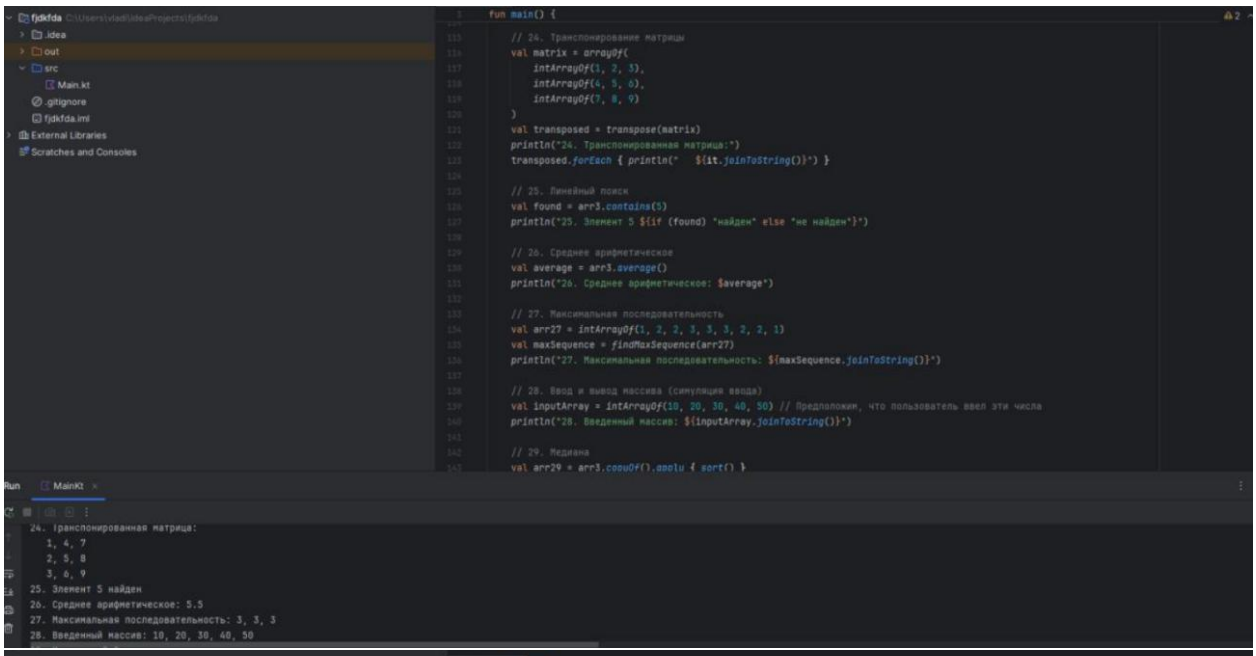
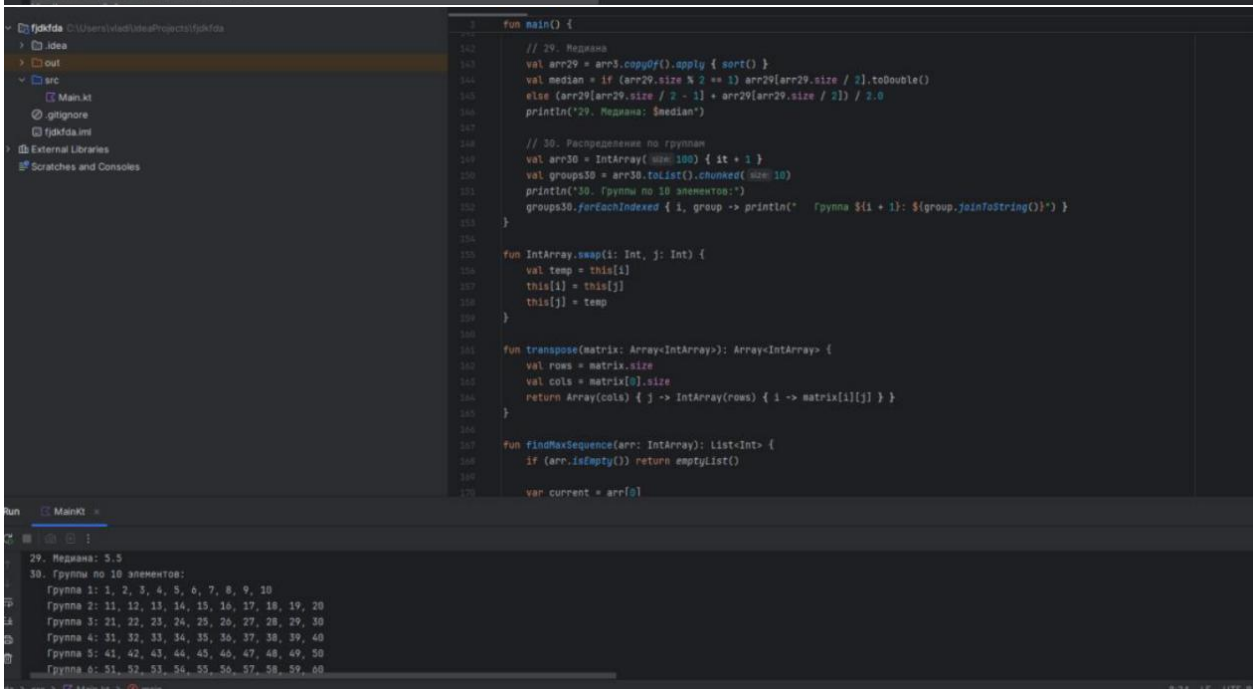
Лабораторная работа №6
Харахардин А. ИС233




fjdkfda C:\Users\vlad\IdeaProjects\fjdkfda
.idea
out
src
Main.kt
.gitignore
fjdkfda.iml
External Libraries
Scratches and Consoles
fun main() {
// 13. Заполнение случайными числами
val arr13 = IntArray(size: 20) { Random.nextInt(from: 1, until: 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(initial: 1) { acc, i -> acc * i }
println("17. Сумма: $sum17, Произведение: $product17")
// 18. Группировка чисел
val arr18 = IntArray(size: 25) { it + 1 }
val groups = arr18.toList().chunked(size: 5)
println("18. Группы по 5 элементов:")
groups.forEachIndexed { i, group -> println(" Группа ${i + 1}: ${group.joinToString()}") }
Run
MainKt
13. Случайные числа: 95, 18, 56, 39, 1, 26, 31, 62, 52, 59, 28, 65, 1, 51, 67, 56, 68, 32, 58, 55
14. Числа, делящиеся на 3: 3, 9, 6
15. Массив является палиндромом
16. Конкатенация: 1, 2, 3, 4, 5, 6
17. Сумма: 55, Произведение: 3628800
18. Группы по 5 элементов:
Группа 1: 1, 2, 3, 4, 5
Группа 2: 6, 7, 8, 9, 10

fjdkfda C:\Users\vlad\IdeaProjects\fjdkfda
.idea
out
src
Main.kt
.gitignore
fjdkfda.iml
External Libraries
Scratches and Consoles
fun main() {
// 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(size: 10) { 5 + it * 3 }
println("20. Арифметическая прогрессия: ${arr20.joinToString()}")
// 21. Удаление элемента
val arr21 = arr3.copyOf().toMutableList()
arr21.removeAt(index: 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(
Run
MainKt
Группа 4: 16, 17, 18, 19, 20
Группа 5: 21, 22, 23, 24, 25
19. Слияние массивов: 1, 2, 3, 4, 5, 6, 7, 8
20. Арифметическая прогрессия: 5, 8, 11, 14, 17, 20, 23, 26, 29, 32
21. После удаления элемента: 5, 3, 1, 9, 4, 7, 2, 6, 10
22. Второй максимальный элемент: 9
23. Объединение массивов: 1, 2, 3, 4, 5, 6, 7, 8, 9
24. Транспонированная матрица:


fjdkfda C:\Users\vlad\IdeaProjects\fjdkfda
.idea
out
src
Main.kt
.gitignore
fjdkfda.iml
External Libraries
Scratches and Consoles
fun main() {
// 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() }
Run
MainKt
24. Транспонированная матрица:
1, 4, 7
2, 5, 8
3, 6, 9
25. Элемент 5 найден
26. Среднее арифметическое: 5.5
27. Максимальная последовательность: 3, 3, 3
28. Введенный массив: 10, 20, 30, 40, 50


fjdkfda C:\Users\vlad\IdeaProjects\fjdkfda
.idea
out
src
Main.kt
.gitignore
fjdkfda.iml
External Libraries
Scratches and Consoles
fun main() {
// 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(size: 100) { it + 1 }
val groups30 = arr30.toList().chunked(size: 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]
Run
MainKt
29. Медиана: 5.5
30. Группы 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