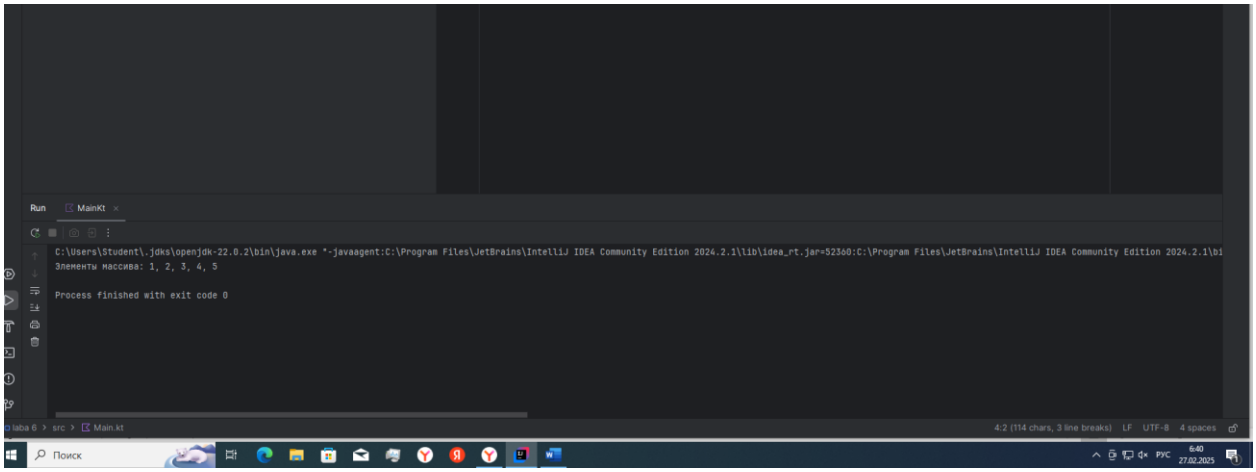
Практическая работа №6

Выполнила Вожегова К.А.

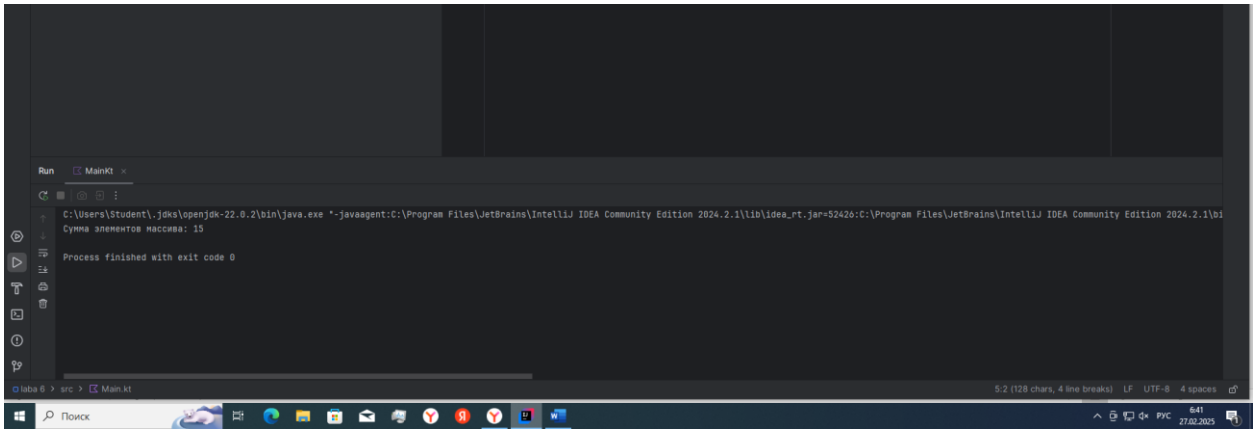
1. Создание и вывод элементов: Создайте массив из 5 целых чисел и выведите их на экран.

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



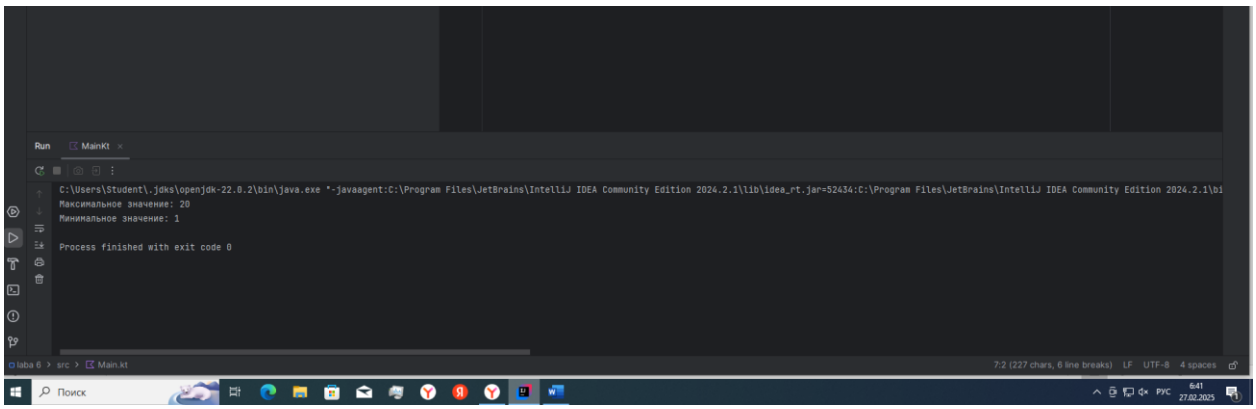
2. Сумма элементов массива: Напишите программу, которая находит сумму всех элементов массива чисел.

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



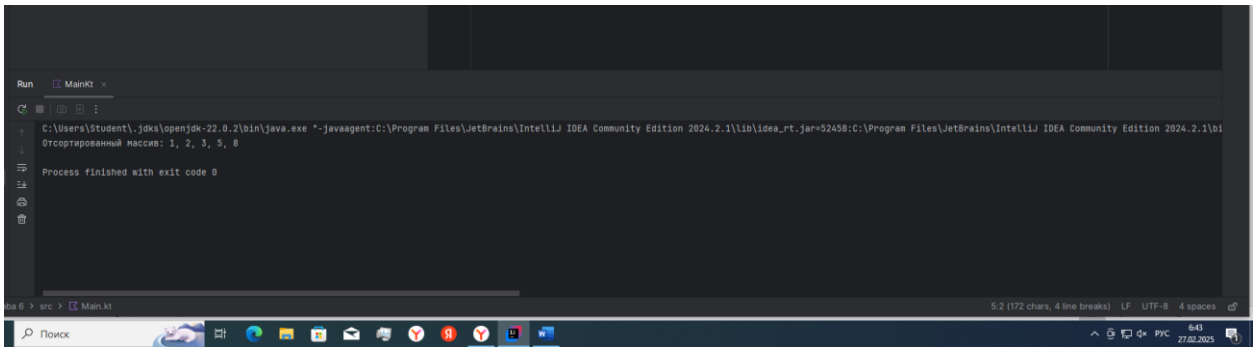
3. Максимальное и минимальное значение: Создайте массив из 10 чисел, найдите и выведите максимальное и минимальное значение.

```
fun main() {
    val numbers = arrayOf(10, 5, 8, 3, 15, 7, 20, 1, 6, 12)
    val max = numbers.maxOrNull()
    val min = numbers.minOrNull()
    println("Максимальное значение: $max")
    println("Минимальное значение: $min")
}
```



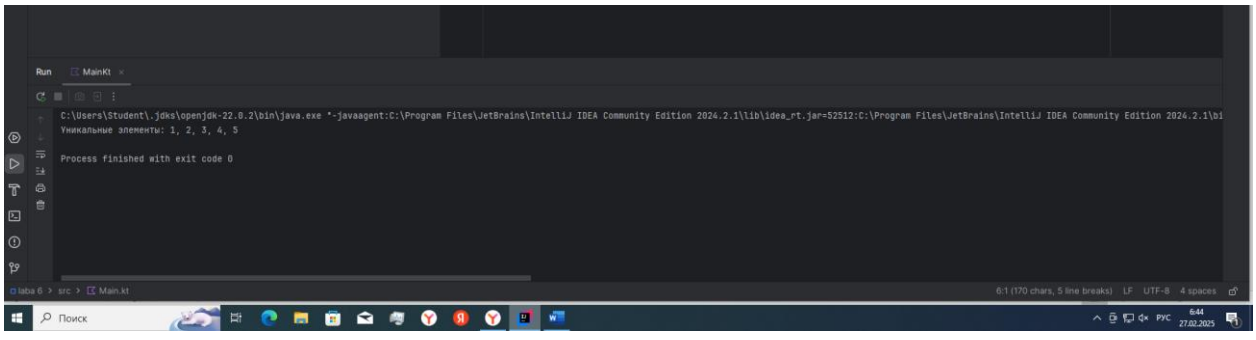
4. Сортировка массива: Реализуйте алгоритм сортировки для массива чисел и выведите отсортированный массив.

```
fun main() {
    val numbers = arrayOf(5, 3, 8, 1, 2)
    val sortedNumbers = numbers.sortedArray()
    println("Отсортированный массив: ${sortedNumbers.joinToString()}")
}
```



5. Уникальные элементы: Напишите программу, которая выводит уникальные элементы из массива.

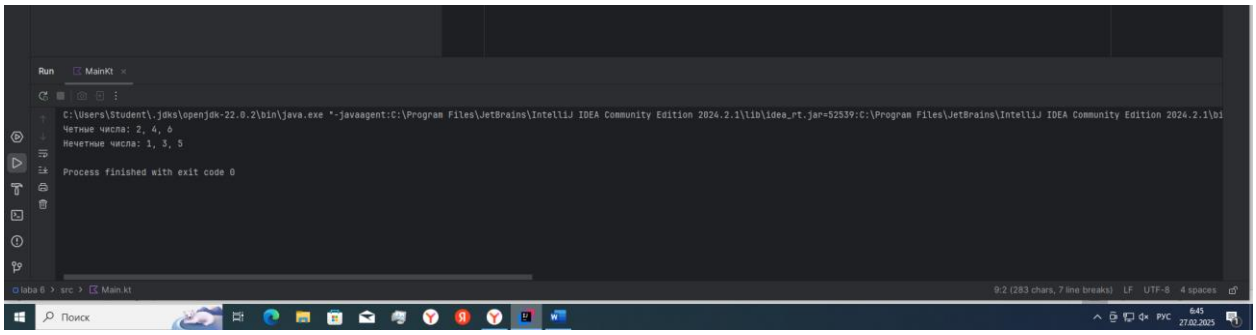
```
fun main() {
    val numbers = arrayOf(1, 2, 2, 3, 4, 4, 5)
    val uniqueNumbers = numbers.toSet()
    println("Уникальные элементы: ${uniqueNumbers.joinToString()}")
}
```



6. Четные и нечетные числа: Создайте массив и разделите его на четные и нечетные числа, сохранив их в разные массивы.

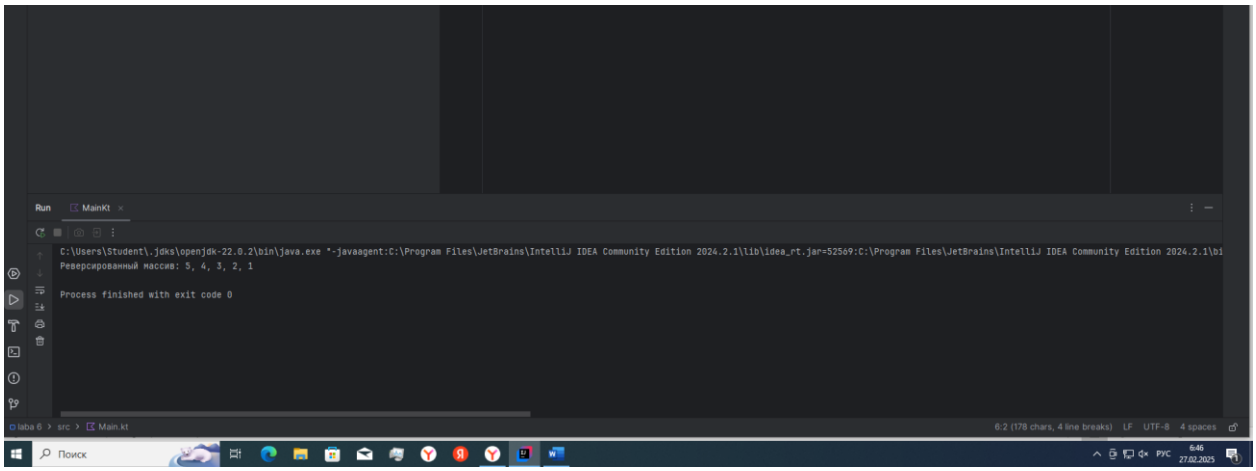
```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5, 6)
    val evenNumbers = numbers.filter { it % 2 == 0 }
    val oddNumbers = numbers.filter { it % 2 != 0 }
```

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



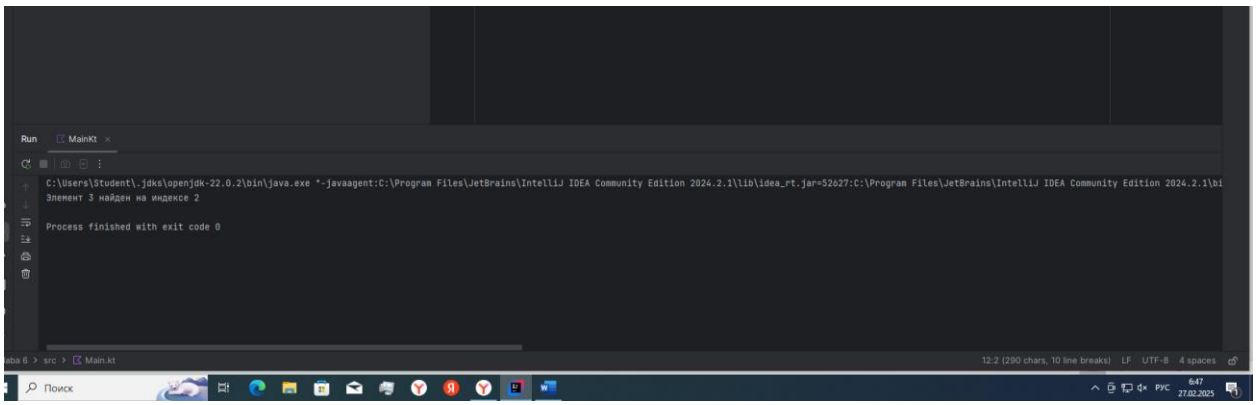
7. Реверс массива: Напишите программу, которая реверсирует массив чисел.

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5)
    val reversedNumbers = numbers.reversedArray()
    println("Реверсированный массив: ${reversedNumbers.joinToString()}")
}
```



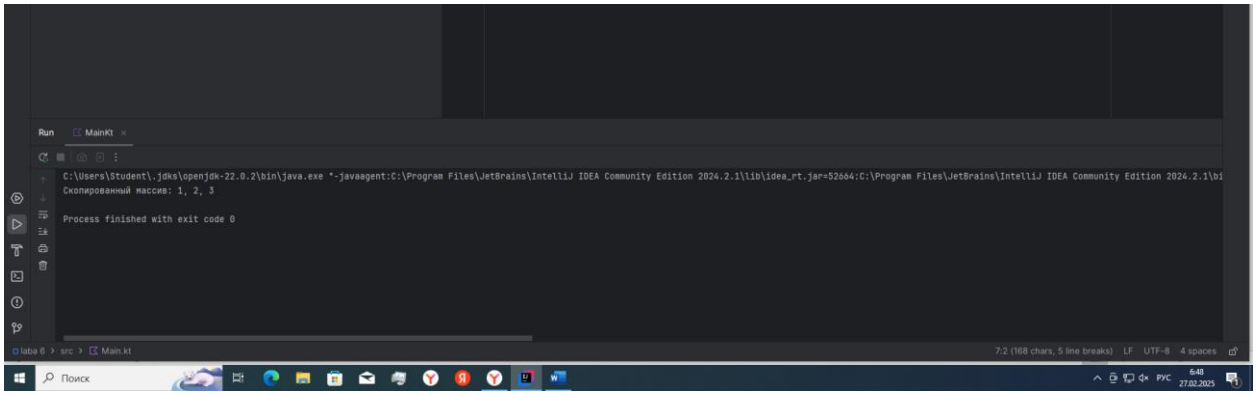
8. Поиск элемента: Реализуйте поиск элемента в массиве и выводите его индекс.

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5)
    val elementToFind = 3
    val index = numbers.indexOf(elementToFind)

    if (index != -1) {
        println("Элемент $elementToFind найден на индексе $index")
    } else {
        println("Элемент $elementToFind не найден")
    }
}
```



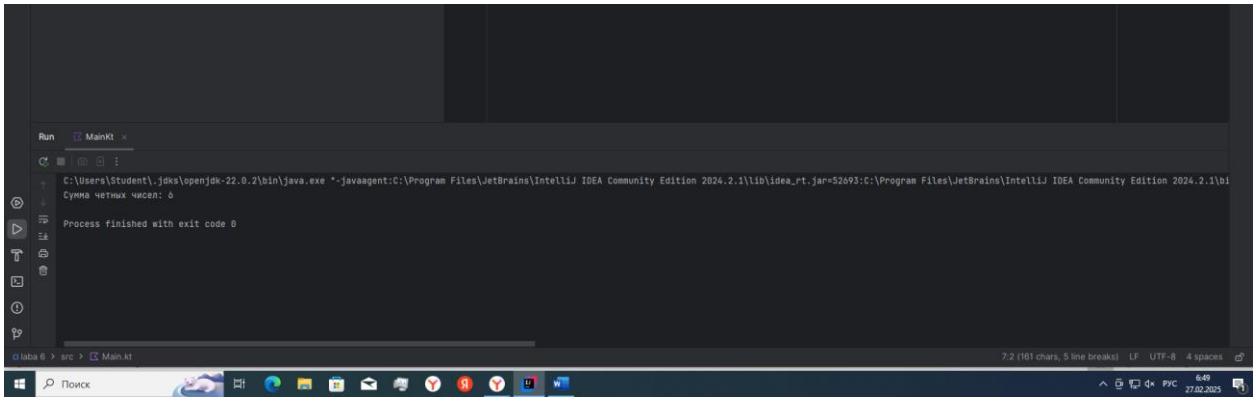
9. Копирование массива: Создайте новый массив, скопировав в него элементы из другого массива.

```
fun main() {
    val originalArray = arrayOf(1, 2, 3)
    val copiedArray = originalArray.copyOf()

    println("Скопированный массив: ${copiedArray.joinToString()}")
}
```



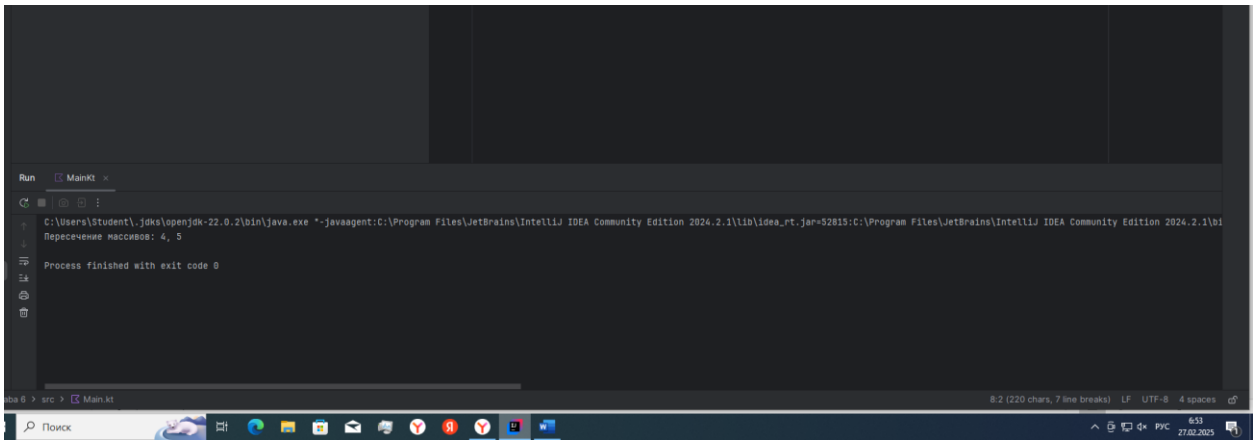
10.Сумма четных чисел: Напишите программу, которая находит сумму всех четных чисел в массиве.

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5)
    val sumOfEvens = numbers.filter { it % 2 == 0 }.sum()

    println("Сумма четных чисел: $sumOfEvens")
}
```



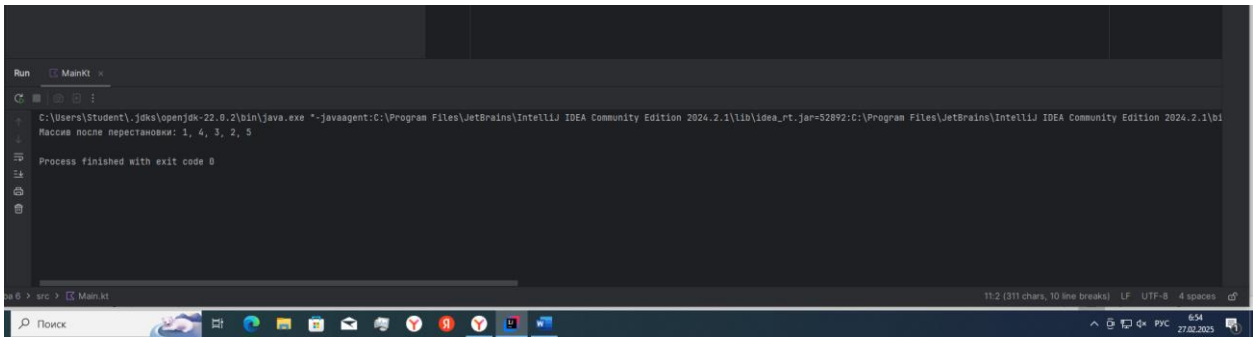
11.Пересечение массивов: Напишите программу, которая находит пересечение двух массивов и выводит результат.

```
fun main() {
    val array1 = arrayOf(1, 2, 3, 4, 5)
    val array2 = arrayOf(4, 5, 6, 7, 8)
    val intersection = array1.intersect(array2.toSet())

    println("Пересечение массивов: ${intersection.joinToString()}")
}
```



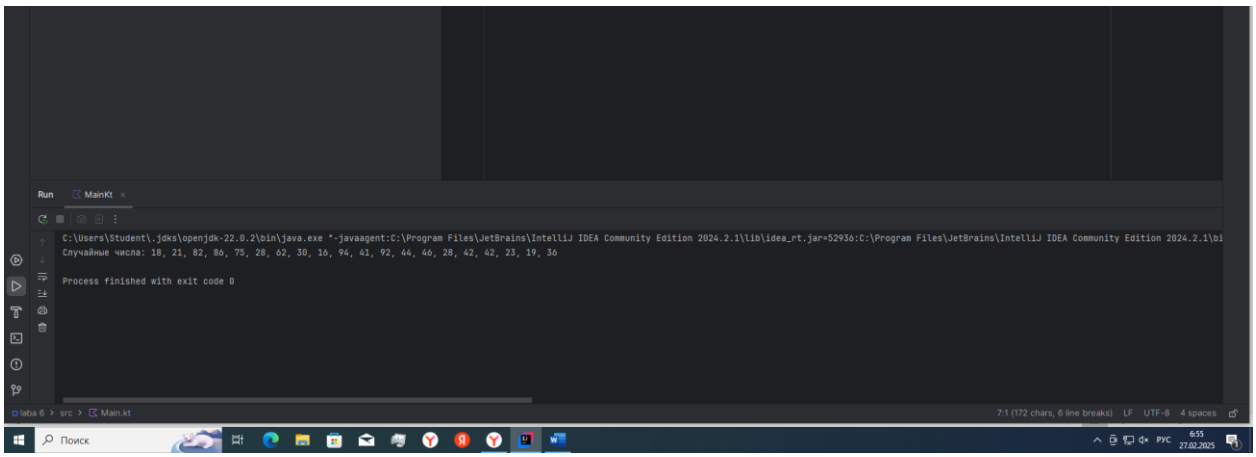
12.Перестановка элементов: Реализуйте функцию, которая меняет местами два элемента в массиве.

```
fun swapElements(array: IntArray, index1: Int, index2: Int) {
    val temp = array[index1]
    array[index1] = array[index2]
    array[index2] = temp
}

fun main() {
    val numbers = intArrayOf(1, 2, 3, 4, 5)
    swapElements(numbers, 1, 3)
    println("Массив после перестановки: ${numbers.joinToString()}")
}
```



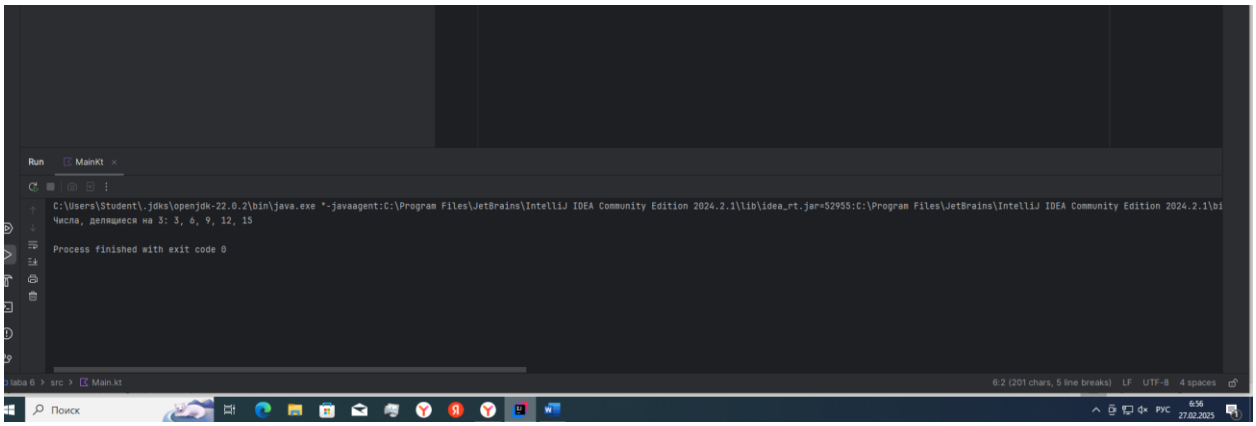
13.Заполнение случайными числами: Создайте массив из 20 случайных чисел от 1 до 100 и выведите его на экран.

```
import kotlin.random.Random

fun main() {
    val randomNumbers = IntArray(20) { Random.nextInt(1, 101) }
    println("Случайные числа: ${randomNumbers.joinToString()}")
}
```



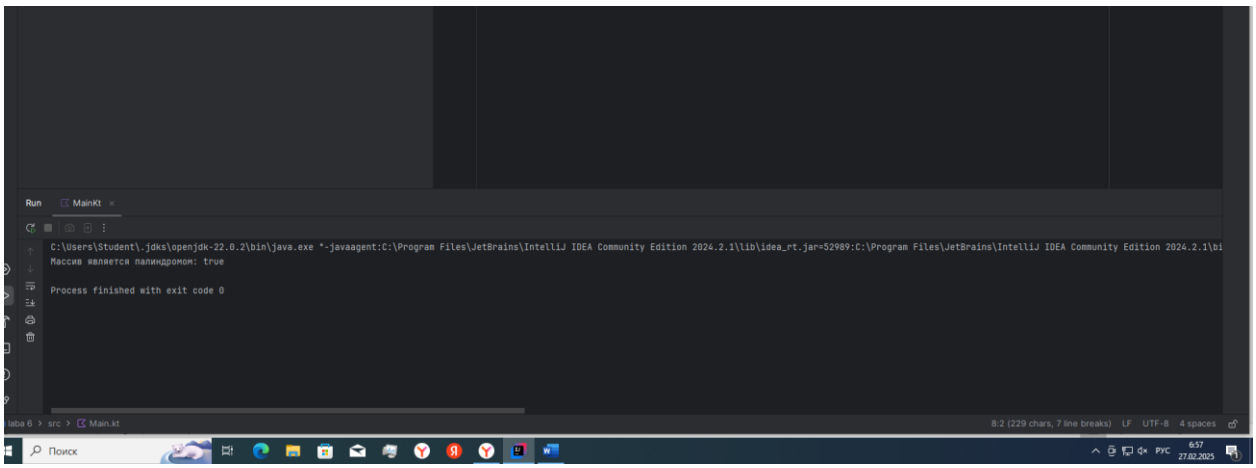
14.Числа Прокопенко: Напишите программу, которая выводит все числа в массиве, делящиеся на 3.

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5, 6, 9, 12, 15)
    val divisibleByThree = numbers.filter { it % 3 == 0 }

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



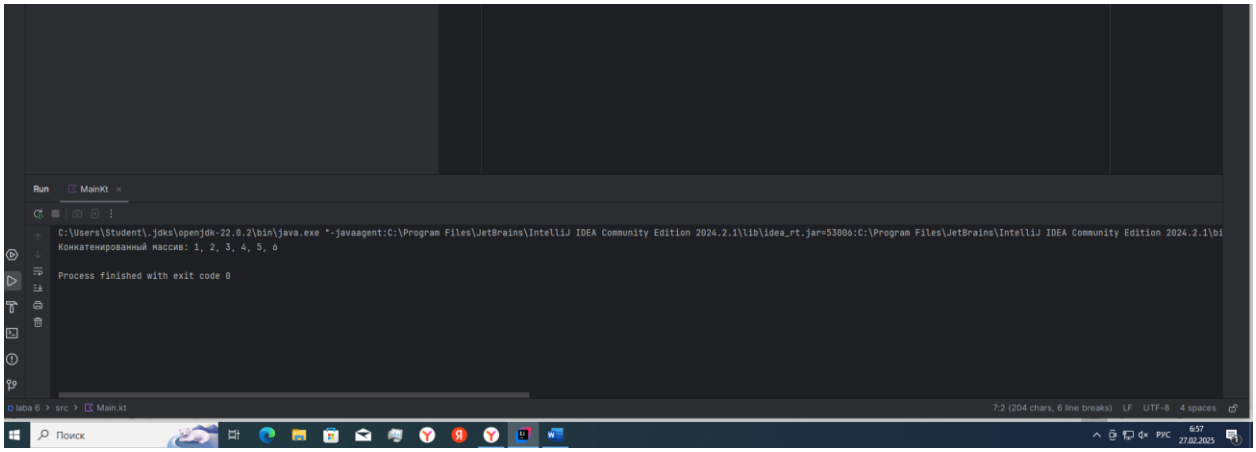
15.Проверка на палиндром: Напишите программу, которая проверяет, является ли массив палиндромом.

```
fun isPalindrome(array: IntArray): Boolean {
    return array.contentEquals(array.reversedArray())
}

fun main() {
    val numbers = intArrayOf(1, 2, 3, 2, 1)
    println("Массив является палиндромом: ${isPalindrome(numbers)}")
}
```



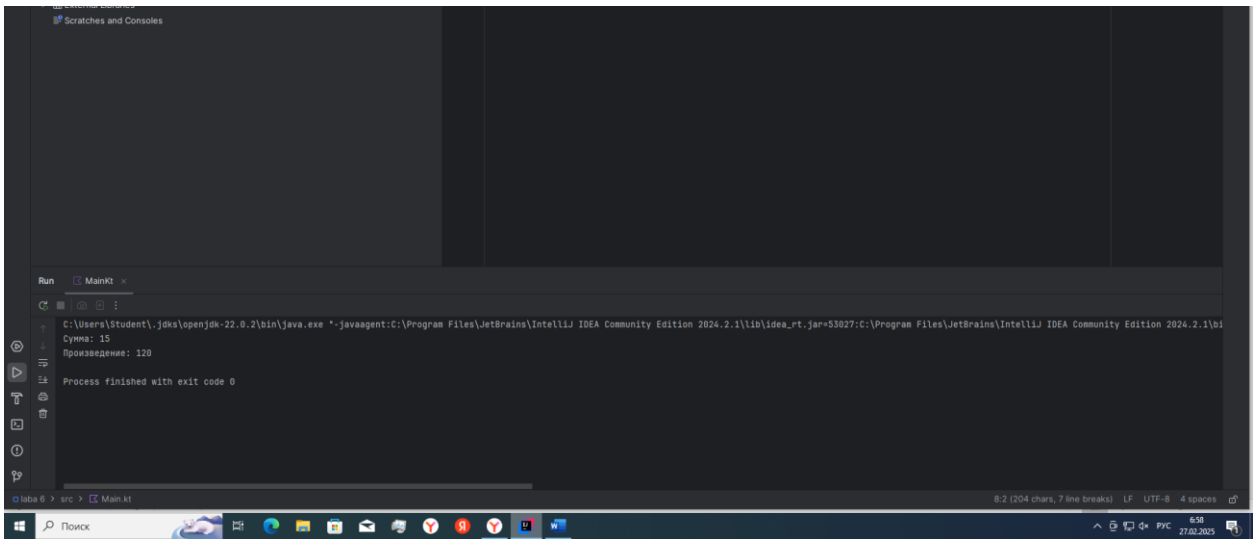
16.Конкатенация двух массивов: Создайте два массива и соедините их в один.

```
fun main() {
    val array1 = arrayOf(1, 2, 3)
    val array2 = arrayOf(4, 5, 6)
    val concatenatedArray = array1 + array2

    println("Конкатенированный массив: ${concatenatedArray.joinToString()}")
}
```



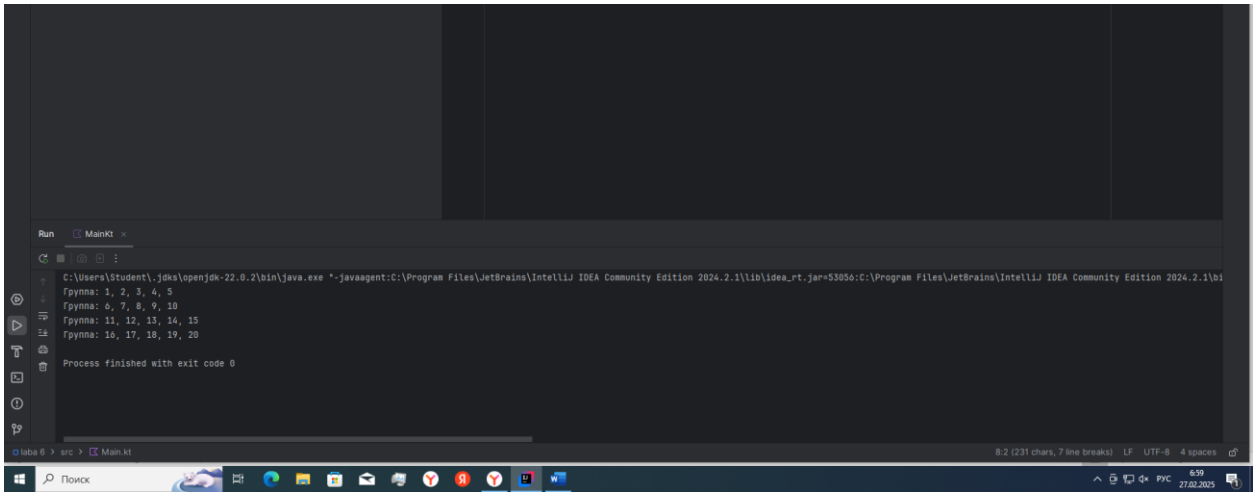
17.Сумма и произведение: Напишите программу, которая находит и выводит сумму и произведение всех элементов массива.

```
fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5)
    val sum = numbers.sum()
    val product = numbers.reduce { acc, i -> acc * i }

    println("Сумма: $sum")
    println("Произведение: $product")
}
```



18.Группировка чисел: Разделите массив на группы по 5 элементов и выведите их.

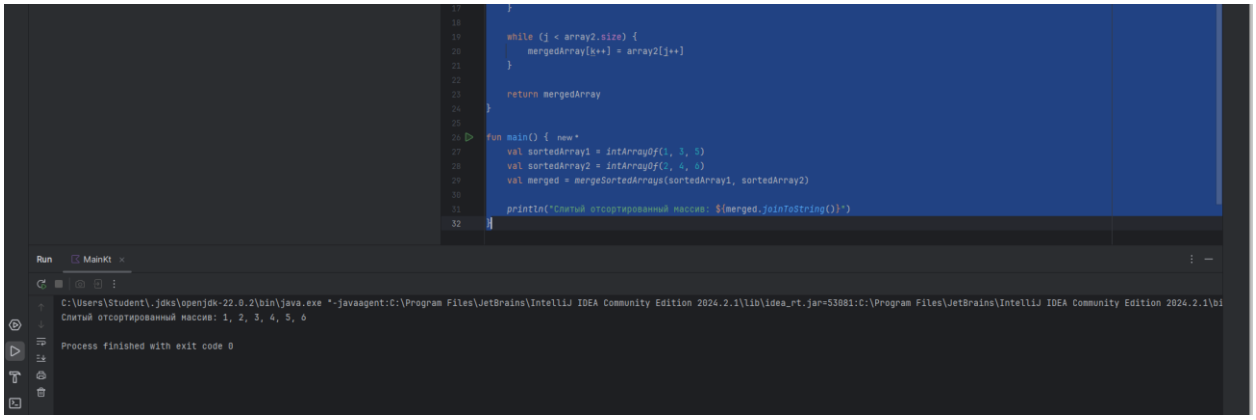
```
fun main() {
    val numbers = (1..20).toList().toIntArray()

    for (i in numbers.indices step 5) {
        val group = numbers.copyOfRange(i, minOf(i + 5, numbers.size))
        println("Группа: ${group.joinToString()}")
    }
}
```



19.Слияние двух массивов: Напишите программу, которая сливает два отсортированных массива в один отсортированный массив.

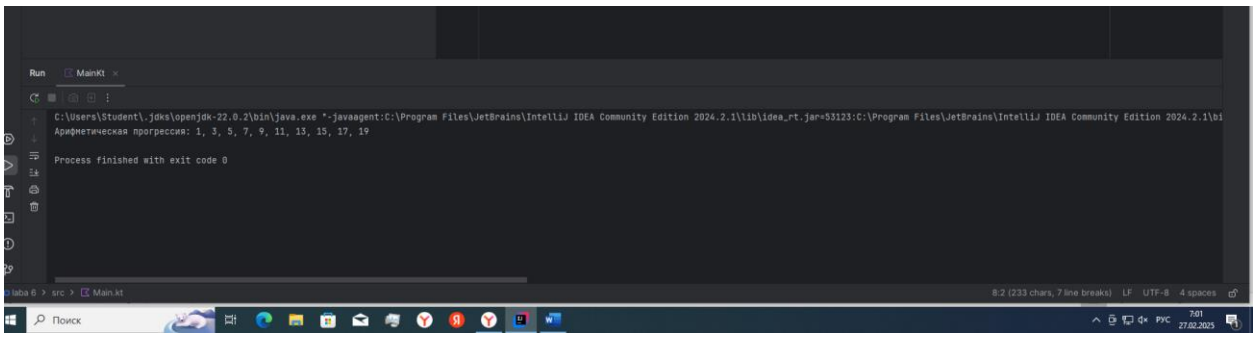
```
fun mergeSortedArrays(array1: IntArray, array2: IntArray): IntArray {
    val mergedArray = IntArray(array1.size + array2.size)
    var i = 0
    var j = 0
    var k = 0

    while (i < array1.size && j < array2.size) {
        if (array1[i] < array2[j]) {
            mergedArray[k++] = array1[i++]
        } else {
            mergedArray[k++] = array2[j++]
        }
```

```
    }

    while (i < array1.size) {
        mergedArray[k++] = array1[i++]
    }

    while (j < array2.size) {
        mergedArray[k++] = array2[j++]
    }

    return mergedArray
}

fun main() {
    val sortedArray1 = intArrayOf(1, 3, 5)
    val sortedArray2 = intArrayOf(2, 4, 6)
    val merged = mergeSortedArrays(sortedArray1, sortedArray2)

    println("Слитый отсортированный массив: ${merged.joinToString()}")
}
```



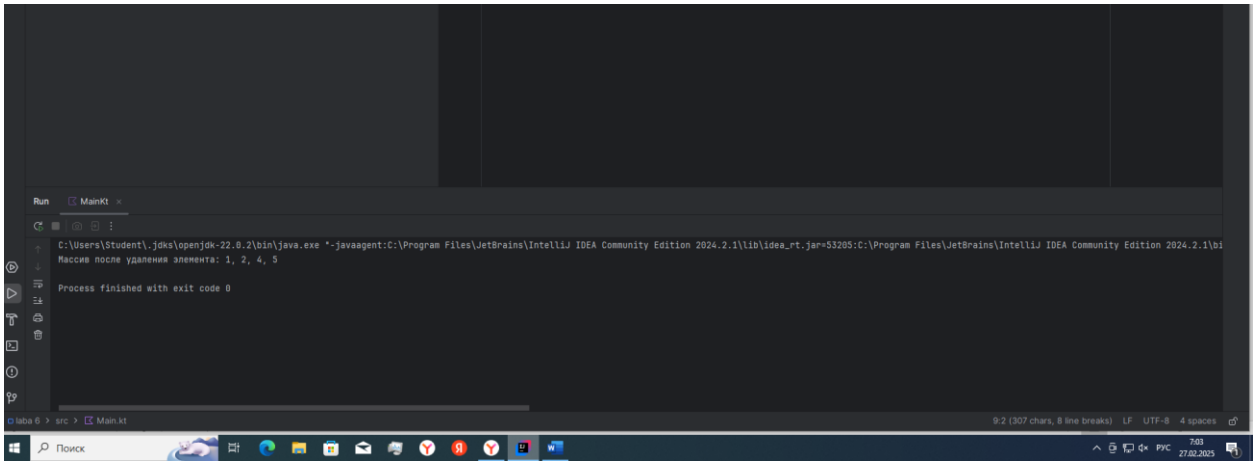
20.Числовая последовательность: Создайте массив целых чисел, представляющий арефметическую прогрессию, и выведите его.

```
fun main() {
    val start = 1
    val difference = 2
    val count = 10
    val arithmeticProgression = IntArray(count) { start + it * difference }

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



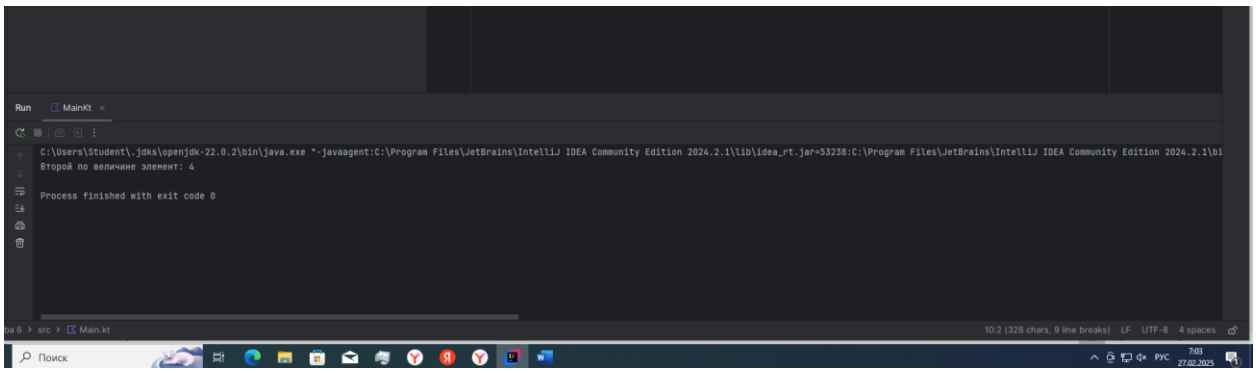
21.Удаление элемента: Реализуйте функцию, которая удаляет заданный элемент из массива.

```
fun removeElement(array: IntArray, element: Int): IntArray {
    return array.filter { it != element }.toIntArray()
}

fun main() {
    val numbers = intArrayOf(1, 2, 3, 4, 5, 3)
    val updatedArray = removeElement(numbers, 3)
    println("Массив после удаления элемента: ${updatedArray.joinToString()}")
}
```



22.Поиск второго максимального: Напишите программу, которая находит второй по величине элемент в массиве.

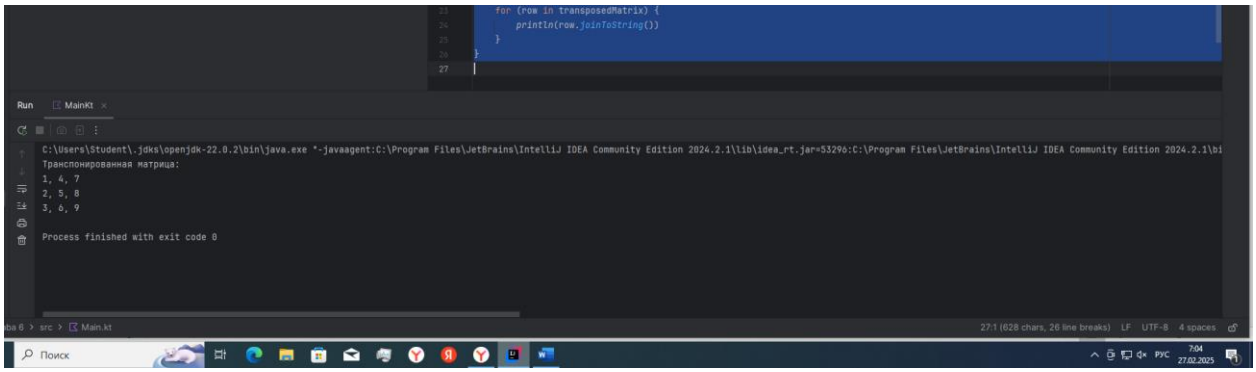
```
fun findSecondMax(array: IntArray): Int? {
    val uniqueNumbers = array.distinct().sortedDescending()
    return if (uniqueNumbers.size >= 2) uniqueNumbers[1] else null
}

fun main() {
    val numbers = intArrayOf(1, 2, 3, 4, 5)
    val secondMax = findSecondMax(numbers)
    println("Второй по величине элемент: $secondMax")
}
```



23.Объединение массивов: Напишите функцию, которая объединяет несколько массивов чисел и выводит результирующий массив.

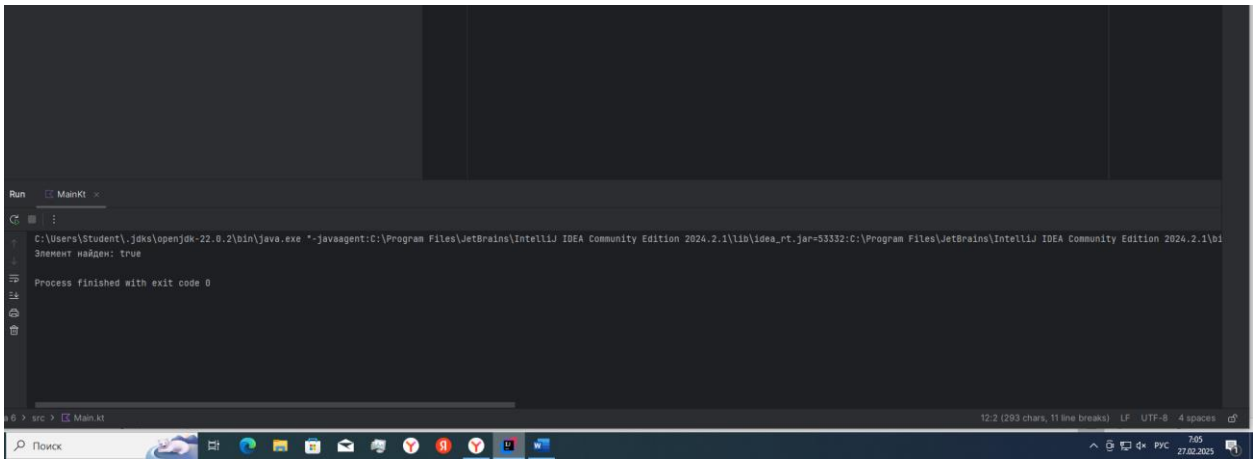
24.Транспонирование матрицы: Создайте матрицу (двумерный массив) и напишите программу, которая транспонирует её.

```
fun transposeMatrix(matrix: Array<IntArray>): Array<IntArray> {
    val rows = matrix.size
    val cols = matrix[0].size
    val transposed = Array(cols) { IntArray(rows) }

    for (i in 0 until rows) {
        for (j in 0 until cols) {
            transposed[j][i] = matrix[i][j]
        }
    }
    return transposed
}

fun main() {
    val matrix = arrayOf(
        intArrayOf(1, 2, 3),
        intArrayOf(4, 5, 6),
        intArrayOf(7, 8, 9)
    )

    val transposedMatrix = transposeMatrix(matrix)
    println("Транспонированная матрица:")
    for (row in transposedMatrix) {
        println(row.joinToString())
    }
}
```



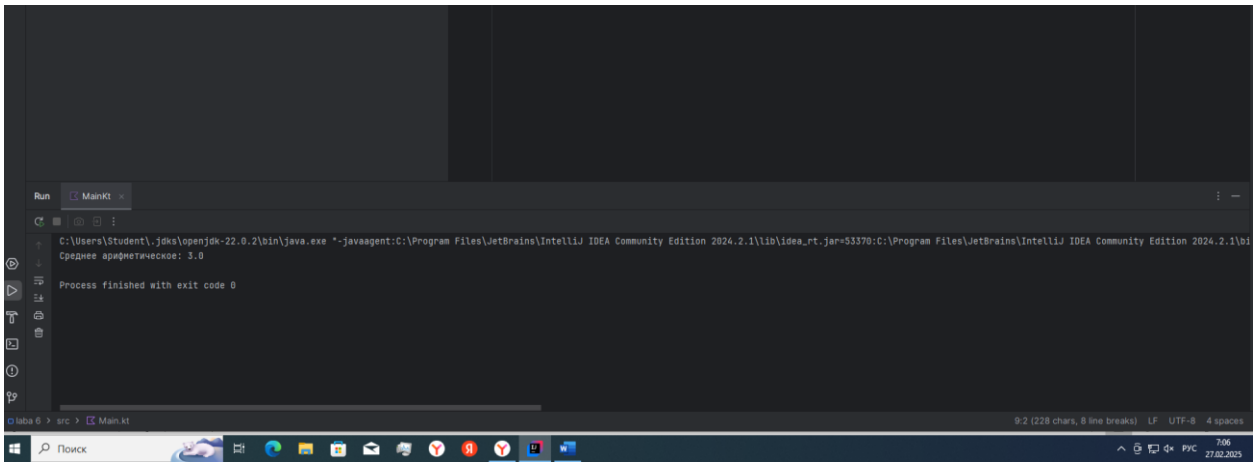
25.Линейный поиск: Реализуйте линейный поиск элемента в массиве с возвратомBool-значения (найден или нет).

```
fun linearSearch(array: IntArray, target: Int): Boolean {
    for (element in array) {
        if (element == target) return true
    }
    return false
}

fun main() {
    val numbers = intArrayOf(1, 2, 3, 4, 5)
    val found = linearSearch(numbers, 3)
    println("Элемент найден: $found")
}
```



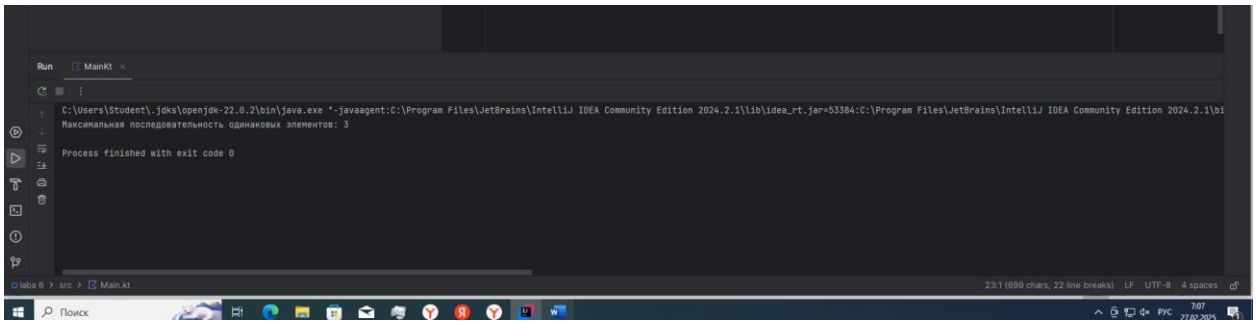
26.Среднее арифметическое: Напишите программу, которая находит среднее арифметическое всех чисел в массиве.

```
fun calculateAverage(array: IntArray): Double {
    return array.average()
}

fun main() {
    val numbers = intArrayOf(1, 2, 3, 4, 5)
    val average = calculateAverage(numbers)
    println("Среднее арифметическое: $average")
}
```



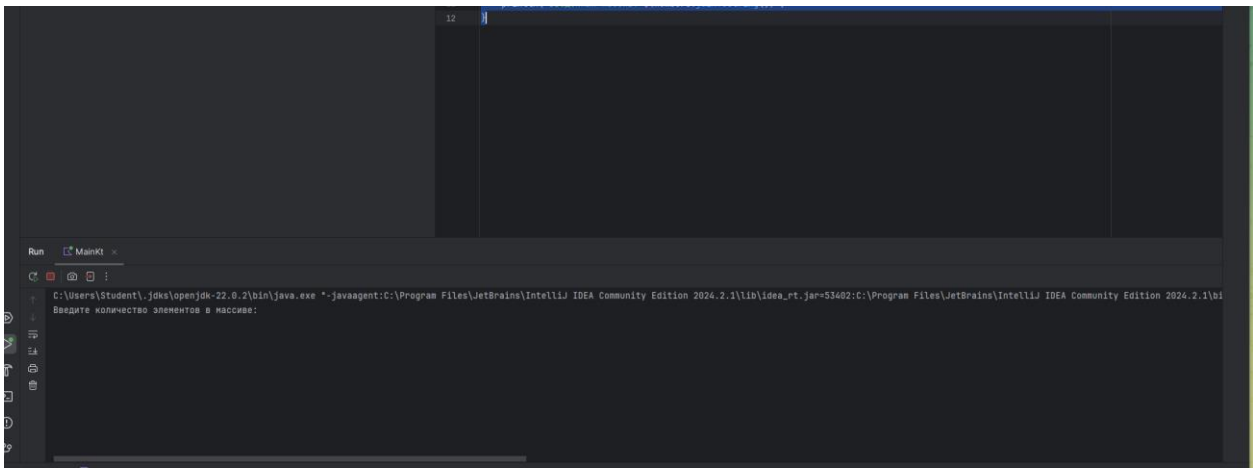
27.Максимальная последовательность: Найдите максимальную последовательность одинаковых элементов в массиве.

```
fun maxSequenceLength(array: IntArray): Pair<Int, Int> {
    var maxLength = 1
    var currentLength = 1

    for (i in 1 until array.size) {
        if (array[i] == array[i - 1]) {
            currentLength++
        } else {
            maxLength = maxOf(maxLength, currentLength)
            currentLength = 1
        }
    }

    maxLength = maxOf(maxLength, currentLength) // проверка в конце массива
    return maxLength to array.size // возвращаем максимальную длину и размер
массива
}

fun main() {
    val numbers = intArrayOf(1, 1, 2, 2, 2, 3, 3)
    val (maxLength, totalSize) = maxSequenceLength(numbers)
    println("Максимальная последовательность одинаковых элементов:
$maxLength")
}
```



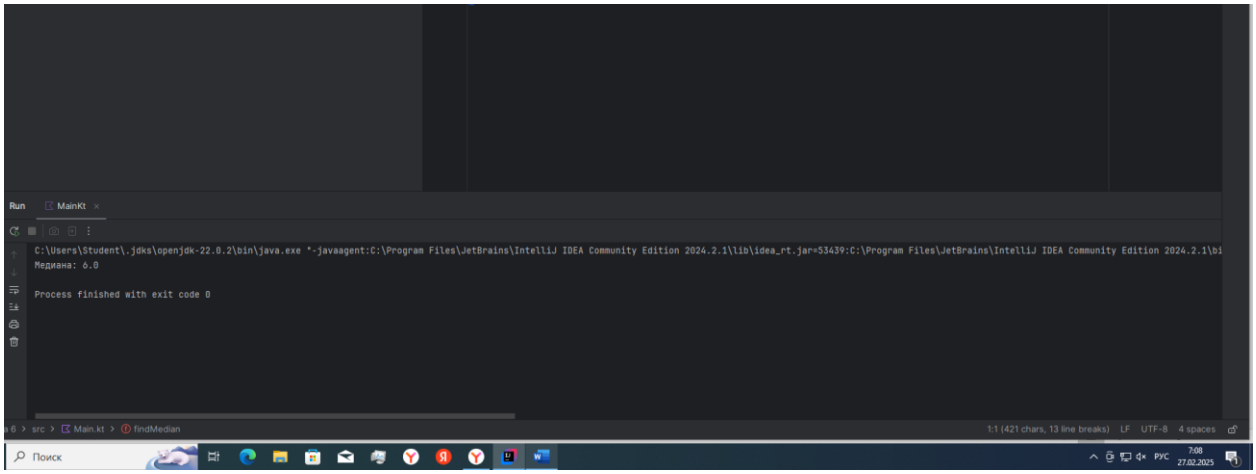
28.Ввод и вывод массива: Напишите программу, которая запрашивает у пользователя ввод массива чисел и затем выводит его.

```
fun main() {
    println("Введите количество элементов в массиве:")
    val size = readLine()?.toIntOrNull() ?: return
    val numbers = IntArray(size)

    println("Введите элементы массива:")
    for (i in numbers.indices) {
        numbers[i] = readLine()?.toIntOrNull() ?: return
    }

    println("Введенный массив: ${numbers.joinToString()}")
}
```



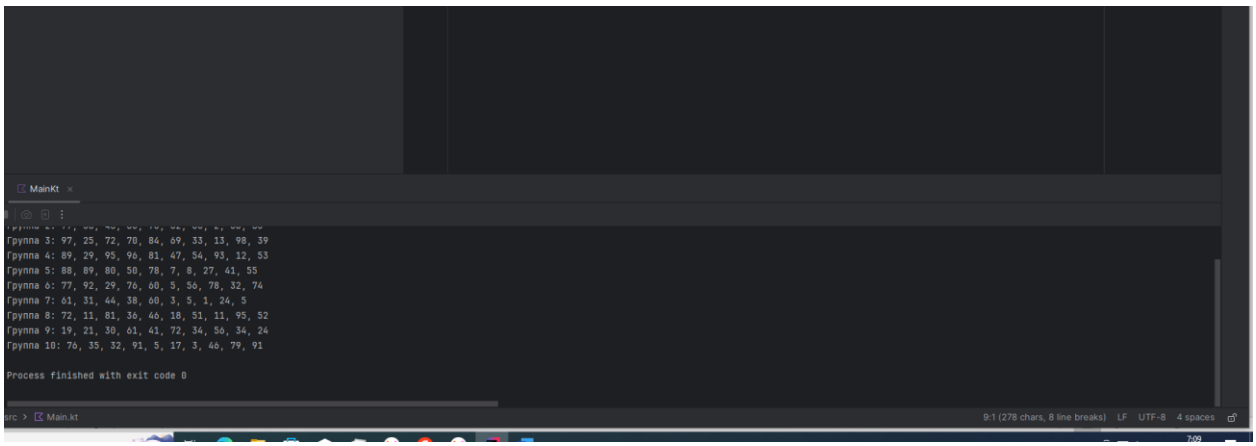
29.Нахождение медианы: Напишите программу, которая находит медиану в массиве.

```
fun findMedian(array: IntArray): Double {
    val sortedArray = array.sorted()
    return if (sortedArray.size % 2 == 0) {
        (sortedArray[sortedArray.size / 2 - 1] + sortedArray[sortedArray.size
/ 2]) / 2.0
    } else {
        sortedArray[sortedArray.size / 2].toDouble()
    }
}

fun main() {
    val numbers = intArrayOf(1, 3, 3, 6, 7, 8, 9)
    val median = findMedian(numbers)
    println("Медиана: $median")
}
```



30.Распределение по группам: Создайте массив из 100 целых чисел и разделите их на 10 групп по 10 элементов, затем выведите результаты.

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

    for (i in randomNumbers.indices step 10) {
        val group = randomNumbers.copyOfRange(i, minOf(i + 10,
randomNumbers.size))
        println("Группа ${i / 10 + 1}: ${group.joinToString()}")
    }
}
```


MainKt
Группа 3: 97, 25, 72, 70, 84, 69, 33, 13, 98, 39
Группа 4: 89, 29, 95, 96, 81, 47, 54, 93, 12, 53
Группа 5: 88, 89, 80, 50, 78, 7, 8, 27, 41, 55
Группа 6: 77, 92, 29, 76, 60, 5, 56, 78, 32, 74
Группа 7: 61, 31, 44, 38, 60, 3, 5, 1, 24, 5
Группа 8: 72, 11, 81, 36, 46, 18, 51, 11, 95, 52
Группа 9: 19, 21, 30, 61, 41, 72, 34, 56, 34, 24
Группа 10: 76, 35, 32, 91, 5, 17, 3, 46, 79, 91
Process finished with exit code 0
src > Main.kt
9:1 (278 chars, 8 line breaks) LF UTF-8 4 spaces