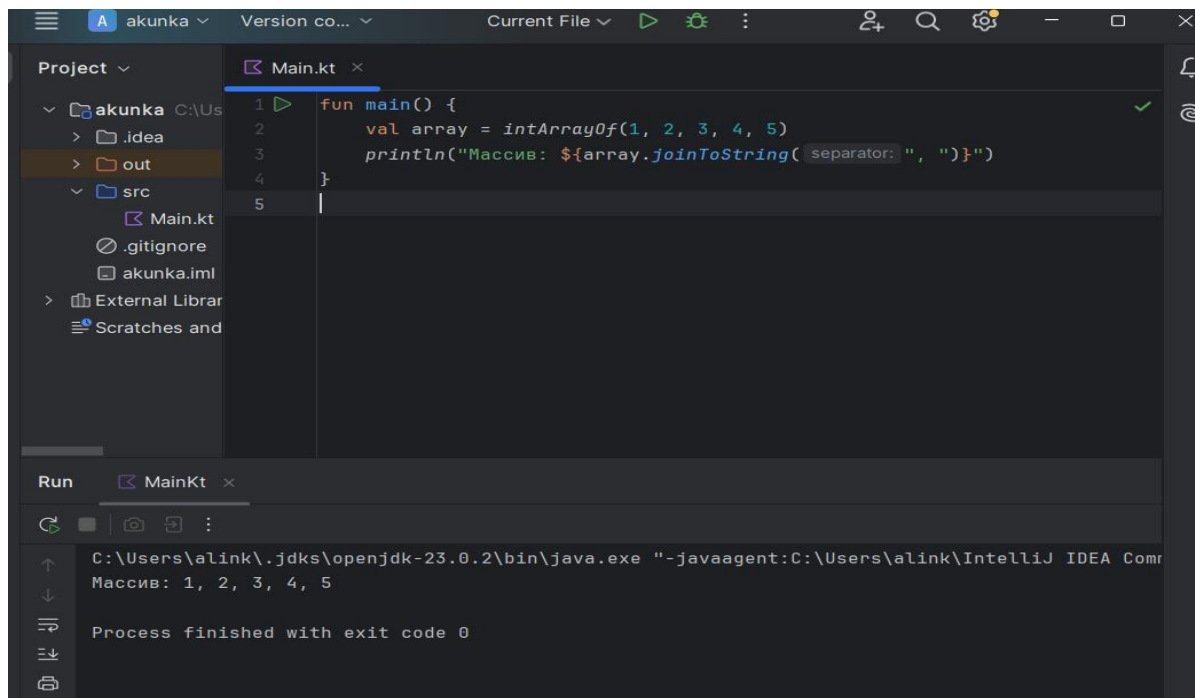


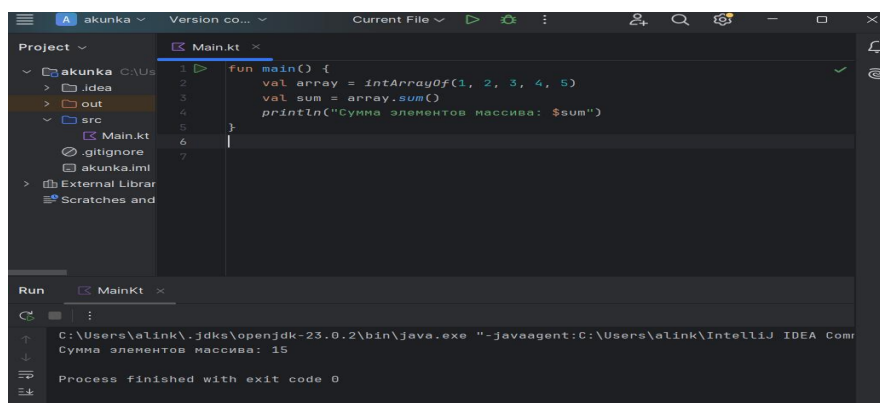
Практическая работа № 6.

Выполнили : Андрухова , Загородняя.

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



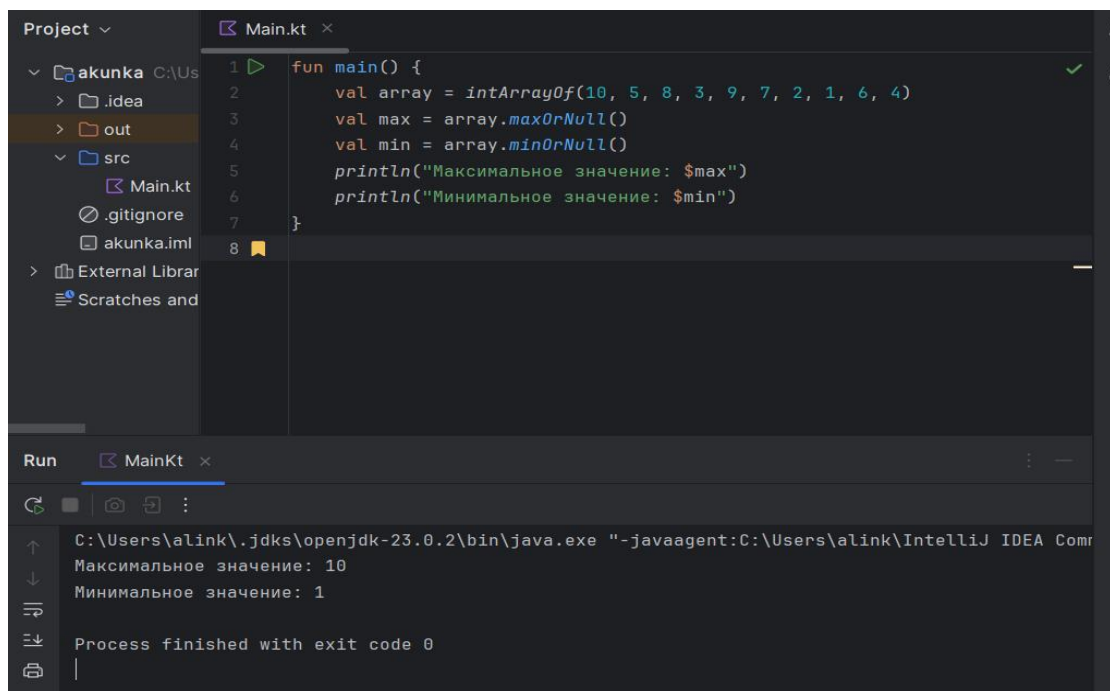
```
2.fun main() {  
    val array = intArrayOf(1, 2, 3, 4, 5)  
    val sum = array.sum()  
    println("Сумма элементов массива: $sum")  
}
```



```

3.fun main() {
    val array = intArrayOf(10, 5, 8, 3, 9, 7, 2, 1, 6, 4)
    val max = array.maxOrNull()
    val min = array.minOrNull()
    println("Максимальное значение: $max")
    println("Минимальное значение: $min")
}

```

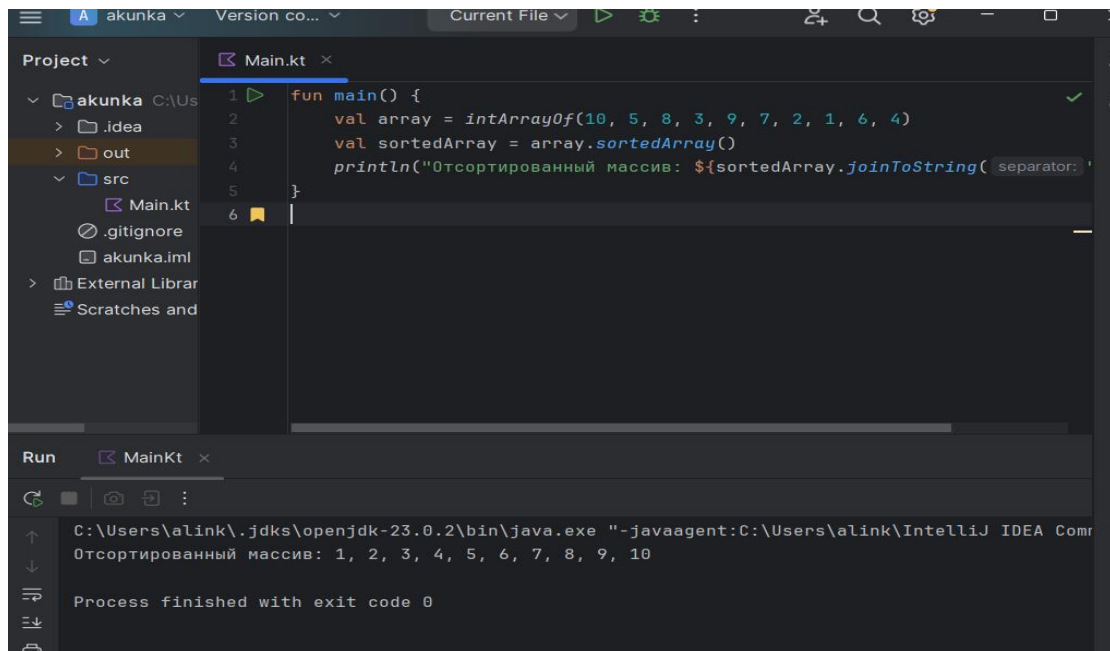


4.

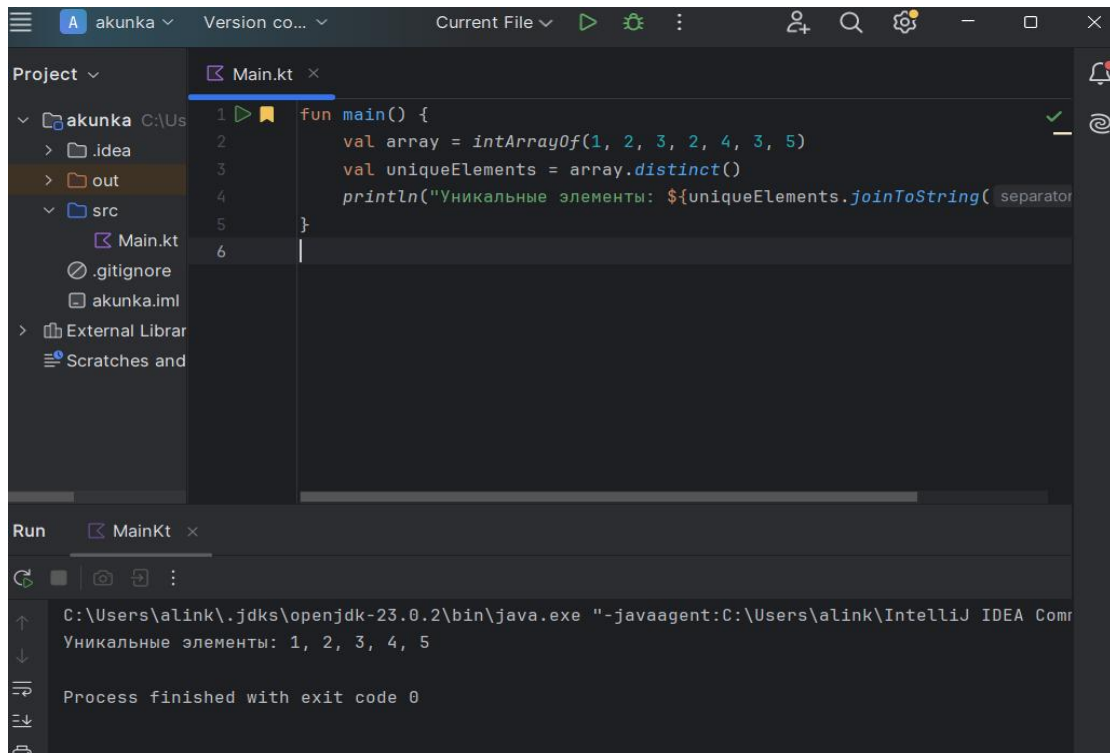
```

fun main() {
    val array = intArrayOf(10, 5, 8, 3, 9, 7, 2, 1, 6, 4)
    val sortedArray = array.sortedArray()
    println("Отсортированный массив: ${sortedArray.joinToString(",
    ")}")
}

```



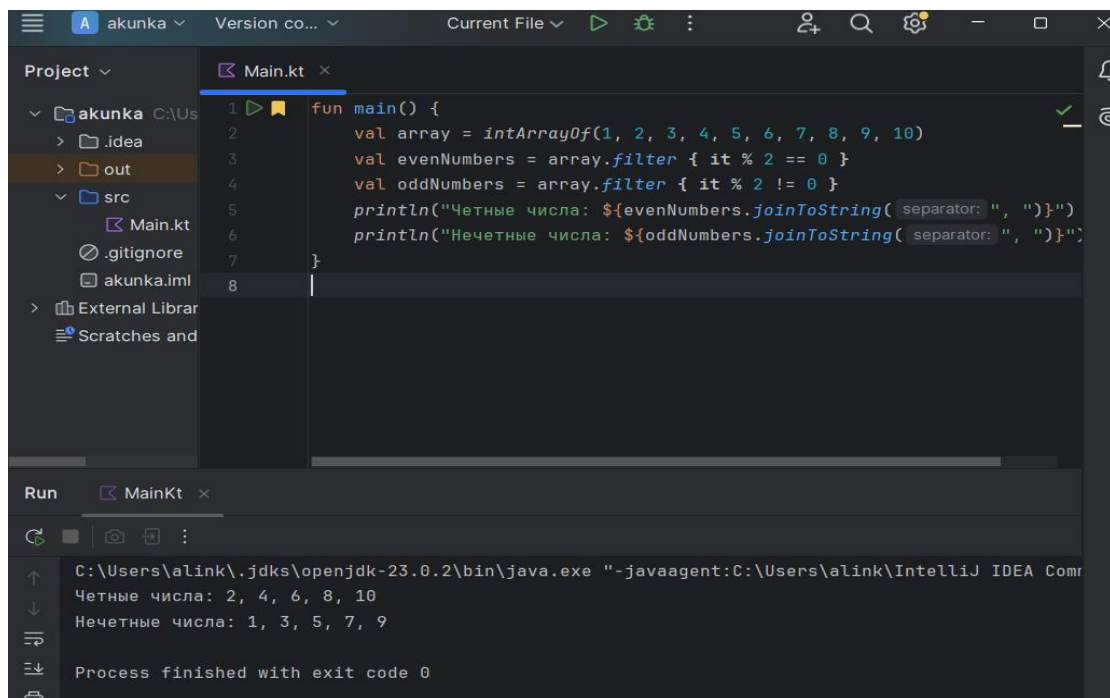
```
5.fun main() {  
    val array = intArrayOf(1, 2, 3, 2, 4, 3, 5)  
    val uniqueElements = array.distinct()  
    println("Уникальные элементы: ${uniqueElements.joinToString(",  
    ")}")  
}
```



```

6.fun main() {
    val array = intArrayOf(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
    val evenNumbers = array.filter { it % 2 == 0 }
    val oddNumbers = array.filter { it % 2 != 0 }
    println("Четные числа: ${evenNumbers.joinToString(", ")}")
    println("Нечетные числа: ${oddNumbers.joinToString(", ")}")
}

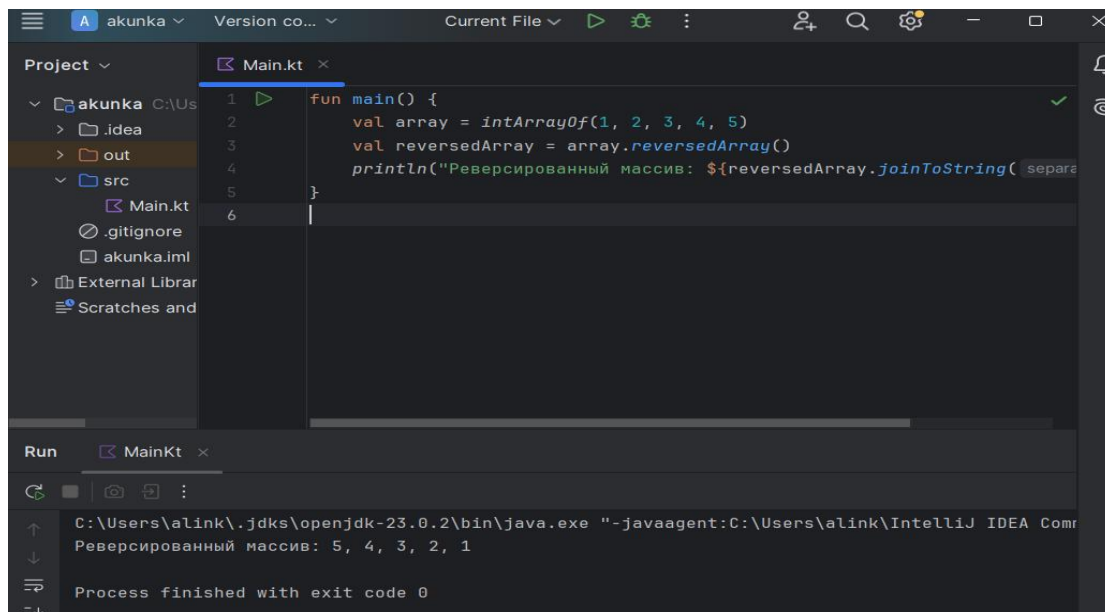
```



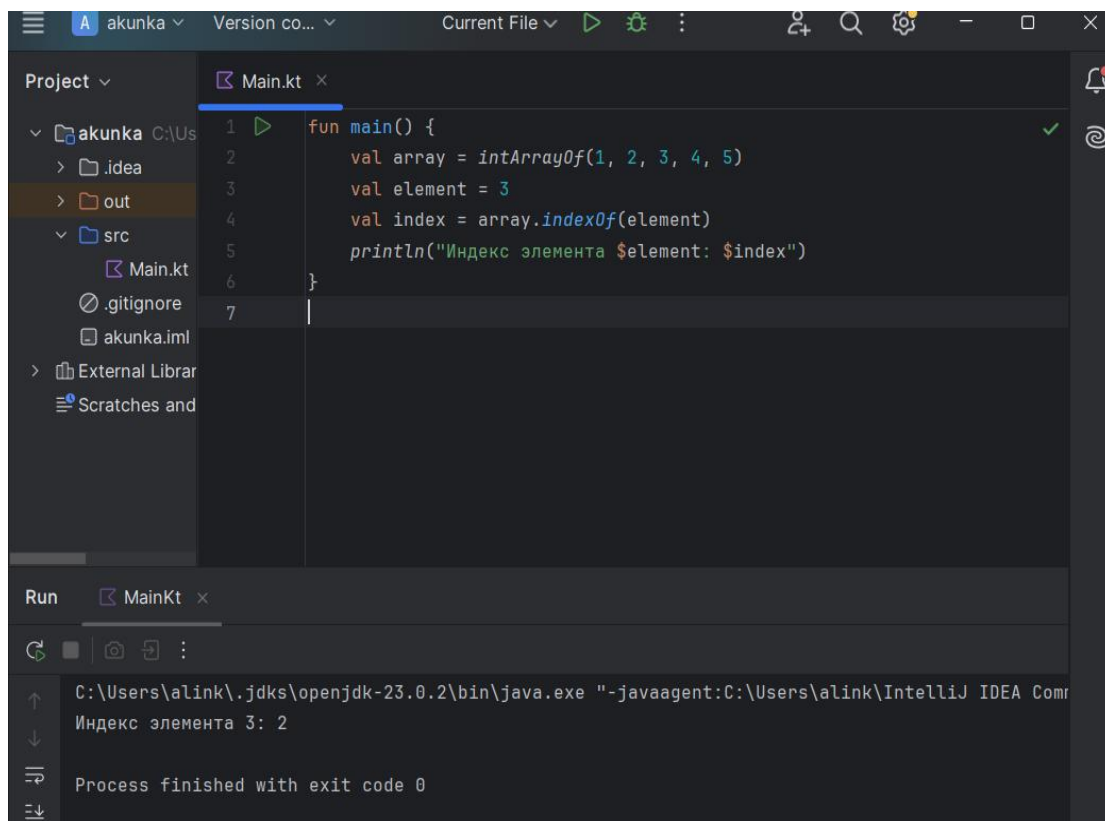
```

7.fun main() {
    val array = intArrayOf(1, 2, 3, 4, 5)
    val reversedArray = array.reversedArray()
    println("Реверсированный массив: ${reversedArray.joinToString(", ")})")
}

```



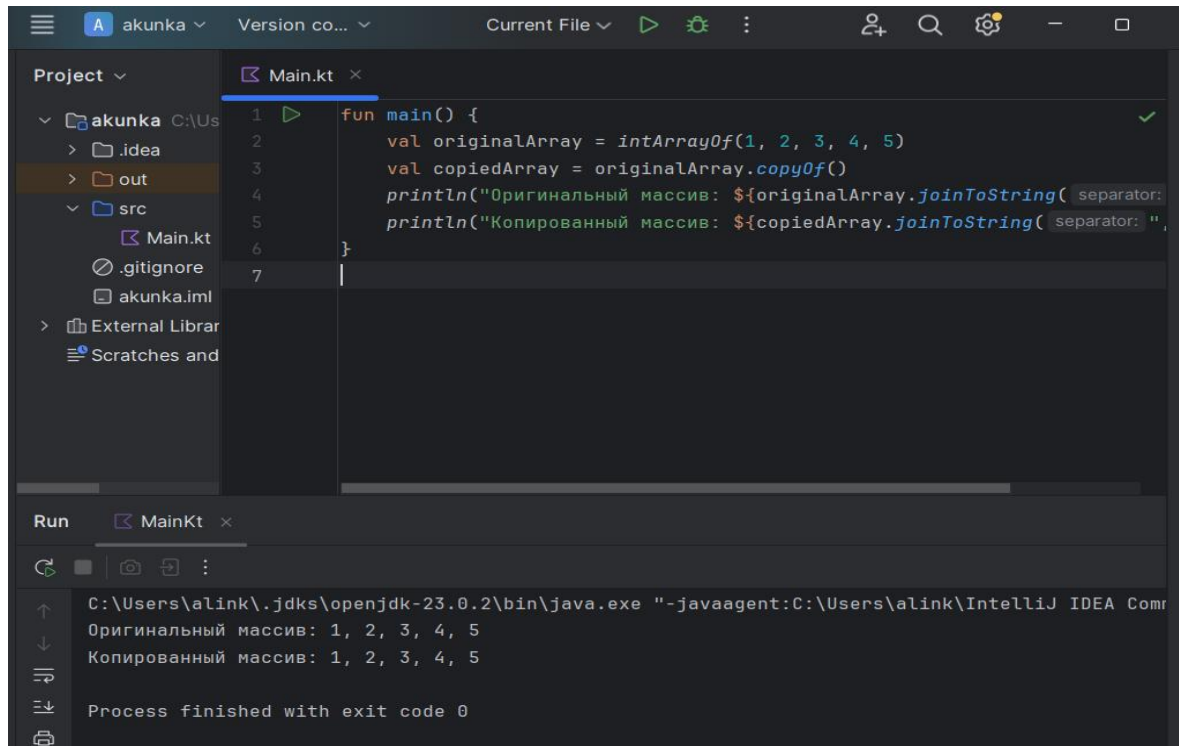
```
8.fun main() {  
    val array = intArrayOf(1, 2, 3, 4, 5)  
    val element = 3  
    val index = array.indexOf(element)  
    println("Индекс элемента $element: $index")  
}
```



```

9.fun main() {
    val originalArray = intArrayOf(1, 2, 3, 4, 5)
    val copiedArray = originalArray.copyOf()
    println("Оригинальный массив: ${originalArray.joinToString(", ")}")
    println("Копированный массив: ${copiedArray.joinToString(", ")}")
}

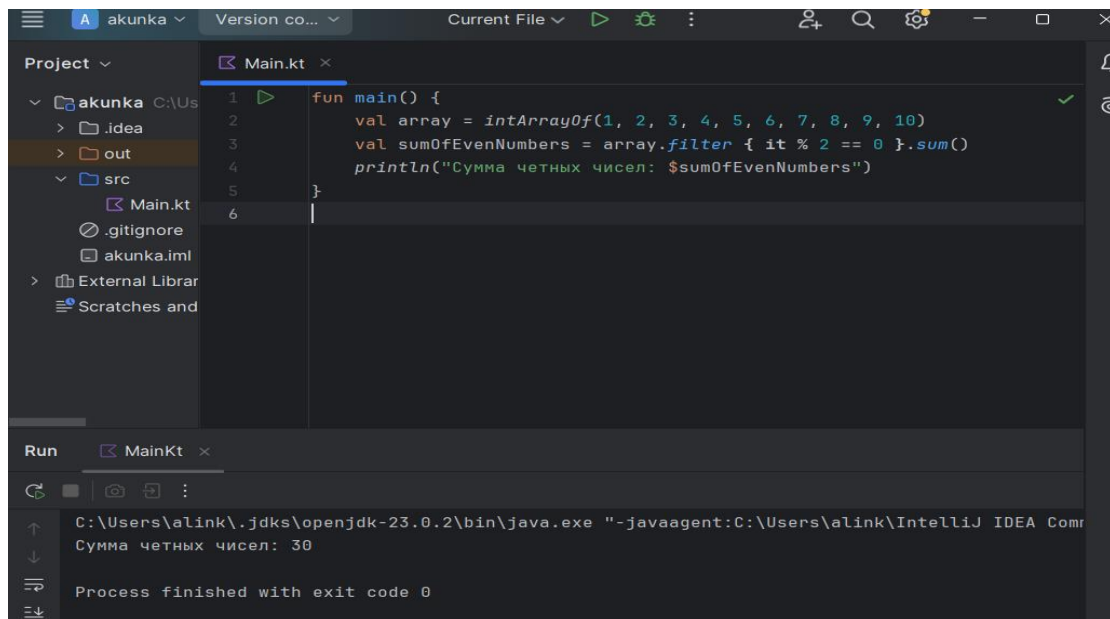
```



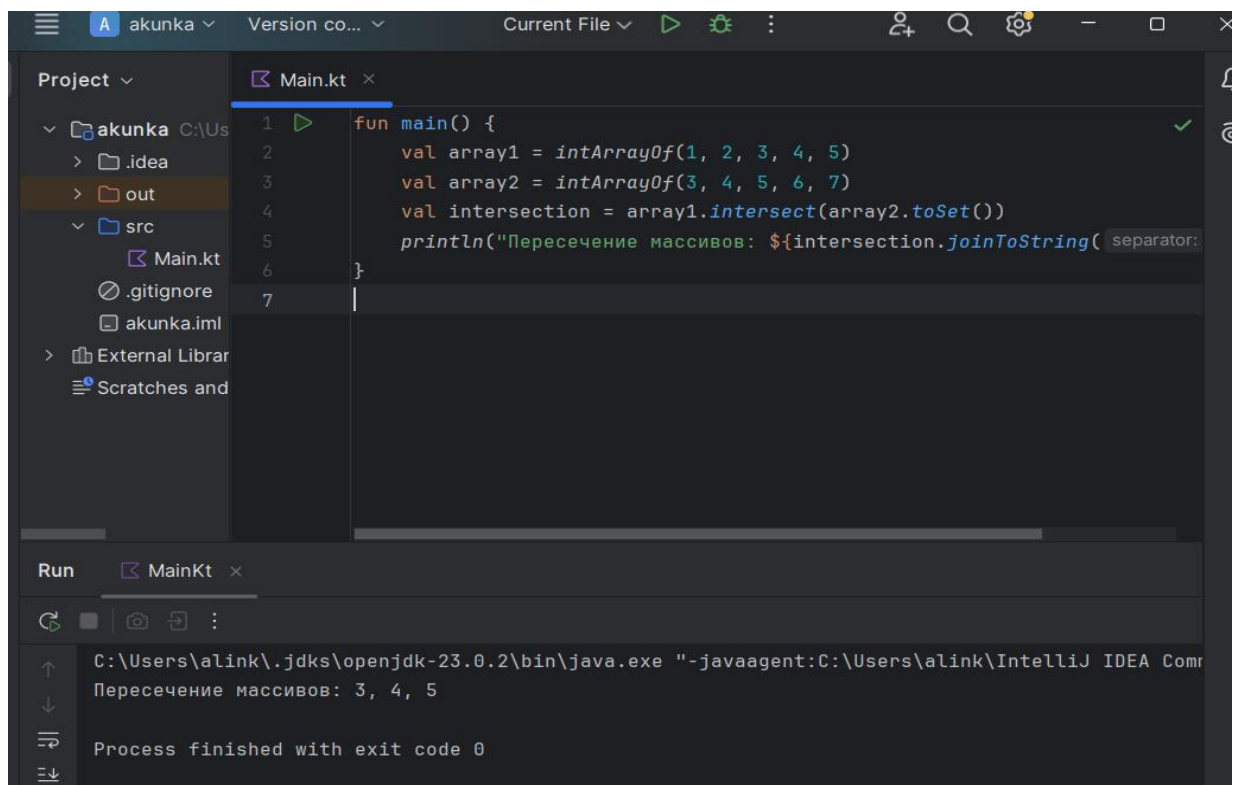
```

10.fun main() {
    val array = intArrayOf(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
    val sumOfEvenNumbers = array.filter { it % 2 == 0 }.sum()
    println("Сумма четных чисел: $sumOfEvenNumbers")
}

```



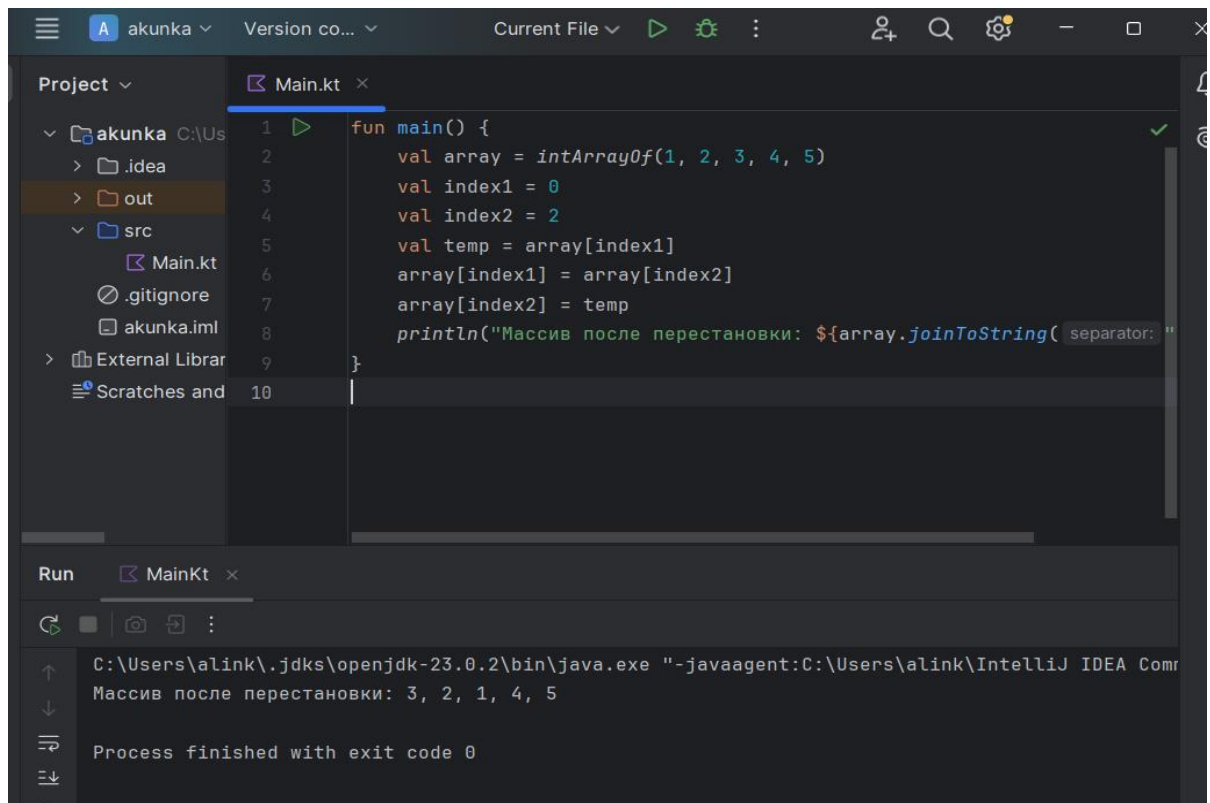
```
11.fun main() {  
    val array1 = intArrayOf(1, 2, 3, 4, 5)  
    val array2 = intArrayOf(3, 4, 5, 6, 7)  
    val intersection = array1.intersect(array2.toSet())  
    println("Пересечение массивов: ${intersection.joinToString(", ")}")  
}
```




```

12.fun main() {
    val array = intArrayOf(1, 2, 3, 4, 5)
    val index1 = 0
    val index2 = 2
    val temp = array[index1]
    array[index1] = array[index2]
    array[index2] = temp
    println("Массив после перестановки: ${array.joinToString(", ")}")
}

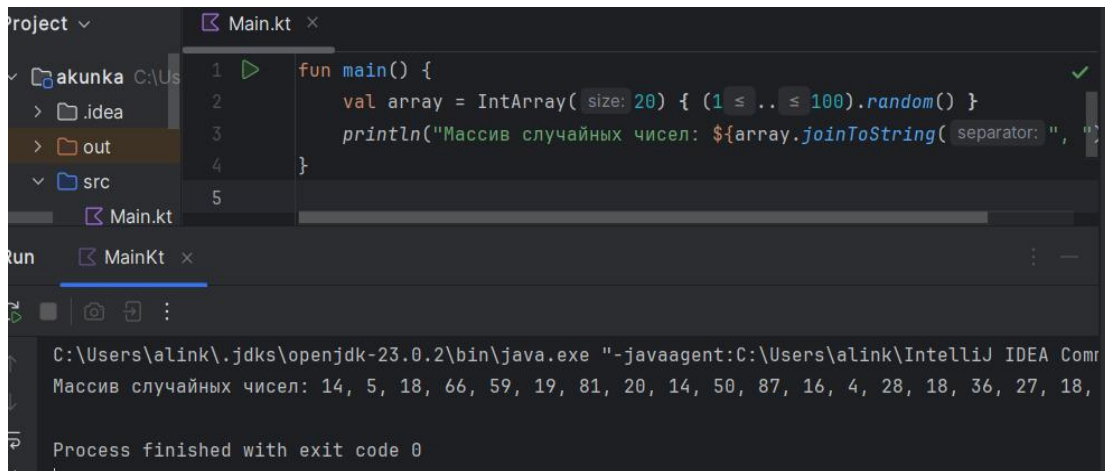
```



```

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

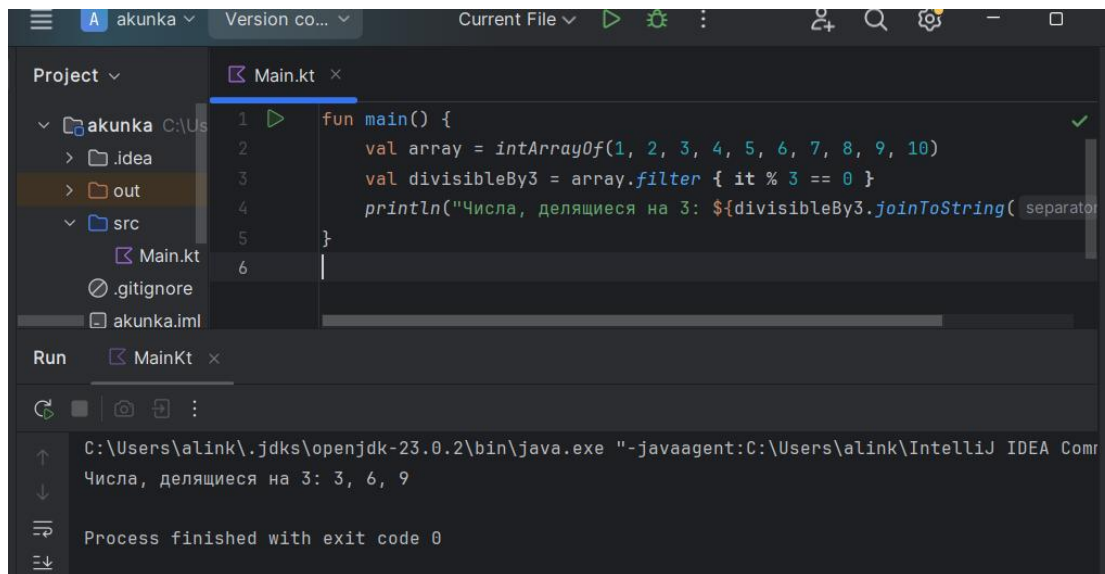
```

```
Project ▾ Main.kt ×
1 fun main() {
2     val array = IntArray(size: 20) { (1 ≤ .. ≤ 100).random() }
3     println("Массив случайных чисел: ${array.joinToString(separator: ", ")}")
4 }
5

Run MainKt ×
C:\Users\alink\jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Users\alink\IntelliJ IDEA Comr
Массив случайных чисел: 14, 5, 18, 66, 59, 19, 81, 20, 14, 50, 87, 16, 4, 28, 18, 36, 27, 18,
Process finished with exit code 0
```

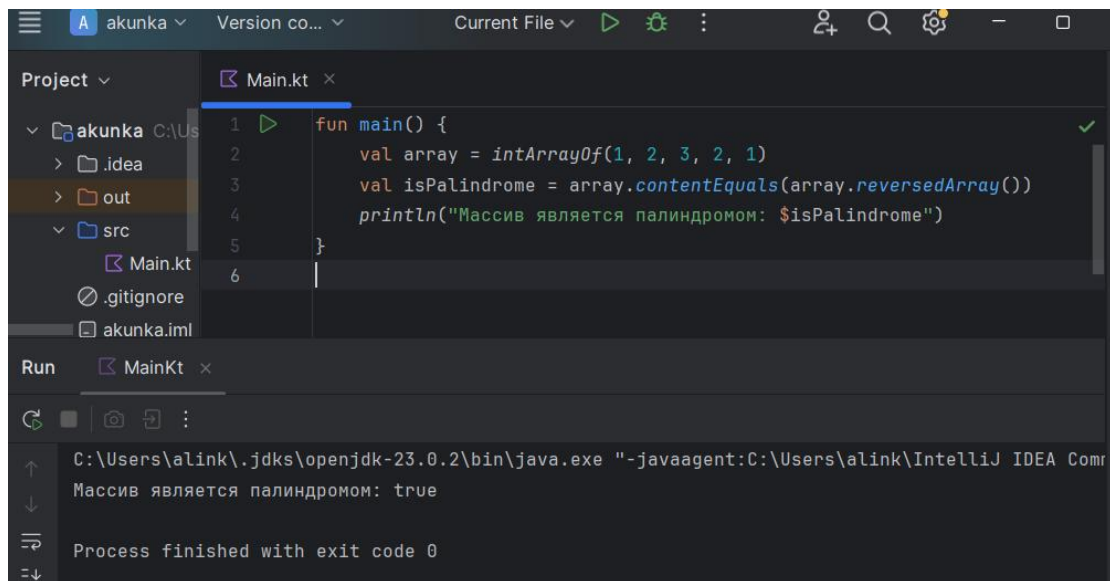
```
14.fun main() {
    val array = intArrayOf(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
    val divisibleBy3 = array.filter { it % 3 == 0 }
    println("Числа, делящиеся на 3: ${divisibleBy3.joinToString(", ")}")
}
```



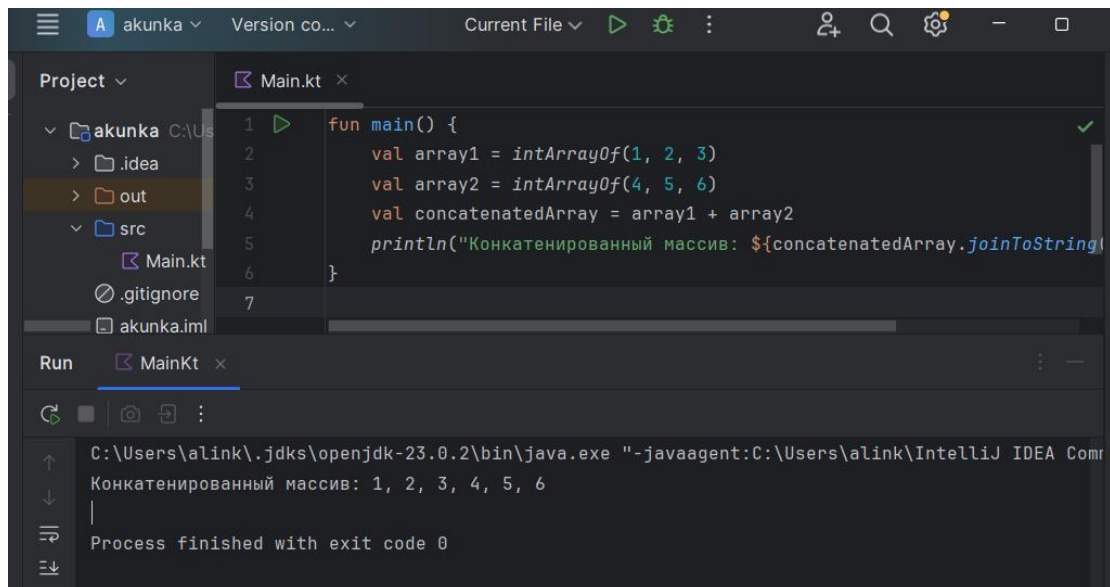
```
Project ▾ Main.kt ×
1 fun main() {
2     val array = intArrayOf(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
3     val divisibleBy3 = array.filter { it % 3 == 0 }
4     println("Числа, делящиеся на 3: ${divisibleBy3.joinToString(separator: ", ")}")
5 }
6

Run MainKt ×
C:\Users\alink\jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Users\alink\IntelliJ IDEA Comr
Числа, делящиеся на 3: 3, 6, 9
Process finished with exit code 0
```

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



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



```
17.fun main() {
    val array = intArrayOf(1, 2, 3, 4, 5)
    val sum = array.sum()
}
```

```

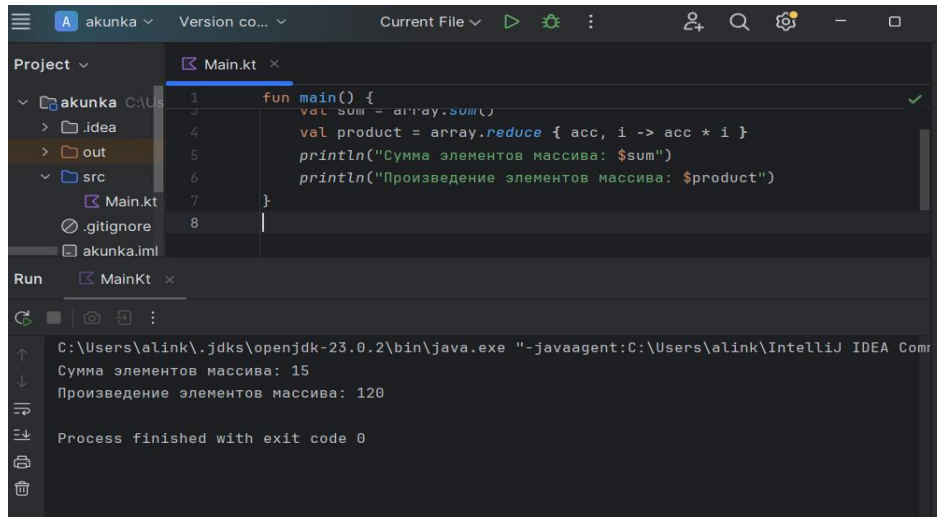
val product = array.reduce { acc, i -> acc * i }

println("Сумма элементов массива: $sum")

println("Произведение элементов массива: $product")

}

```



```

18.fun main() {

    val array = listOf(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16)

    val groups = array.chunked(5)

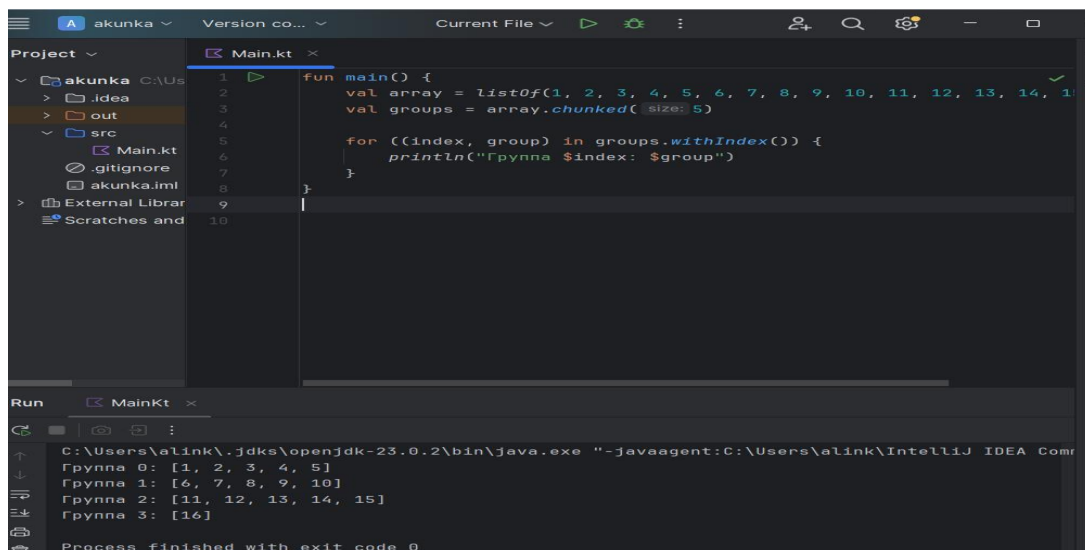
    for ((index, group) in groups.withIndex()) {

        println("Группа $index: $group")

    }

}

```



```

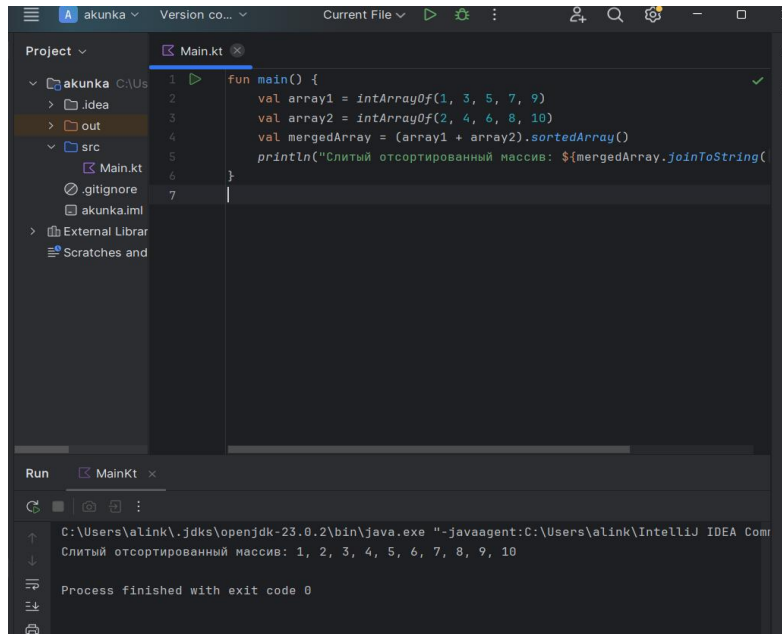
19.fun main() {

```

```

val array1 = intArrayOf(1, 3, 5, 7, 9)
val array2 = intArrayOf(2, 4, 6, 8, 10)
val mergedArray = (array1 + array2).sortedArray()
println("Слитый отсортированный массив:
${mergedArray.joinToString(", ")}")
}

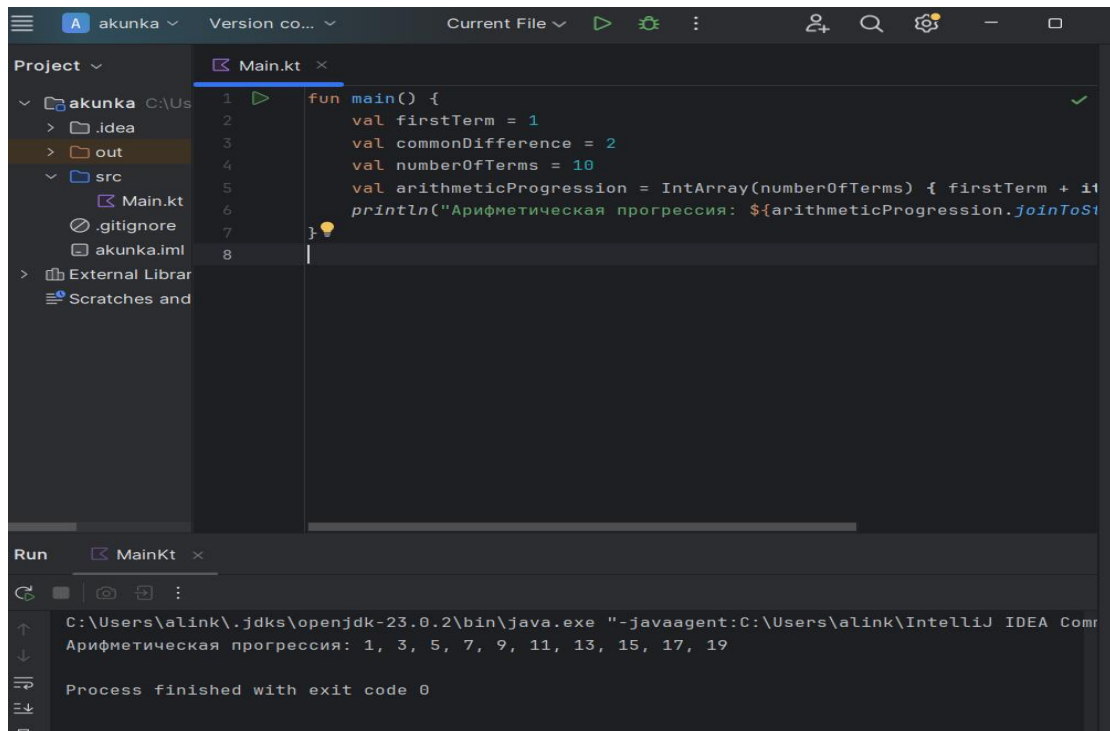
```



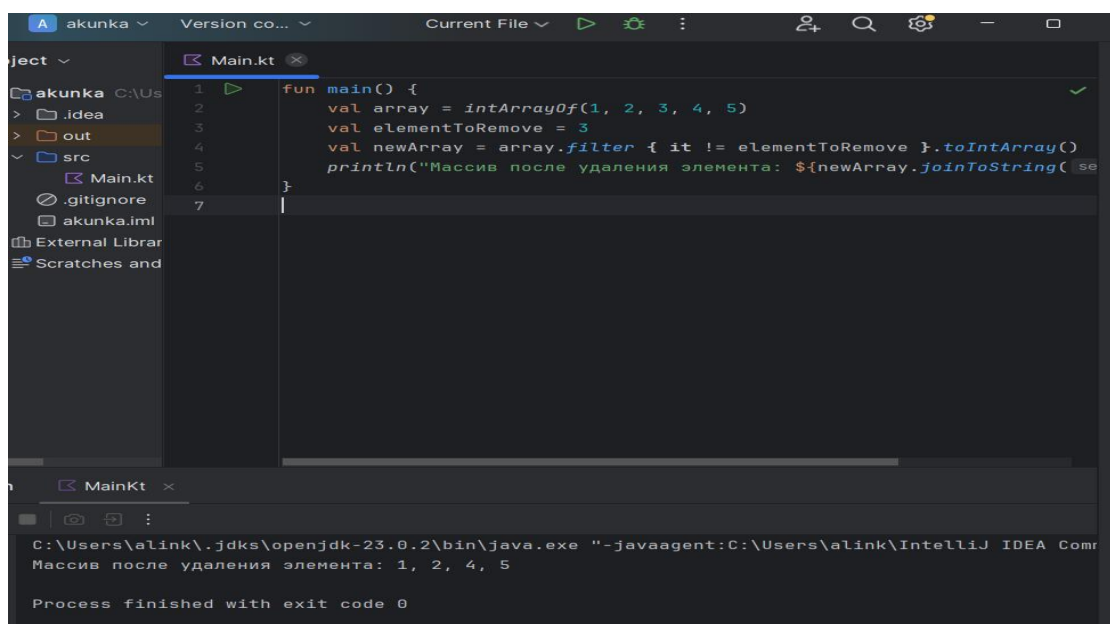
```

20.fun main() {
    val firstTerm = 1
    val commonDifference = 2
    val numberOfTerms = 10
    val arithmeticProgression = IntArray(numberOfTerms) { firstTerm + it
* commonDifference }
    println("Арифметическая прогрессия:
${arithmeticProgression.joinToString(", ")}")
}

```



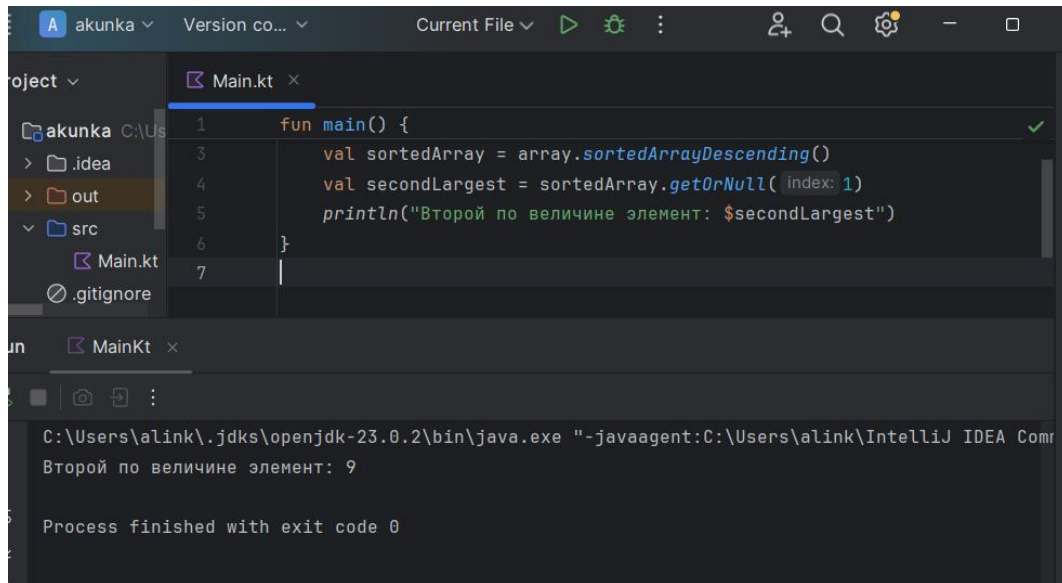
```
21.fun main() {  
    val array = intArrayOf(1, 2, 3, 4, 5)  
    val elementToRemove = 3  
    val newArray = array.filter { it != elementToRemove }.toIntArray()  
    println("Массив после удаления элемента:  
${newArray.joinToString(", ")}")  
}
```



```

22.fun main() {
    val array = intArrayOf(10, 5, 8, 3, 9, 7, 2, 1, 6, 4)
    val sortedArray = array.sortedArrayDescending()
    val secondLargest = sortedArray.getOrNull(1)
    println("Второй по величине элемент: $secondLargest")
}

```



```

23.fun main() {
    val array1 = intArrayOf(1, 2, 3)
    val array2 = intArrayOf(4, 5, 6)
    val array3 = intArrayOf(7, 8, 9)
    val combinedArray = array1 + array2 + array3
    println("Объединенный массив: ${combinedArray.joinToString(",
    ")}")
}

```

The screenshot shows the IntelliJ IDEA interface. The Project view on the left shows a project named 'akunka' with a file 'Main.kt'. The editor displays the following Kotlin code:

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

The Run view at the bottom shows the output: "Объединенный массив: 1, 2, 3, 4, 5, 6, 7, 8, 9".

```
24.fun main() {  
    val matrix = arrayOf(  
        intArrayOf(1, 2, 3),  
        intArrayOf(4, 5, 6),  
        intArrayOf(7, 8, 9)  
    )  
    val transposedMatrix = matrix.mapIndexed { i, row ->  
row.mapIndexed { j, _ -> matrix[j][i] }.toIntArray() }  
    println("Транспонированная матрица:")  
    transposedMatrix.forEach { println(it.joinToString(", ")) }  
}
```

The screenshot shows the IntelliJ IDEA interface. The Project view on the left shows a project named 'akunka' with a file 'Main.kt'. The editor displays the following Kotlin code:

```
1 fun main() {  
2     val matrix = arrayOf(  
3         intArrayOf(1, 2, 3),  
4         intArrayOf(4, 5, 6),  
5         intArrayOf(7, 8, 9)  
6     )  
7     val transposedMatrix = matrix.mapIndexed { i, row -> row.mapIndexed  
8     println("Транспонированная матрица:")  
9     transposedMatrix.forEach { println(it.joinToString(separator: ", ")) }  
10 }  
11
```

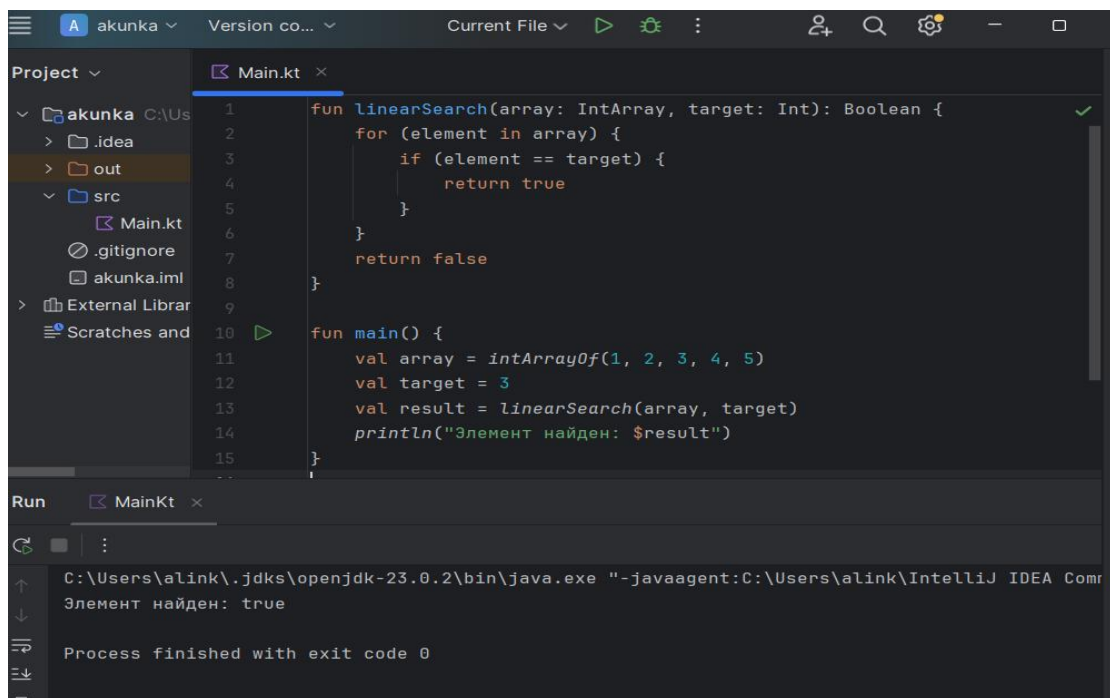
The Run view at the bottom shows the output: "Транспонированная матрица:" followed by three lines: "1, 4, 7", "2, 5, 8", and "3, 6, 9".


```

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

fun main() {
    val array = intArrayOf(1, 2, 3, 4, 5)
    val target = 3
    val result = linearSearch(array, target)
    println("Элемент найден: $result")
}

```

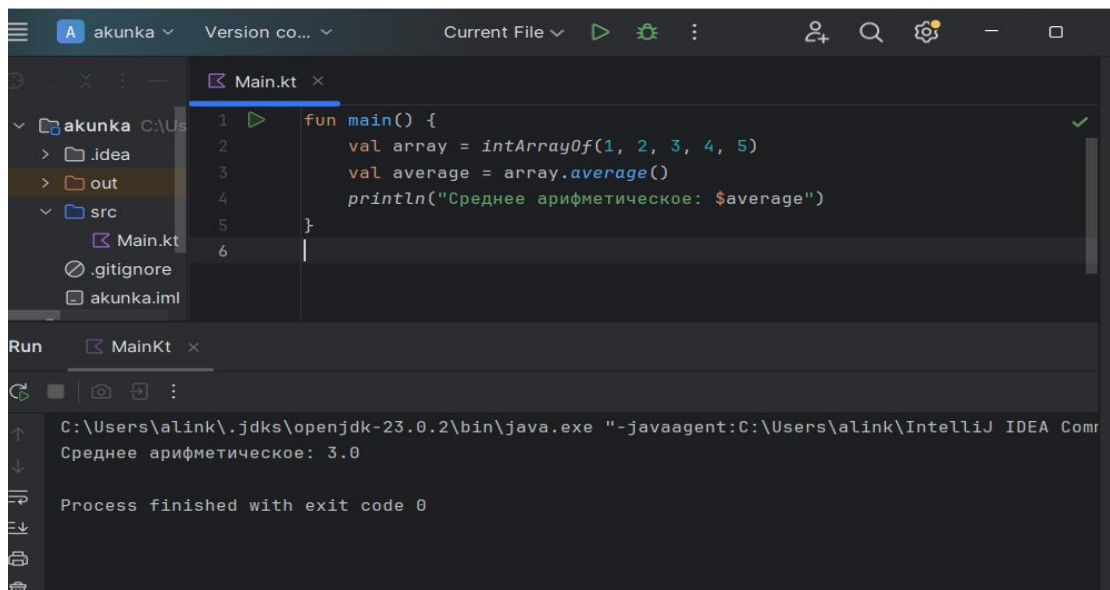


```

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

```

}



```
27.fun main() {  
    val array = intArrayOf(1, 1, 1, 2, 2, 3, 3, 3, 3, 4)  
    var maxLength = 1  
    var currentLength = 1  
    var maxElement = array[0]  
    for (i in 1 until array.size) {  
        if (array[i] == array[i - 1]) {  
            currentLength++  
            if (currentLength > maxLength) {  
                maxLength = currentLength  
                maxElement = array[i]  
            }  
        } else {  
            currentLength = 1  
        }  
    }  
    println("Максимальная последовательность: $maxLength  
элементов $maxElement")
```

}

```
1 fun main() {  
2     val array = IntArray(1, 1, 1, 2, 2, 3, 3, 3, 3, 4)  
3     var maxLength = 1  
4     var currentLength = 1  
5     var maxElement = array[0]  
6     for (i in 1 until array.size) {  
7         if (array[i] == array[i - 1]) {  
8             currentLength++  
9             if (currentLength > maxLength) {  
10                maxLength = currentLength  
11                maxElement = array[i]  
12            }  
13        } else {  
14            currentLength = 1  
15        }  
16    }  
17    println("Максимальная последовательность: $maxLength элементов $maxElement")  
18 }
```

Run MainKt

C:\Users\alink\jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Users\alink\IntelliJ IDEA Comr
Максимальная последовательность: 4 элементов 3

Process finished with exit code 0

```
28.fun main() {  
    print("Введите размеры массива: ")  
    val size = readLine()!!.toInt()  
    val array = IntArray(size) { i -> print("Введите элемент $i: ");  
    readLine()!!.toInt() }  
    println("Введенный массив: ${array.joinToString(", ")}")  
}
```

```
1 fun main() {  
2     print("Введите размеры массива: ")  
3     val size = readLine()!!.toInt()  
4     val array = IntArray(size) { i -> print("Введите элемент $i: "); readLine()!!.toInt() }  
5     println("Введенный массив: ${array.joinToString(separator: ", ")}")  
6 }  
7
```

Run MainKt

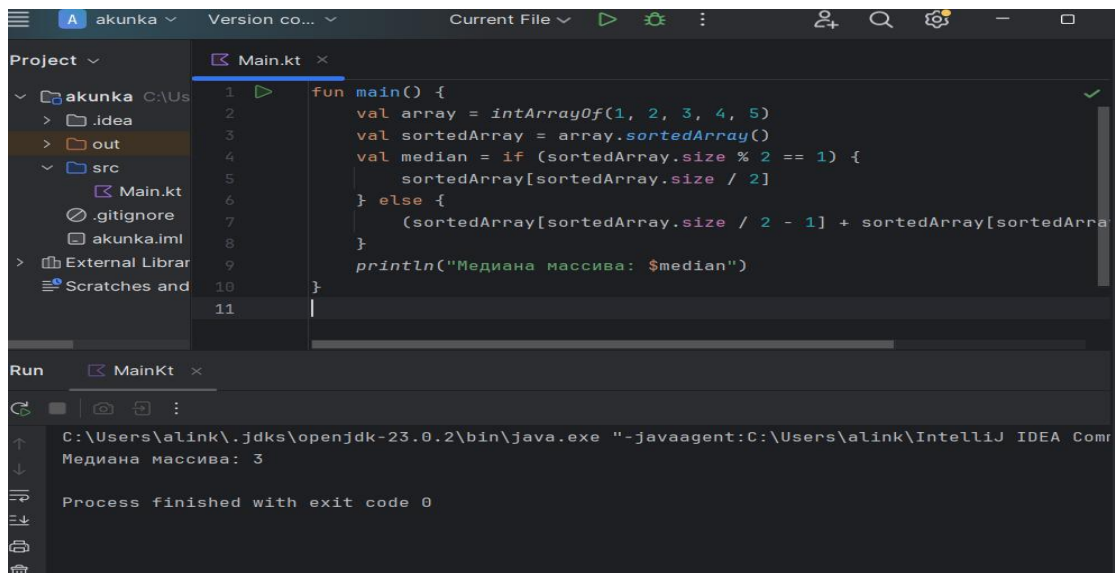
C:\Users\alink\jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Users\alink\IntelliJ IDEA Comr
Введите размеры массива: 2
Введите элемент 0: 5
Введите элемент 1: 8
Введенный массив: 5, 8

Process finished with exit code 0

```

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

```



```

30. fun main() {
    val array = (1..100).toList()
    val groups = array.chunked(10)
    for ((index, group) in groups.withIndex()) {
        println("Группа $index: $group")
    }
}

```

The screenshot displays the IntelliJ IDEA IDE interface. The top toolbar includes icons for running (a green play button), debugging (a green bug icon), and other standard IDE functions. The left sidebar shows the project structure for 'akunka', with the 'src' directory expanded to show 'Main.kt'. The main editor window displays the following Kotlin code in 'Main.kt':

```
1 fun main() {  
2     val array = (1..100).toList()  
3     val groups = array.chunked(size = 10)  
4     for ((index, group) in groups.withIndex()) {  
5         println("Группа $index: $group")  
6     }  
7 }  
8
```

Below the editor, the 'Run' tab is active, showing the execution output. The command executed is: `C:\Users\alink\.jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Users\alink\IntelliJ IDEA`. The output shows ten groups of numbers, each containing 10 elements, printed in Russian: `Группа 0: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]` through `Группа 9: [91, 92, 93, 94, 95, 96, 97, 98, 99, 100]`. The process finished with exit code 0.