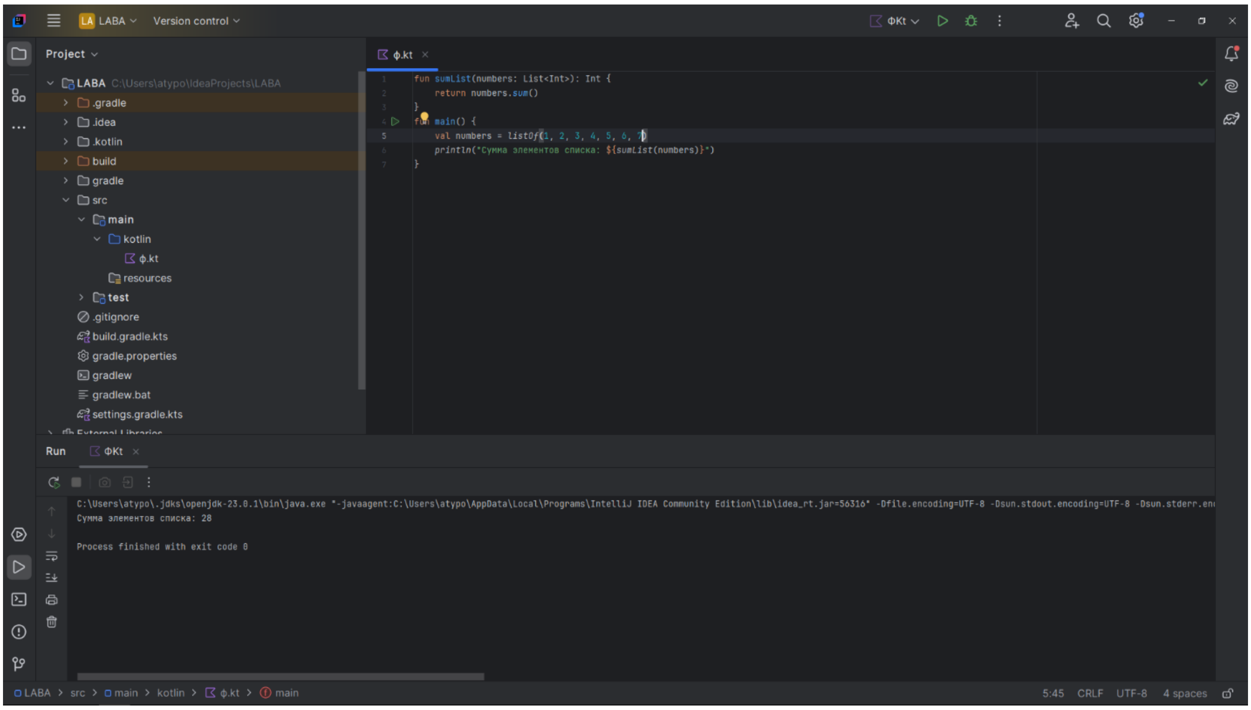
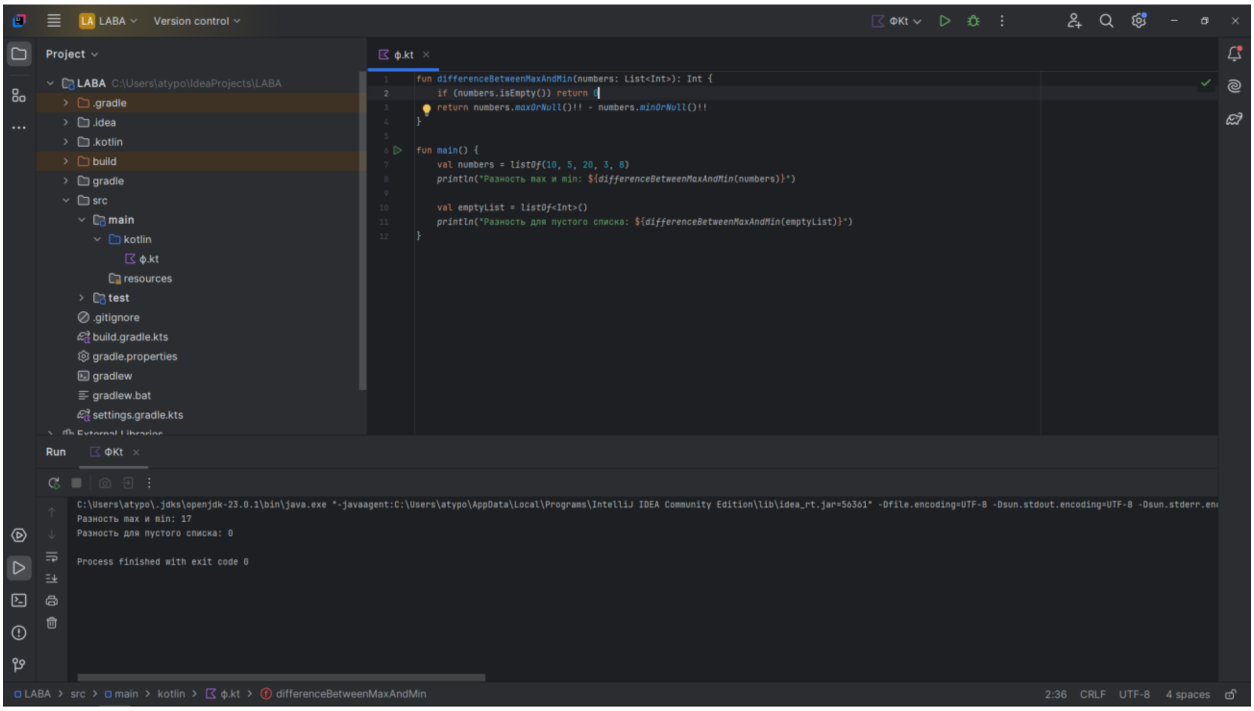
1)



```kotlin
fun sumList(numbers: List<Int>): Int {
    return numbers.sum()
}
fun main() {
    val numbers = listOf(1, 2, 3, 4, 5, 6, 7)
    println("Сумма элементов списка: ${sumList(numbers)}")
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56316" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.enc
Сумма элементов списка: 28

Process finished with exit code 0
```

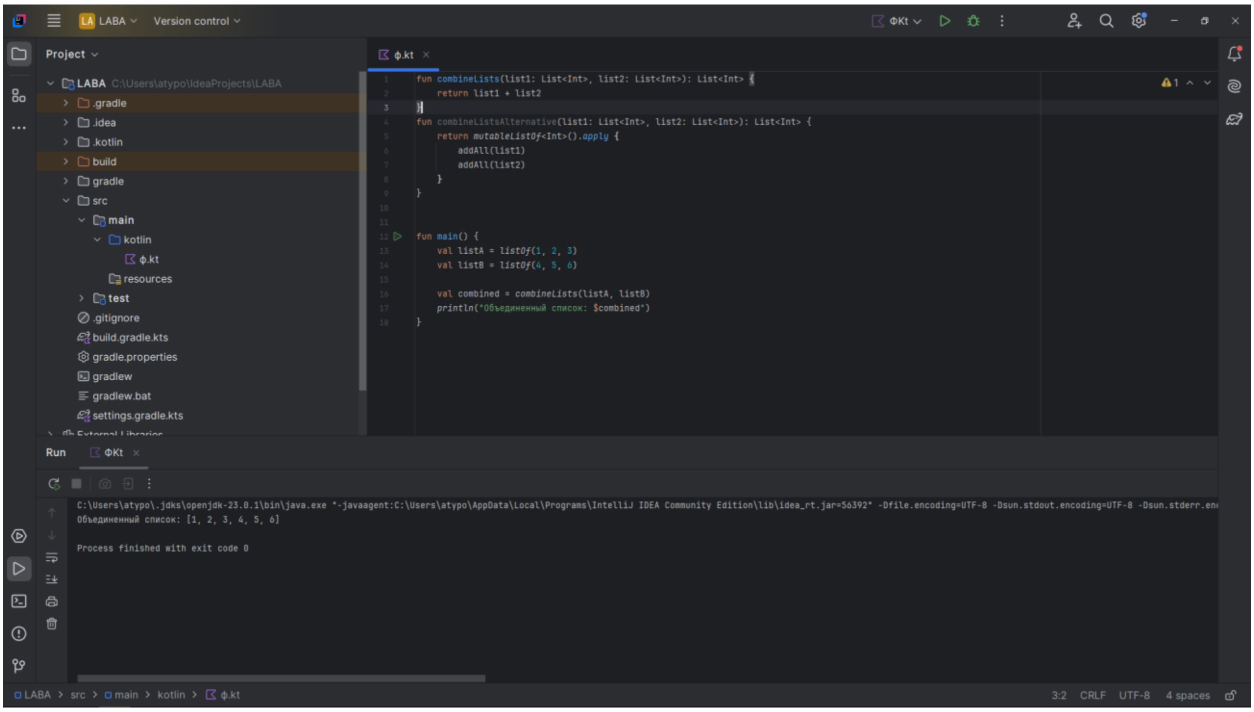
2)



```kotlin
fun differenceBetweenMaxAndMin(numbers: List<Int>): Int {
    if (numbers.isEmpty()) return 0
    return numbers.maxOrNull()!! - numbers.minOrNull()!!
}

fun main() {
    val numbers = listOf(10, 5, 20, 3, 8)
    println("Разность max и min: ${differenceBetweenMaxAndMin(numbers)}")

    val emptyList = listOf<Int>()
    println("Разность для пустого списка: ${differenceBetweenMaxAndMin(emptyList)}")
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56361" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.enc
Разность max и min: 17
Разность для пустого списка: 0

Process finished with exit code 0
```

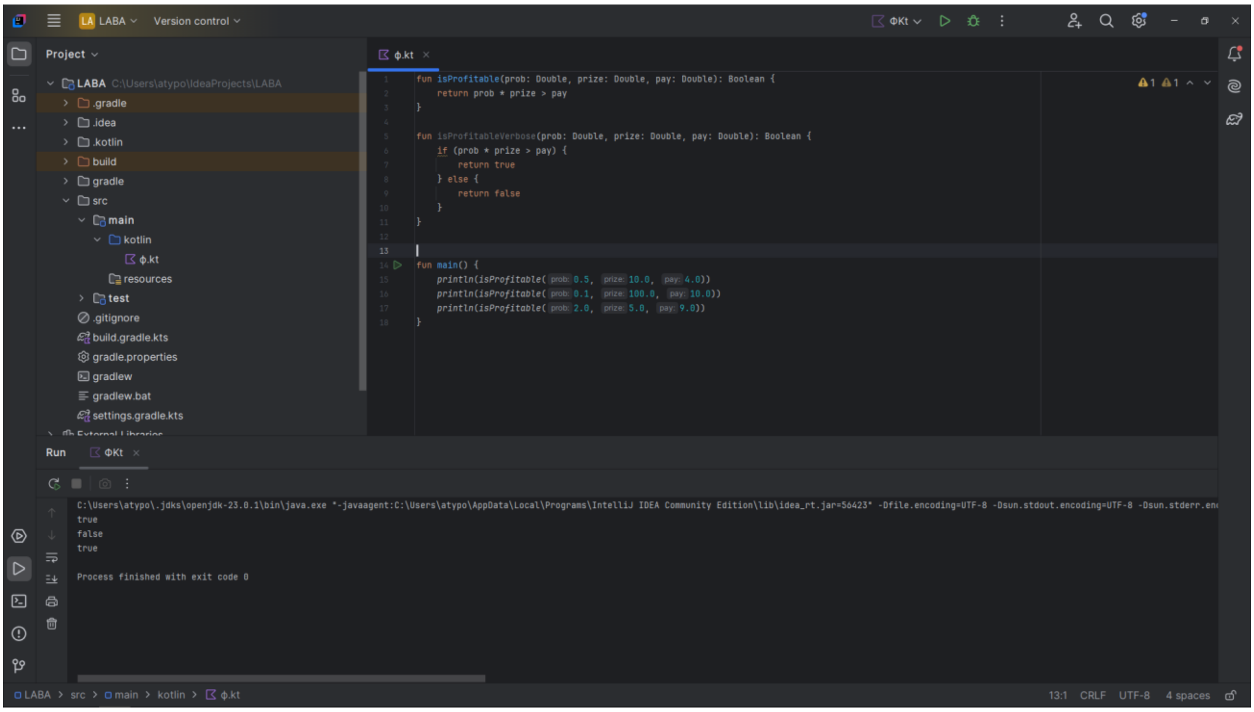
3)



```kotlin
fun combineLists(list1: List<Int>, list2: List<Int>): List<Int> {
    return list1 + list2
}
fun combineListsAlternative(list1: List<Int>, list2: List<Int>): List<Int> {
    return mutableListOf<Int>().apply {
        addAll(list1)
        addAll(list2)
    }
}


fun main() {
    val listA = listOf(1, 2, 3)
    val listB = listOf(4, 5, 6)

    val combined = combineLists(listA, listB)
    println("Объединенный список: $combined")
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56392" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.en
Объединенный список: [1, 2, 3, 4, 5, 6]

Process finished with exit code 0
```

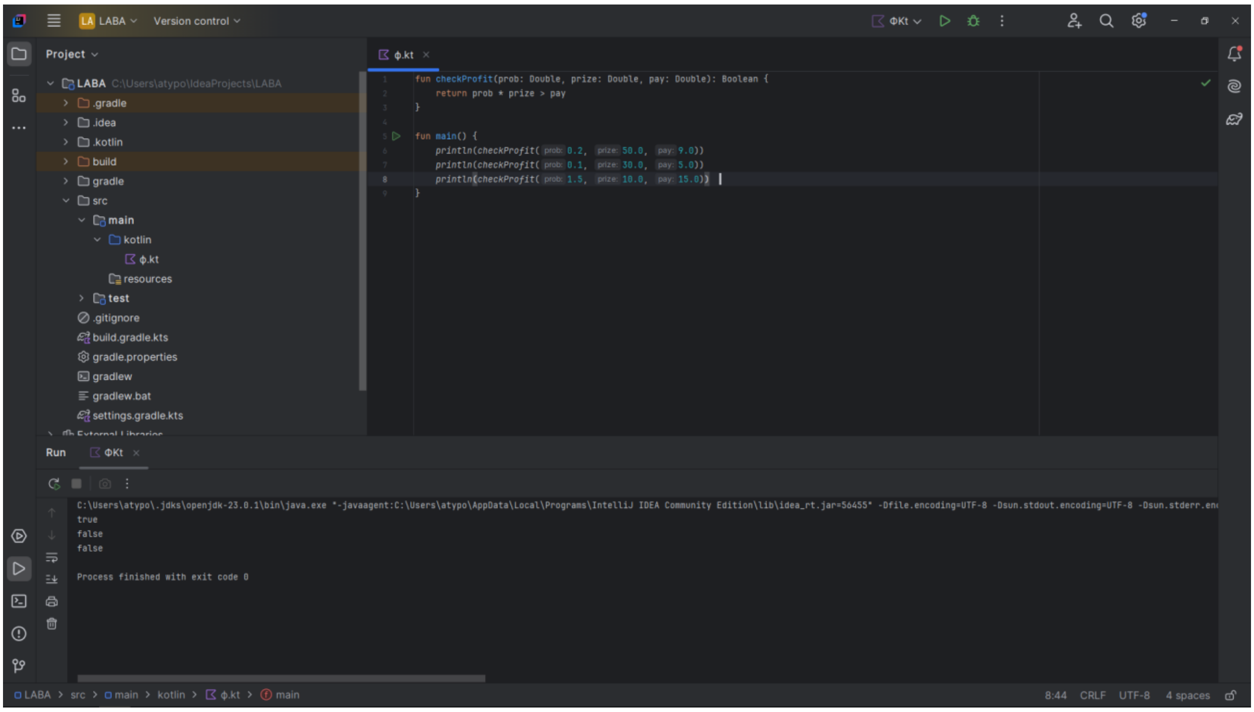
4)



```kotlin
fun isProfitable(prob: Double, prize: Double, pay: Double): Boolean {
    return prob * prize > pay
}

fun isProfitableVerbose(prob: Double, prize: Double, pay: Double): Boolean {
    if (prob * prize > pay) {
        return true
    } else {
        return false
    }
}


fun main() {
    println(isProfitable( prob: 0.5,  prize: 10.0,  pay: 4.0))
    println(isProfitable( prob: 0.1,  prize: 100.0,  pay: 10.0))
    println(isProfitable( prob: 2.0,  prize: 5.0,  pay: 9.0))
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56423" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.en
true
false
true

Process finished with exit code 0
```

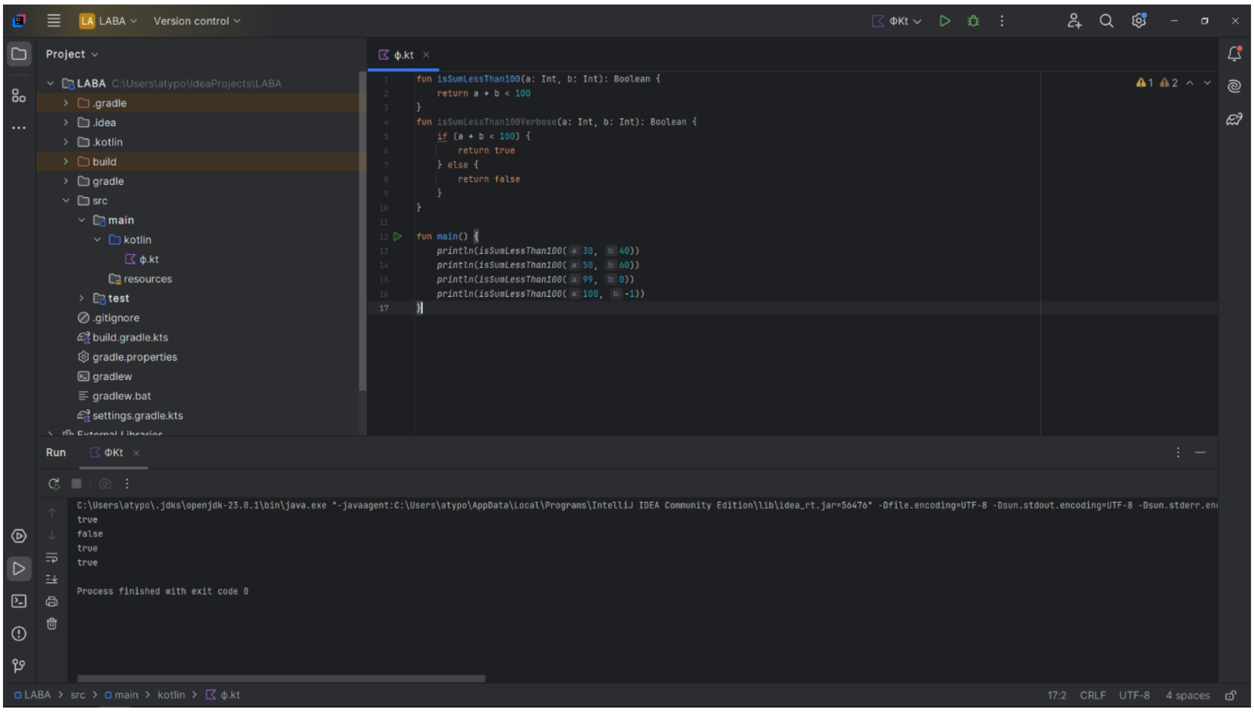
5)



```kotlin
fun checkProfit(prob: Double, prize: Double, pay: Double): Boolean {
    return prob * prize > pay
}

fun main() {
    println(checkProfit(0.2, 50.0, 9.0))
    println(checkProfit(0.1, 30.0, 5.0))
    println(checkProfit(1.5, 10.0, 15.0))
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56455" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.en
true
false
false

Process finished with exit code 0
```

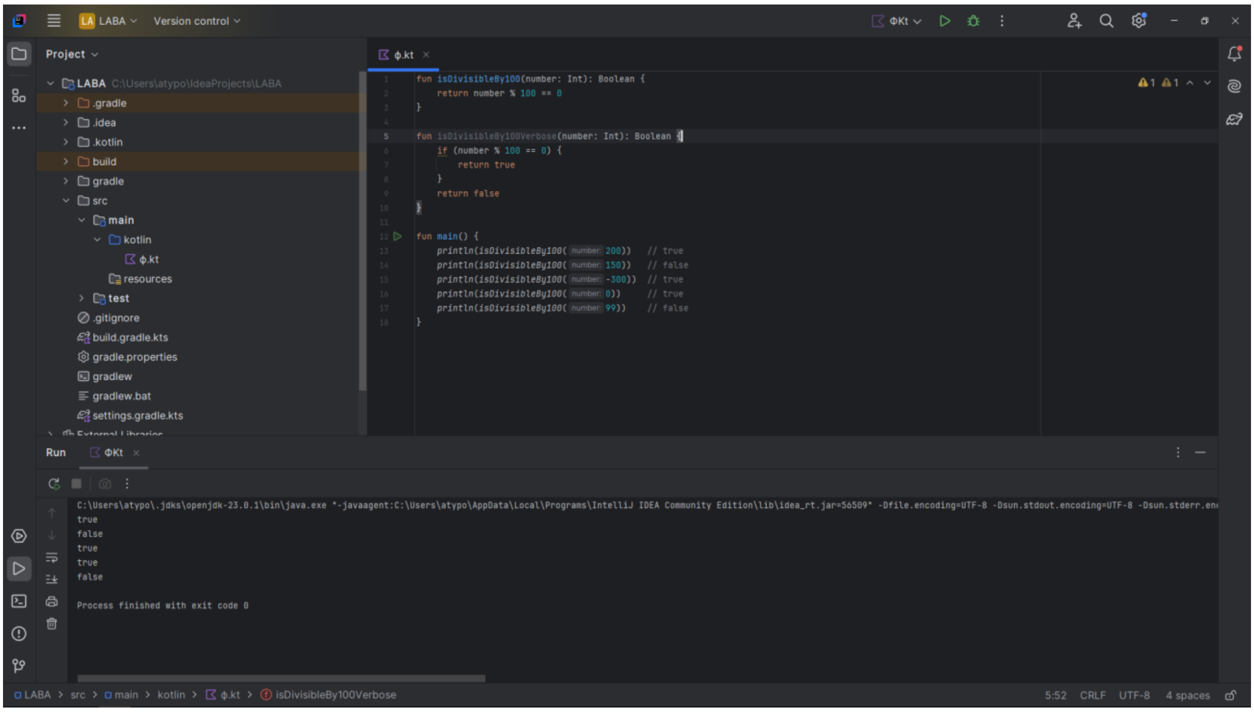
6)



```kotlin
fun isSumLessThan100(a: Int, b: Int): Boolean {
    return a + b < 100
}
fun isSumLessThan100Verbose(a: Int, b: Int): Boolean {
    if (a + b < 100) {
        return true
    } else {
        return false
    }
}

fun main() {
    println(isSumLessThan100(30, 40))
    println(isSumLessThan100(50, 60))
    println(isSumLessThan100(99, 0))
    println(isSumLessThan100(100, -1))
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56476" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.en
true
false
true
true

Process finished with exit code 0
```

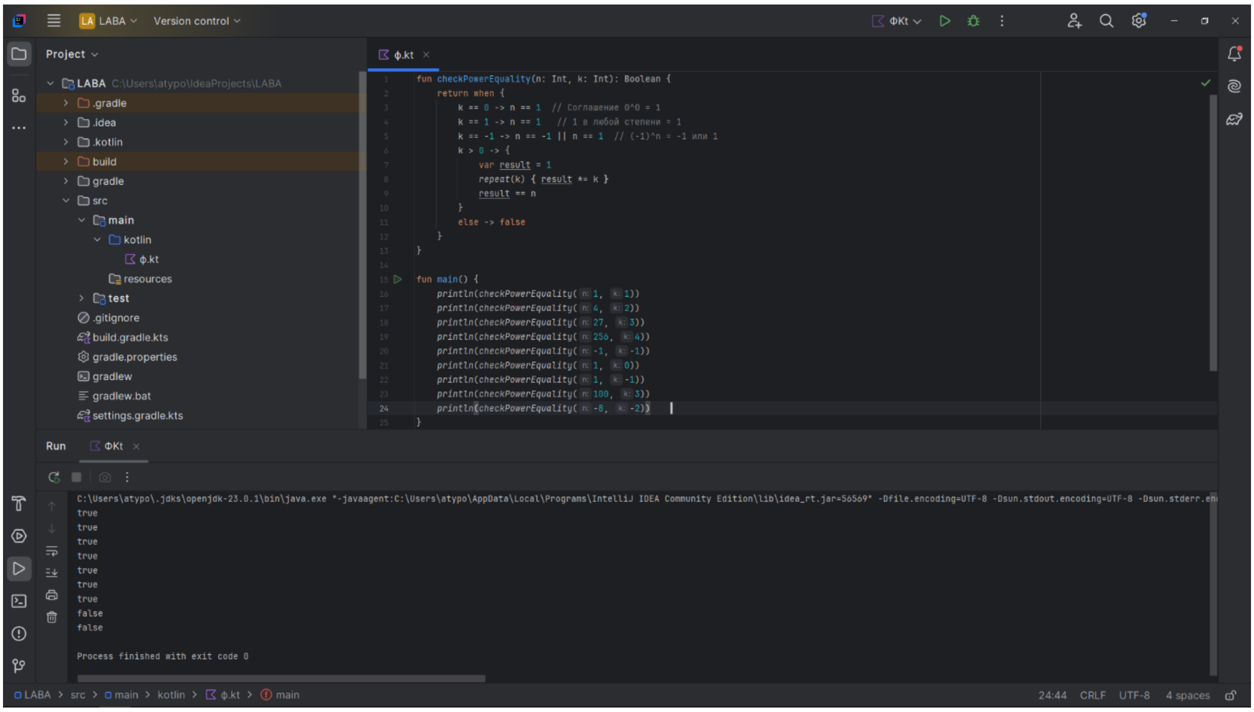
7)



```kotlin
fun isDivisibleBy100(number: Int): Boolean {
    return number % 100 == 0
}

fun isDivisibleBy100Verbose(number: Int): Boolean {
    if (number % 100 == 0) {
        return true
    }
    return false
}

fun main() {
    println(isDivisibleBy100(200))   // true
    println(isDivisibleBy100(150))   // false
    println(isDivisibleBy100(-300))  // true
    println(isDivisibleBy100(0))     // true
    println(isDivisibleBy100(99))    // false
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56509" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.enc
true
false
true
true
false

Process finished with exit code 0
```

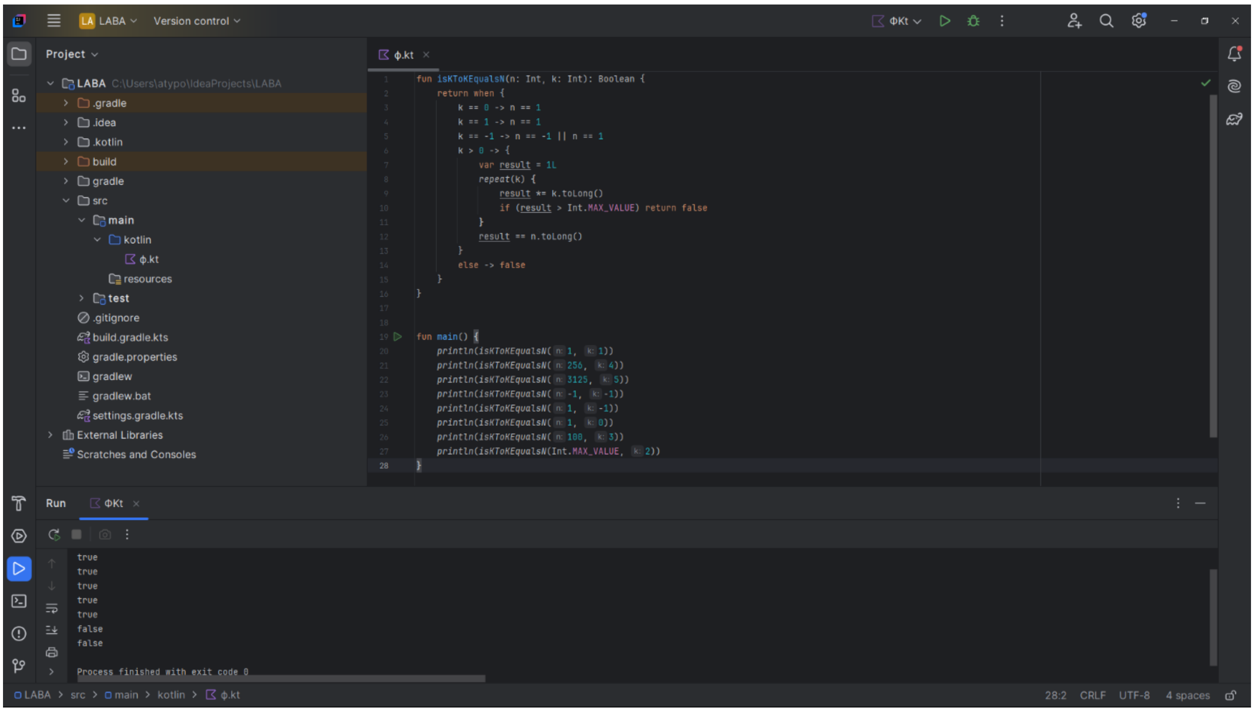
8)



```kotlin
fun checkPowerEquality(n: Int, k: Int): Boolean {
    return when {
        k == 0 -> n == 1  // Соглашение 0^0 = 1
        k == 1 -> n == 1   // 1 в любой степени = 1
        k == -1 -> n == -1 || n == 1  // (-1)^n = -1 или 1
        k > 0 -> {
            var result = 1
            repeat(k) { result *= k }
            result == n
        }
        else -> false
    }
}

fun main() {
    println(checkPowerEquality(1, 1))
    println(checkPowerEquality(4, 2))
    println(checkPowerEquality(27, 3))
    println(checkPowerEquality(256, 4))
    println(checkPowerEquality(-1, -1))
    println(checkPowerEquality(1, 0))
    println(checkPowerEquality(1, -1))
    println(checkPowerEquality(100, 3))
    println(checkPowerEquality(-8, -2))
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56569" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.en
true
true
true
true
true
true
true
false
false

Process finished with exit code 0
```

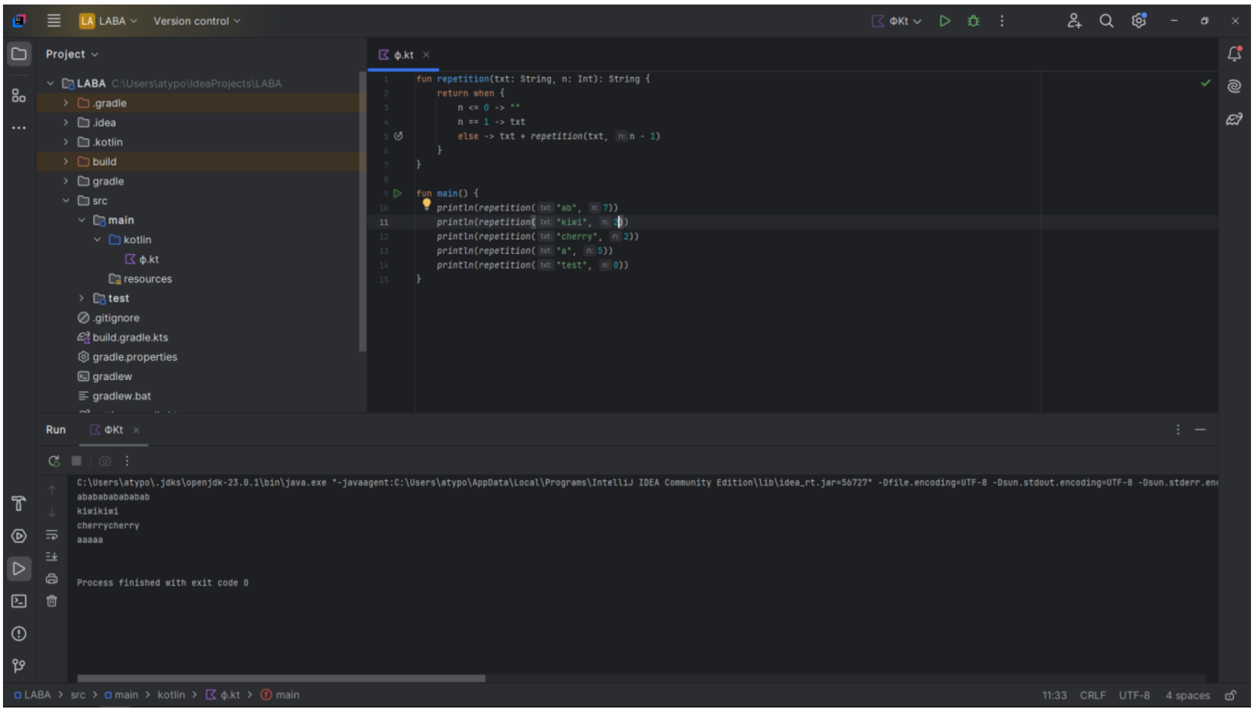
9)



```kotlin
fun isKToKEqualsN(n: Int, k: Int): Boolean {
    return when {
        k == 0 -> n == 1
        k == 1 -> n == 1
        k == -1 -> n == -1 || n == 1
        k > 0 -> {
            var result = 1L
            repeat(k) {
                result *= k.toLong()
                if (result > Int.MAX_VALUE) return false
            }
            result == n.toLong()
        }
        else -> false
    }
}


fun main() {
    println(isKToKEqualsN(1, 1))
    println(isKToKEqualsN(256, 4))
    println(isKToKEqualsN(3125, 5))
    println(isKToKEqualsN(-1, -1))
    println(isKToKEqualsN(1, -1))
    println(isKToKEqualsN(1, 0))
    println(isKToKEqualsN(100, 3))
    println(isKToKEqualsN(Int.MAX_VALUE, 2))
}
```

```
true
true
true
true
true
false
false

Process finished with exit code 0
```

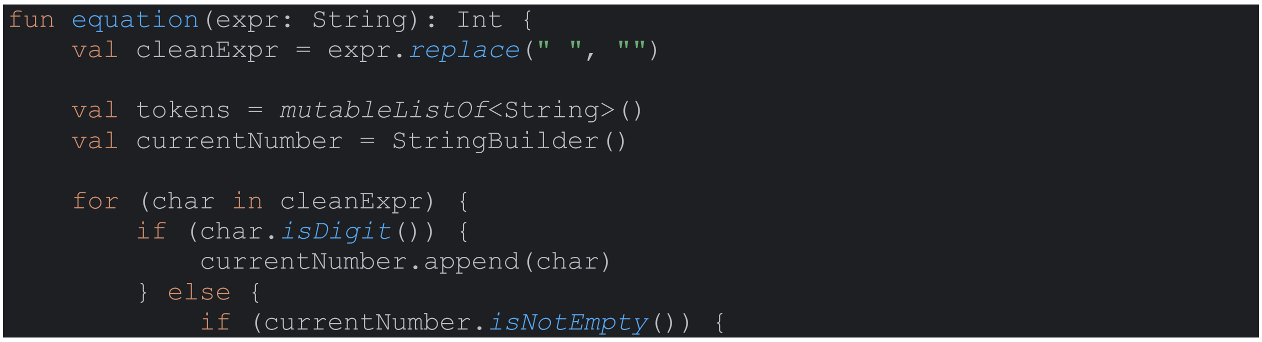
10)



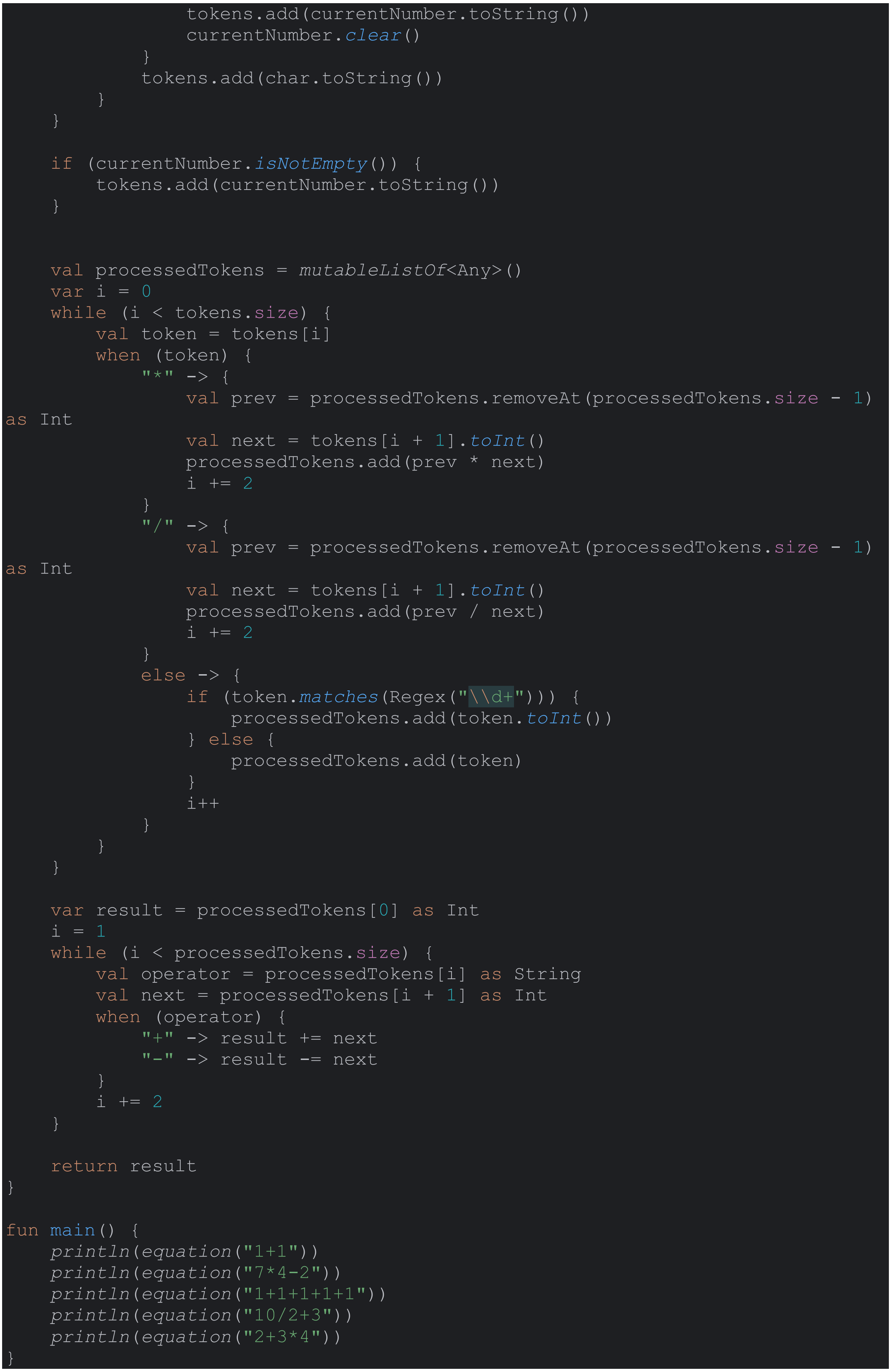
```kotlin
fun repetition(txt: String, n: Int): String {
    return when {
        n <= 0 -> ""
        n == 1 -> txt
        else -> txt + repetition(txt, n - 1)
    }
}

fun main() {
    println(repetition("ab", 7))
    println(repetition("kiwi", 2))
    println(repetition("cherry", 2))
    println(repetition("a", 5))
    println(repetition("test", 0))
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56727" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.en
abababababab
kiwikiwi
cherrycherry
aaaaa


Process finished with exit code 0
```

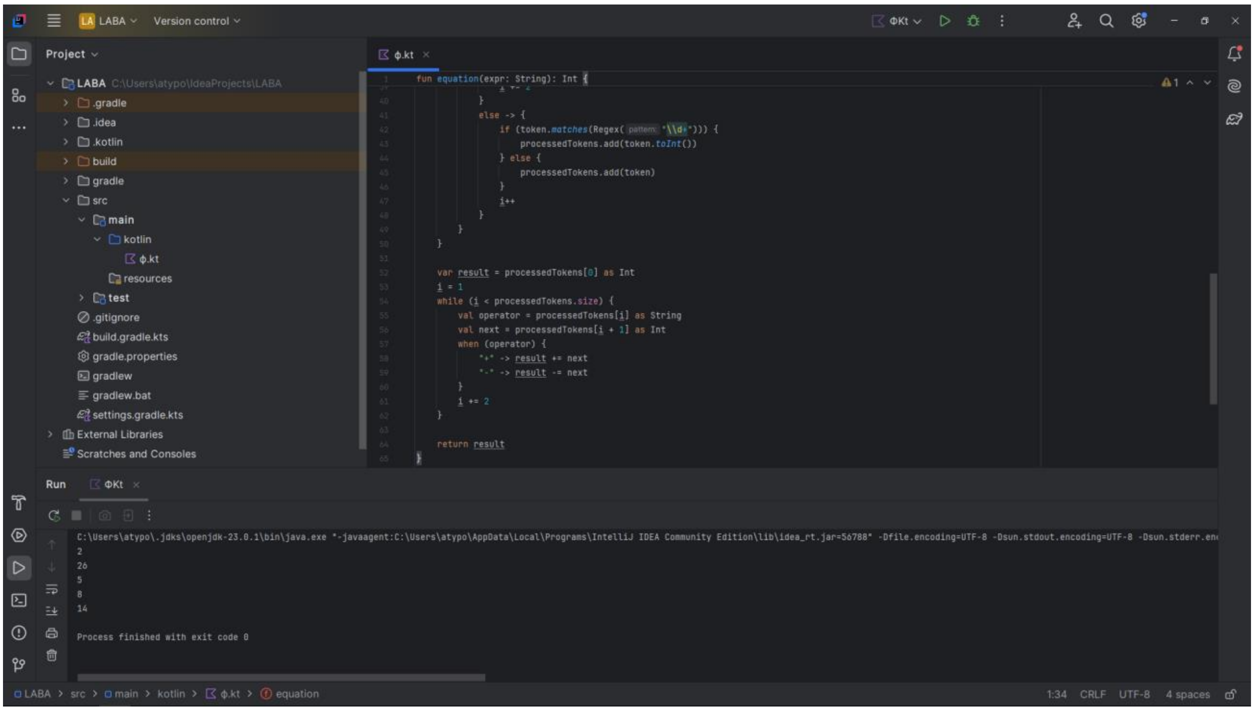
11)



```kotlin
fun equation(expr: String): Int {
    val cleanExpr = expr.replace(" ", "")

    val tokens = mutableListOf<String>()
    val currentNumber = StringBuilder()

    for (char in cleanExpr) {
        if (char.isDigit()) {
            currentNumber.append(char)
        } else {
            if (currentNumber.isNotEmpty()) {
```



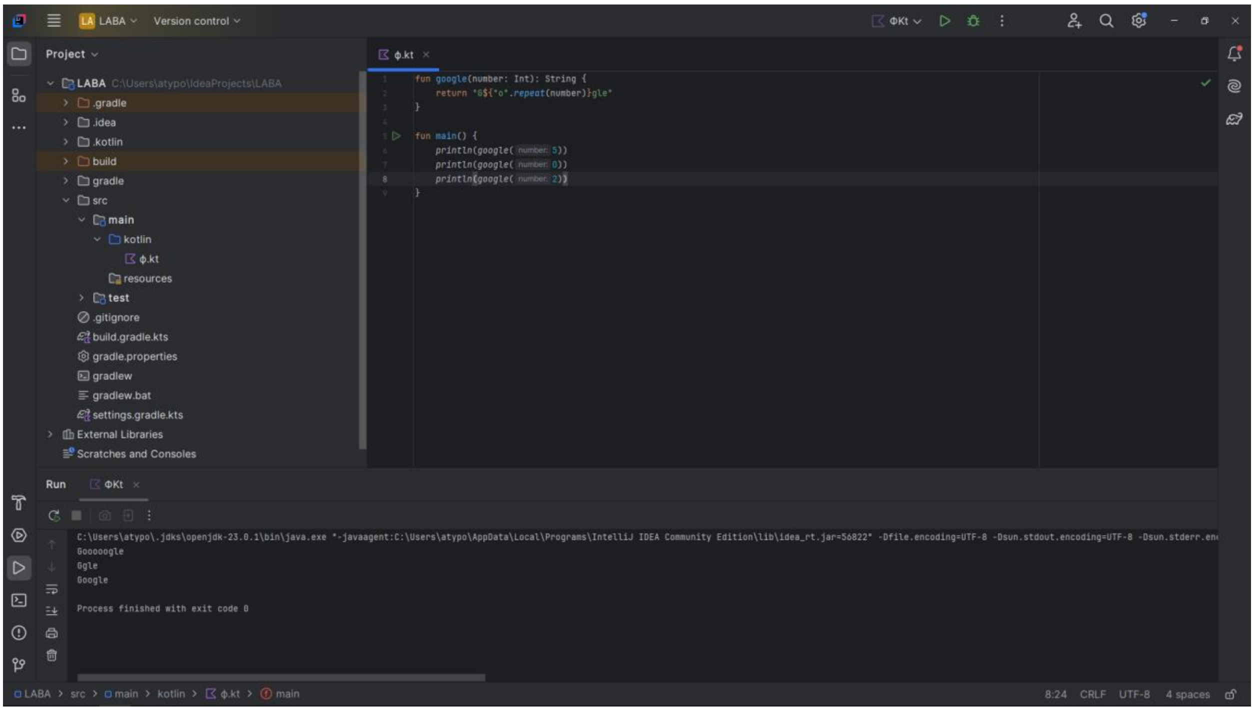
```
                tokens.add(currentNumber.toString())
                currentNumber.clear()
            }
            tokens.add(char.toString())
        }
    }

    if (currentNumber.isNotEmpty()) {
        tokens.add(currentNumber.toString())
    }


    val processedTokens = mutableListOf<Any>()
    var i = 0
    while (i < tokens.size) {
        val token = tokens[i]
        when (token) {
            "*" -> {
                val prev = processedTokens.removeAt(processedTokens.size - 1)
as Int
                val next = tokens[i + 1].toInt()
                processedTokens.add(prev * next)
                i += 2
            }
            "/" -> {
                val prev = processedTokens.removeAt(processedTokens.size - 1)
as Int
                val next = tokens[i + 1].toInt()
                processedTokens.add(prev / next)
                i += 2
            }
            else -> {
                if (token.matches(Regex("\\d+"))) {
                    processedTokens.add(token.toInt())
                } else {
                    processedTokens.add(token)
                }
                i++
            }
        }
    }

    var result = processedTokens[0] as Int
    i = 1
    while (i < processedTokens.size) {
        val operator = processedTokens[i] as String
        val next = processedTokens[i + 1] as Int
        when (operator) {
            "+" -> result += next
            "-" -> result -= next
        }
        i += 2
    }

    return result
}

fun main() {
    println(equation("1+1"))
    println(equation("7*4-2"))
    println(equation("1+1+1+1+1"))
    println(equation("10/2+3"))
    println(equation("2+3*4"))
}
```



```
fun equation(expr: String): Int {
            }
            else -> {
                if (token.matches(Regex("\\d+"))) {
                    processedTokens.add(token.toInt())
                } else {
                    processedTokens.add(token)
                }
                i++
            }
        }
    }

    var result = processedTokens[0] as Int
    i = 1
    while (i < processedTokens.size) {
        val operator = processedTokens[i] as String
        val next = processedTokens[i + 1] as Int
        when (operator) {
            "+" -> result += next
            "-" -> result -= next
        }
        i += 2
    }

    return result
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56788" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.enc
2
26
5
8
14

Process finished with exit code 0
```

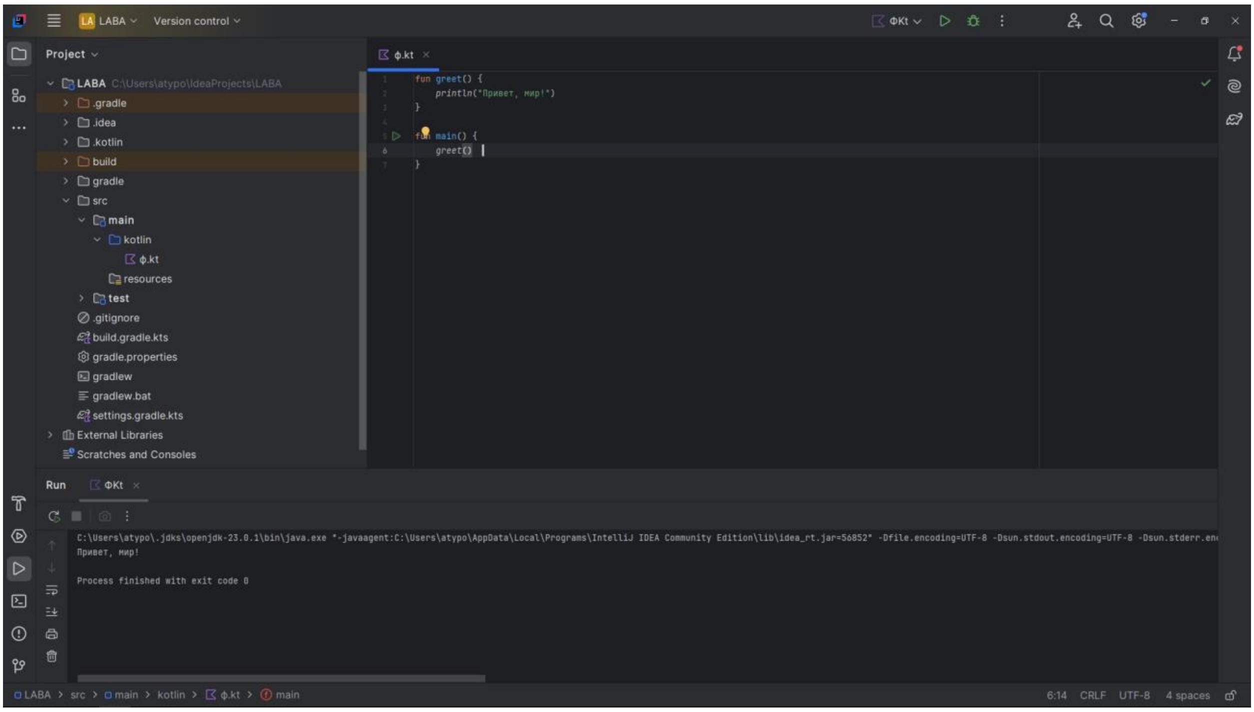
12)



```
fun google(number: Int): String {
    return "G${"o".repeat(number)}gle"
}

fun main() {
    println(google(5))
    println(google(0))
    println(google(2))
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56822" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.enc
Goooooogle
Ggle
Google

Process finished with exit code 0
```

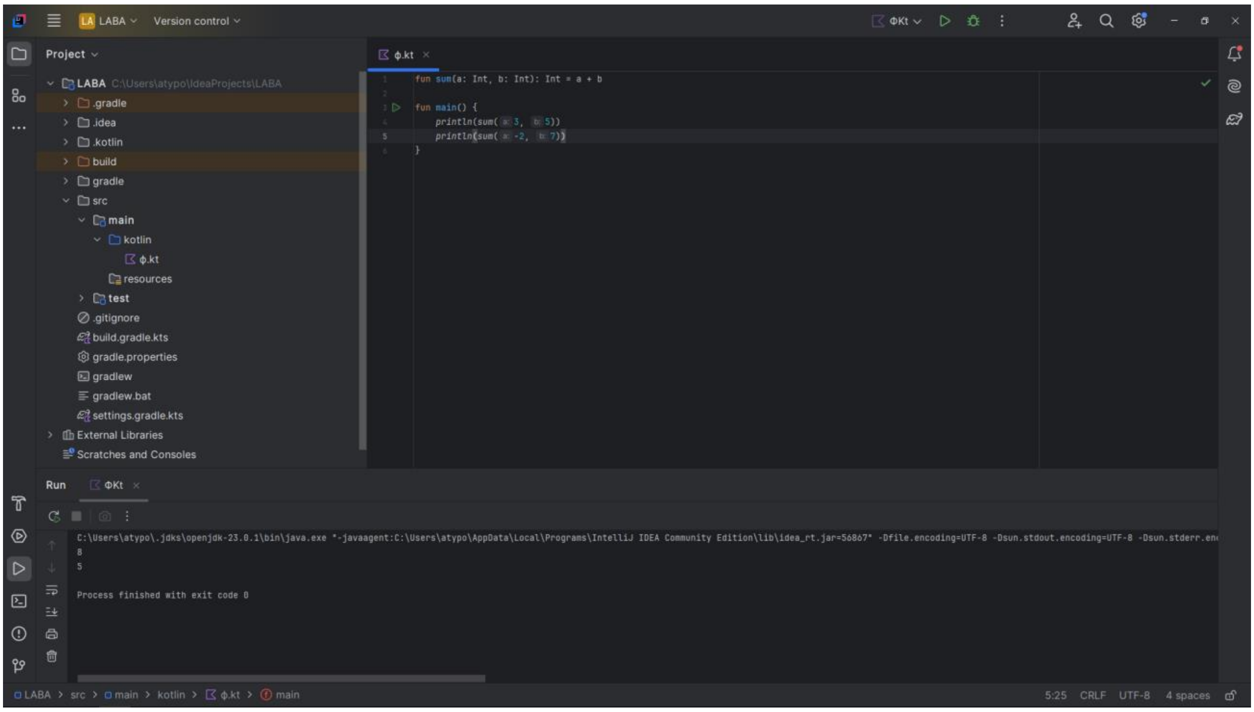
13)



```kotlin
fun greet() {
    println("Привет, мир!")
}

fun main() {
    greet()
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56852" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.enc
Привет, мир!

Process finished with exit code 0
```

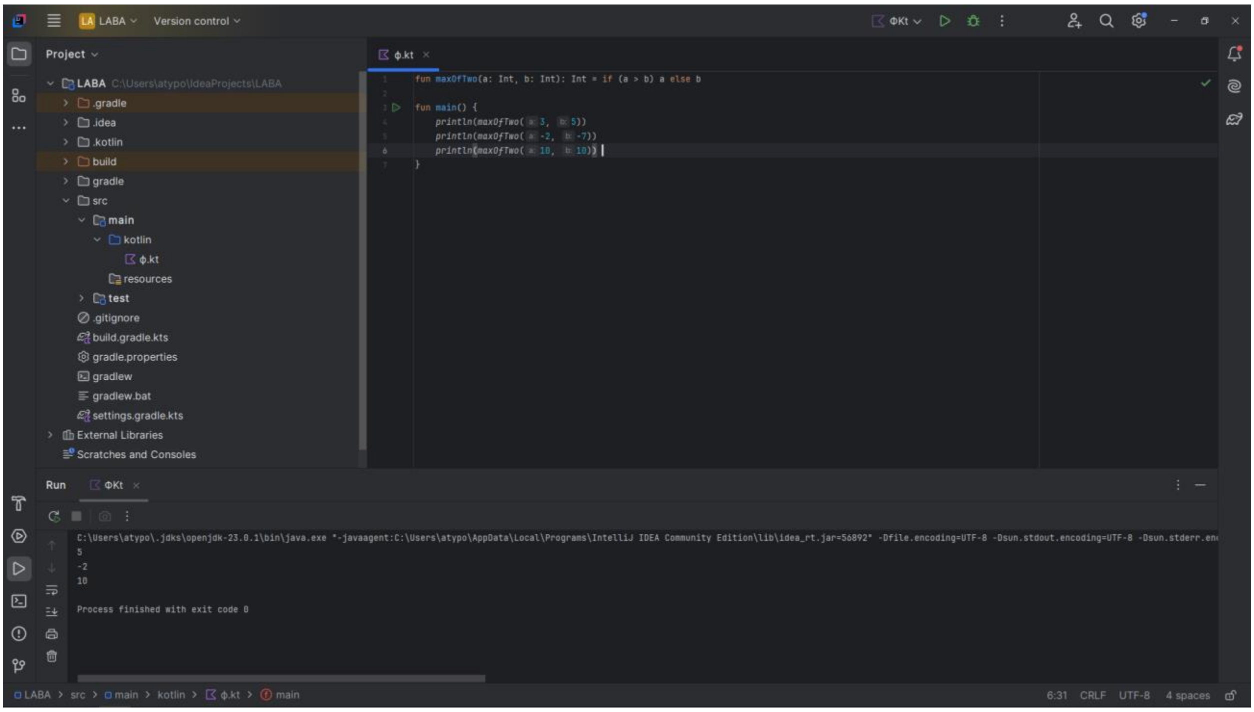
14)



```kotlin
fun sum(a: Int, b: Int): Int = a + b

fun main() {
    println(sum( a: 3,  b: 5))
    println(sum( a: -2,  b: 7))
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56867" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.enc
8
5

Process finished with exit code 0
```

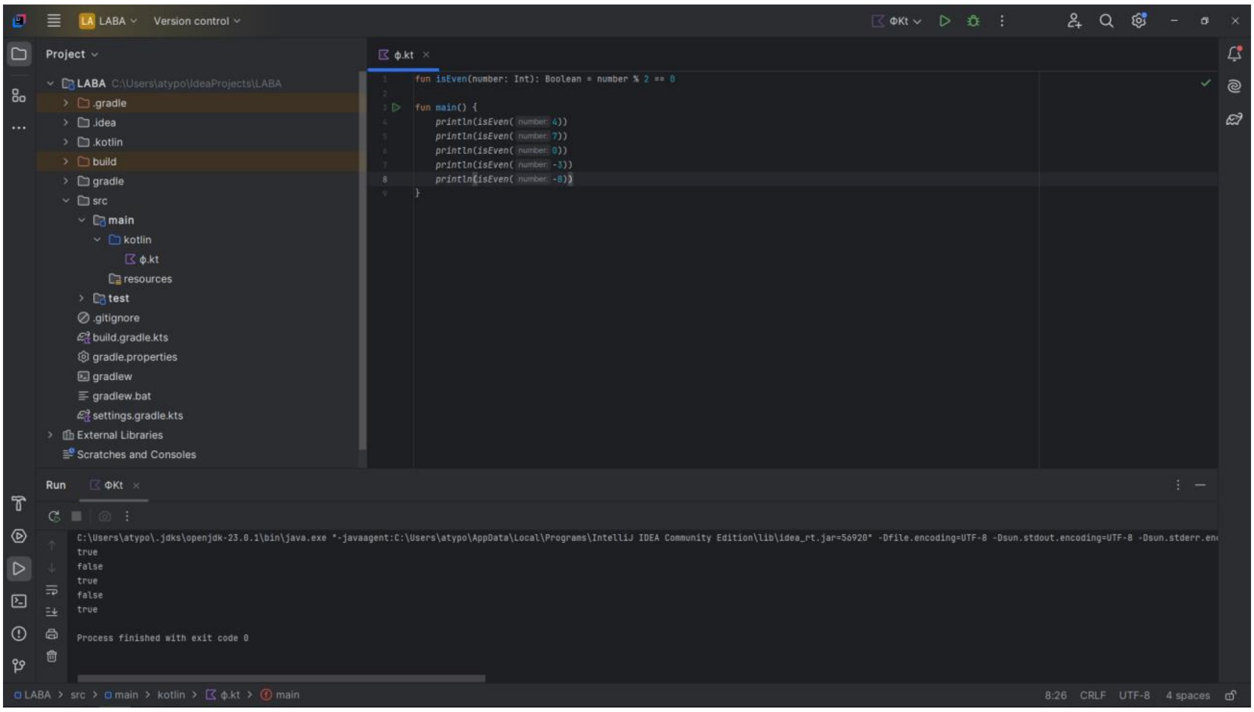
15)



```kotlin
fun maxOfTwo(a: Int, b: Int): Int = if (a > b) a else b

fun main() {
    println(maxOfTwo(3, 5))
    println(maxOfTwo(-2, -7))
    println(maxOfTwo(10, 10))
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56892" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.en
5
-2
10

Process finished with exit code 0
```

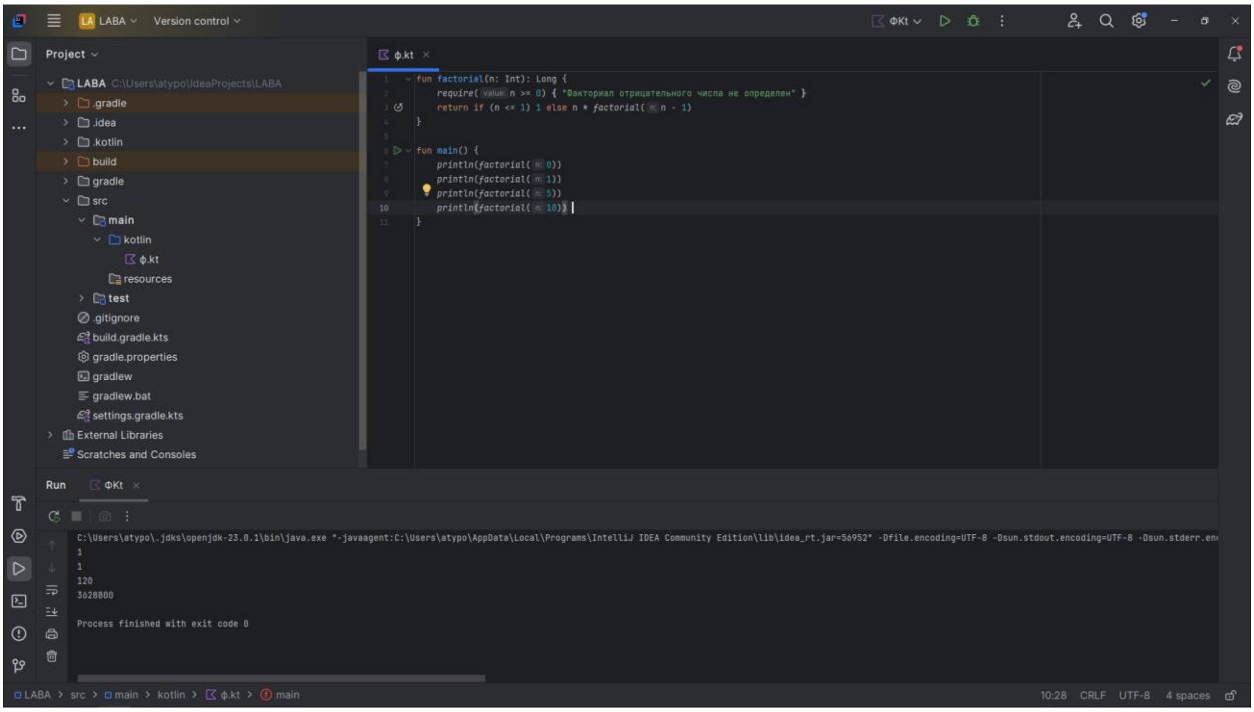
16)



```kotlin
fun isEven(number: Int): Boolean = number % 2 == 0

fun main() {
    println(isEven(4))
    println(isEven(7))
    println(isEven(0))
    println(isEven(-3))
    println(isEven(-8))
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56920" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.en
true
false
true
false
true

Process finished with exit code 0
```

17)

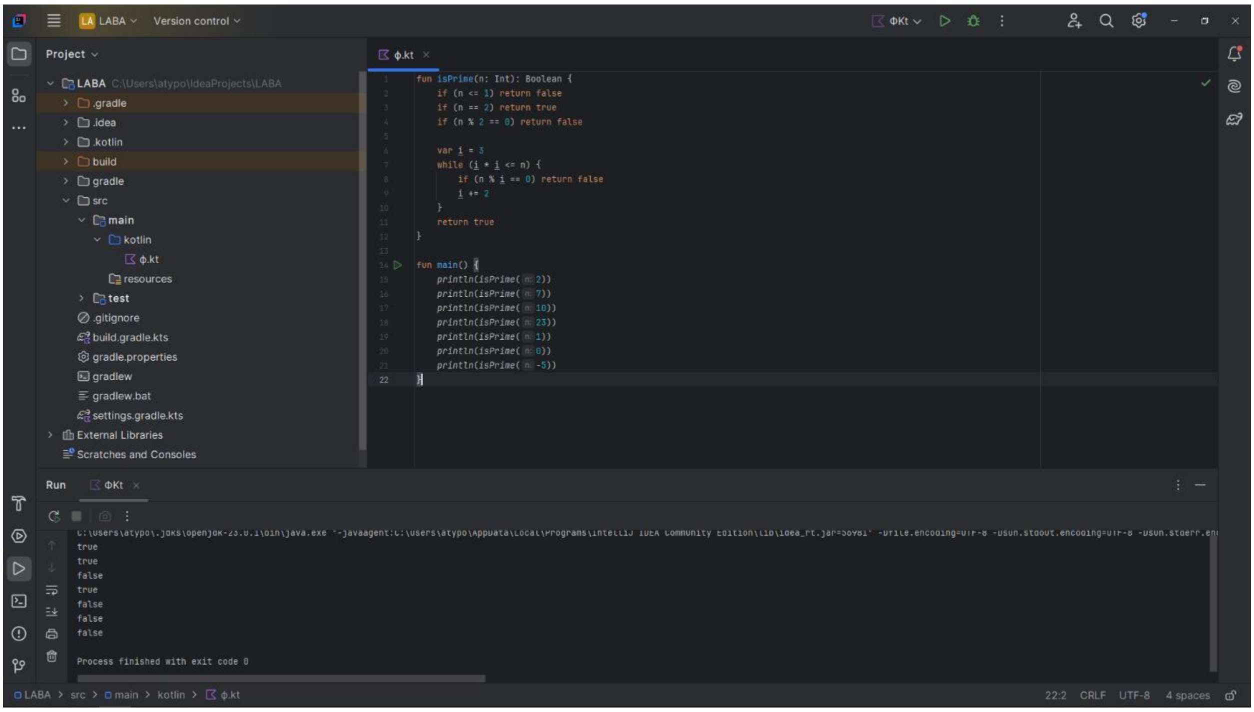


ϕ.kt

```kotlin
fun factorial(n: Int): Long {
    require(n >= 0) { "Факториал отрицательного числа не определен" }
    return if (n <= 1) 1 else n * factorial(n - 1)
}

fun main() {
    println(factorial(0))
    println(factorial(1))
    println(factorial(5))
    println(factorial(10))
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56952" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.enc
1
1
120
3628800

Process finished with exit code 0
```

18)

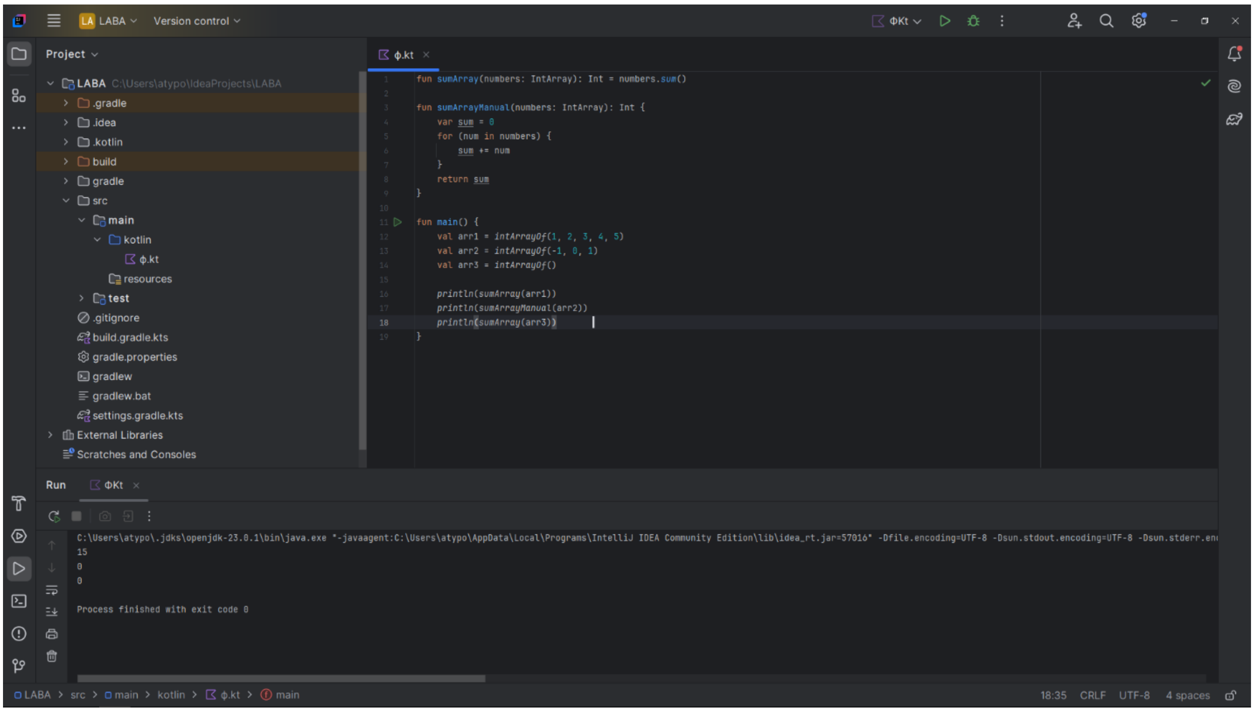


ϕ.kt

```kotlin
fun isPrime(n: Int): Boolean {
    if (n <= 1) return false
    if (n == 2) return true
    if (n % 2 == 0) return false

    var i = 3
    while (i * i <= n) {
        if (n % i == 0) return false
        i += 2
    }
    return true
}

fun main() {
    println(isPrime(2))
    println(isPrime(7))
    println(isPrime(10))
    println(isPrime(23))
    println(isPrime(1))
    println(isPrime(0))
    println(isPrime(-5))
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=56981" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.en
true
true
false
true
false
false
false

Process finished with exit code 0
```

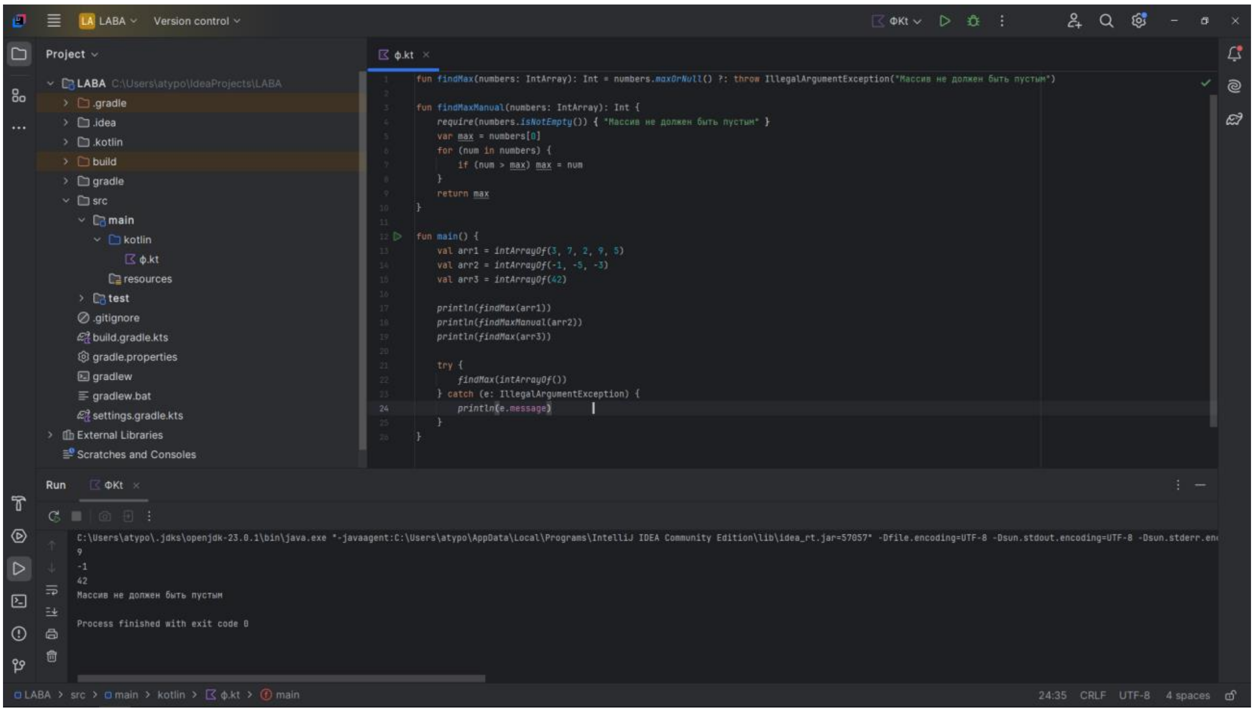
19)



```kotlin
fun sumArray(numbers: IntArray): Int = numbers.sum()

fun sumArrayManual(numbers: IntArray): Int {
    var sum = 0
    for (num in numbers) {
        sum += num
    }
    return sum
}

fun main() {
    val arr1 = intArrayOf(1, 2, 3, 4, 5)
    val arr2 = intArrayOf(-1, 0, 1)
    val arr3 = intArrayOf()

    println(sumArray(arr1))
    println(sumArrayManual(arr2))
    println(sumArray(arr3))
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=57016" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.enc
15
0
0

Process finished with exit code 0
```

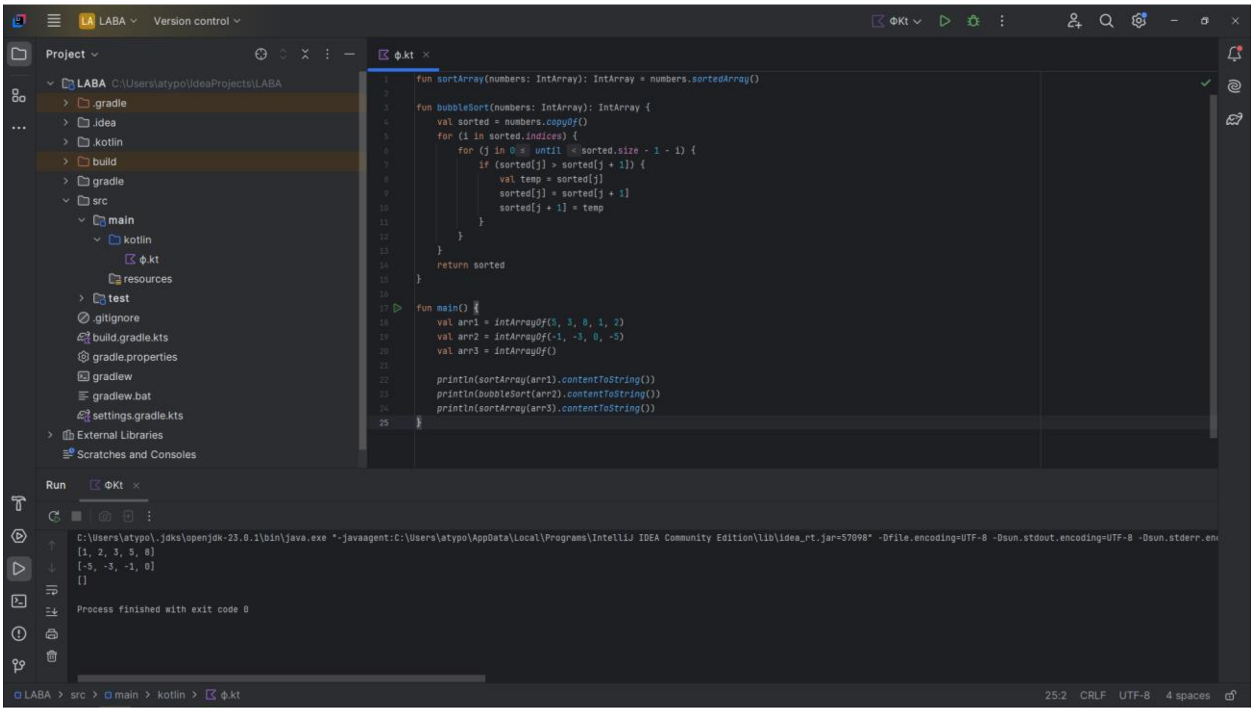
20)



```kotlin
fun findMax(numbers: IntArray): Int = numbers.maxOrNull() ?: throw IllegalArgumentException("Массив не должен быть пустым")

fun findMaxManual(numbers: IntArray): Int {
    require(numbers.isNotEmpty()) { "Массив не должен быть пустым" }
    var max = numbers[0]
    for (num in numbers) {
        if (num > max) max = num
    }
    return max
}

fun main() {
    val arr1 = intArrayOf(3, 7, 2, 9, 5)
    val arr2 = intArrayOf(-1, -5, -3)
    val arr3 = intArrayOf(42)

    println(findMax(arr1))
    println(findMaxManual(arr2))
    println(findMax(arr3))

    try {
        findMax(intArrayOf())
    } catch (e: IllegalArgumentException) {
        println(e.message)
    }
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=57057" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.enc
9
-1
42
Массив не должен быть пустым

Process finished with exit code 0
```

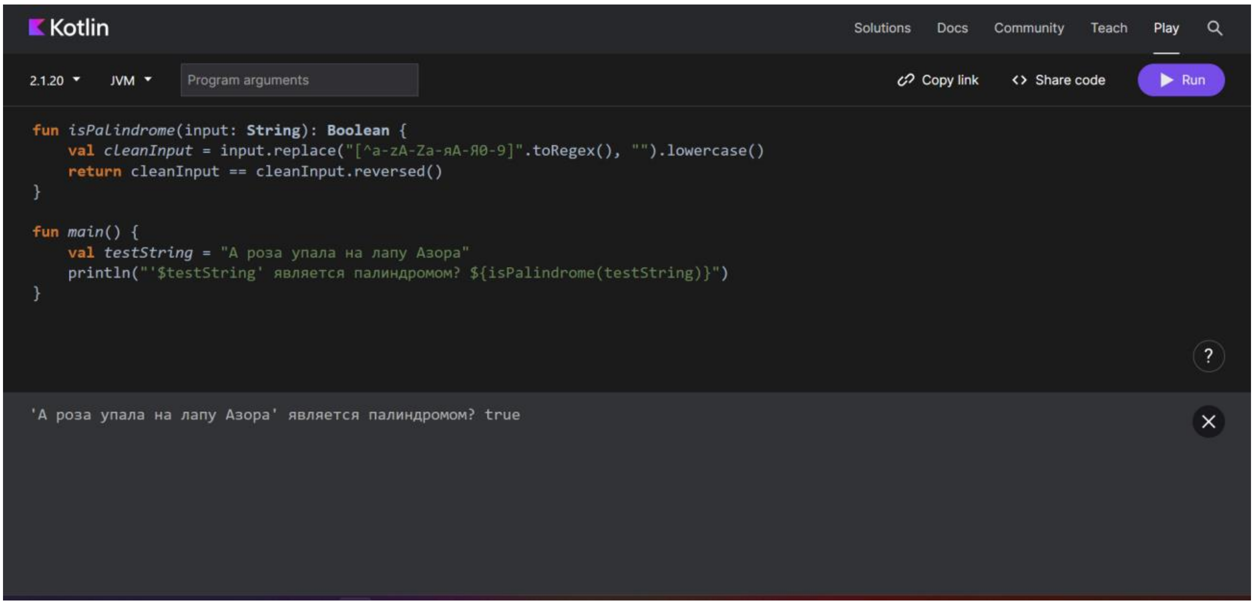
21)



```kotlin
fun sortArray(numbers: IntArray): IntArray = numbers.sortedArray()

fun bubbleSort(numbers: IntArray): IntArray {
    val sorted = numbers.copyOf()
    for (i in sorted.indices) {
        for (j in 0 ≤ until < sorted.size - 1 - i) {
            if (sorted[j] > sorted[j + 1]) {
                val temp = sorted[j]
                sorted[j] = sorted[j + 1]
                sorted[j + 1] = temp
            }
        }
    }
    return sorted
}

fun main() {
    val arr1 = intArrayOf(5, 3, 8, 1, 2)
    val arr2 = intArrayOf(-1, -3, 0, -5)
    val arr3 = intArrayOf()

    println(sortArray(arr1).contentToString())
    println(bubbleSort(arr2).contentToString())
    println(sortArray(arr3).contentToString())
}
```

```
C:\Users\atypo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=57098" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.enc
[1, 2, 3, 5, 8]
[-5, -3, -1, 0]
[]

Process finished with exit code 0
```

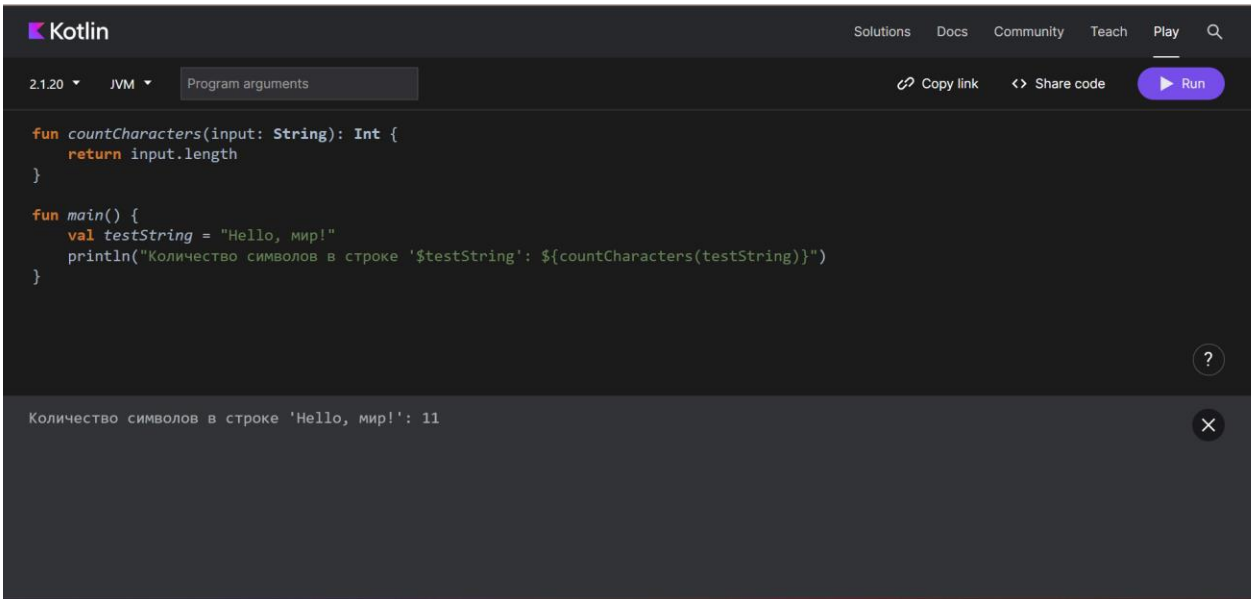
22)



```kotlin
fun isPalindrome(input: String): Boolean {
    val cleanInput = input.replace("[^a-zA-Za-яА-Я0-9]".toRegex(), "").lowercase()
    return cleanInput == cleanInput.reversed()
}

fun main() {
    val testString = "А роза упала на лапу Азора"
    println("'$testString' является палиндромом? ${isPalindrome(testString)}")
}
```

```
'А роза упала на лапу Азора' является палиндромом? true
```

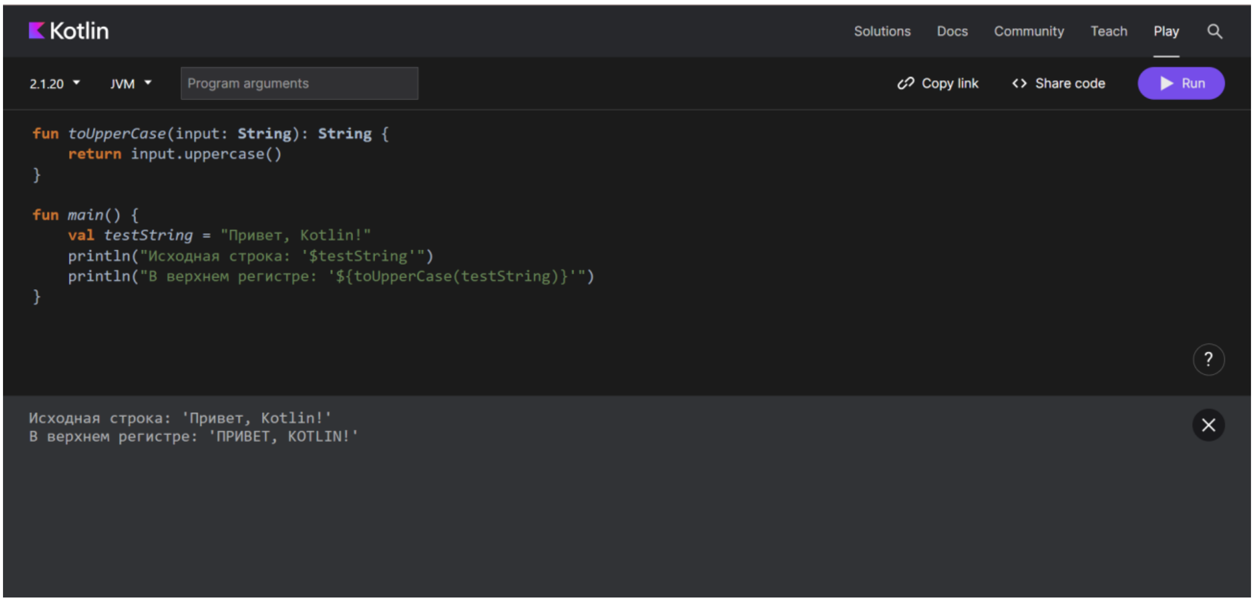
23)



```kotlin
fun countCharacters(input: String): Int {
    return input.length
}

fun main() {
    val testString = "Hello, мир!"
    println("Количество символов в строке '$testString': ${countCharacters(testString)}")
}
```

```
Количество символов в строке 'Hello, мир!': 11
```

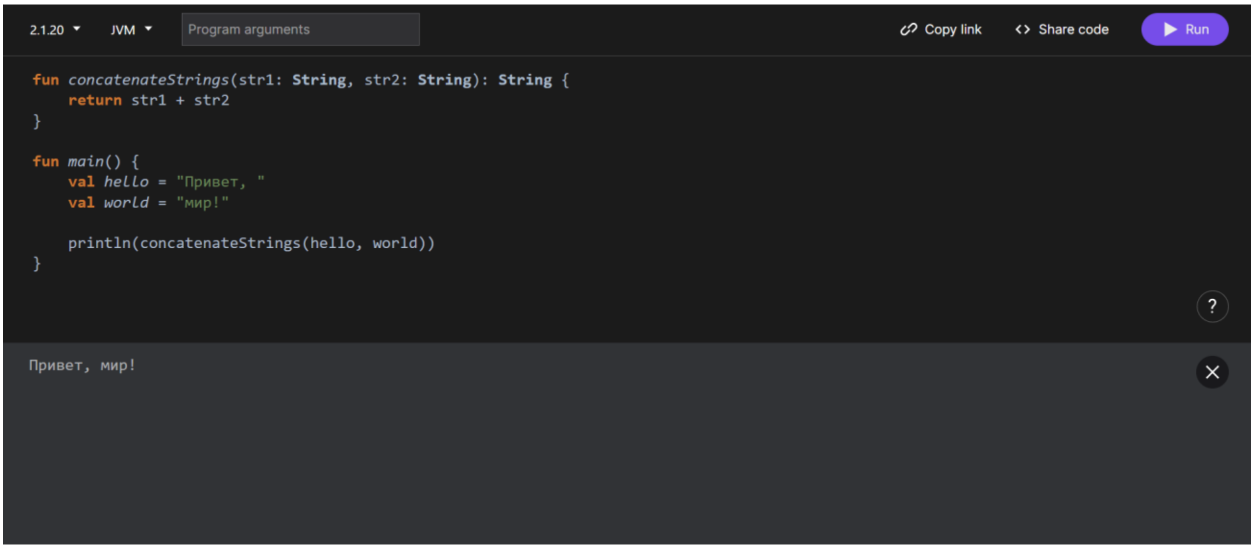
24)



```kotlin
fun toUpperCase(input: String): String {
    return input.uppercase()
}

fun main() {
    val testString = "Привет, Kotlin!"
    println("Исходная строка: '$testString'")
    println("В верхнем регистре: '${toUpperCase(testString)}'")
}
```

```
Исходная строка: 'Привет, Kotlin!'
В верхнем регистре: 'ПРИВЕТ, KOTLIN!'
```

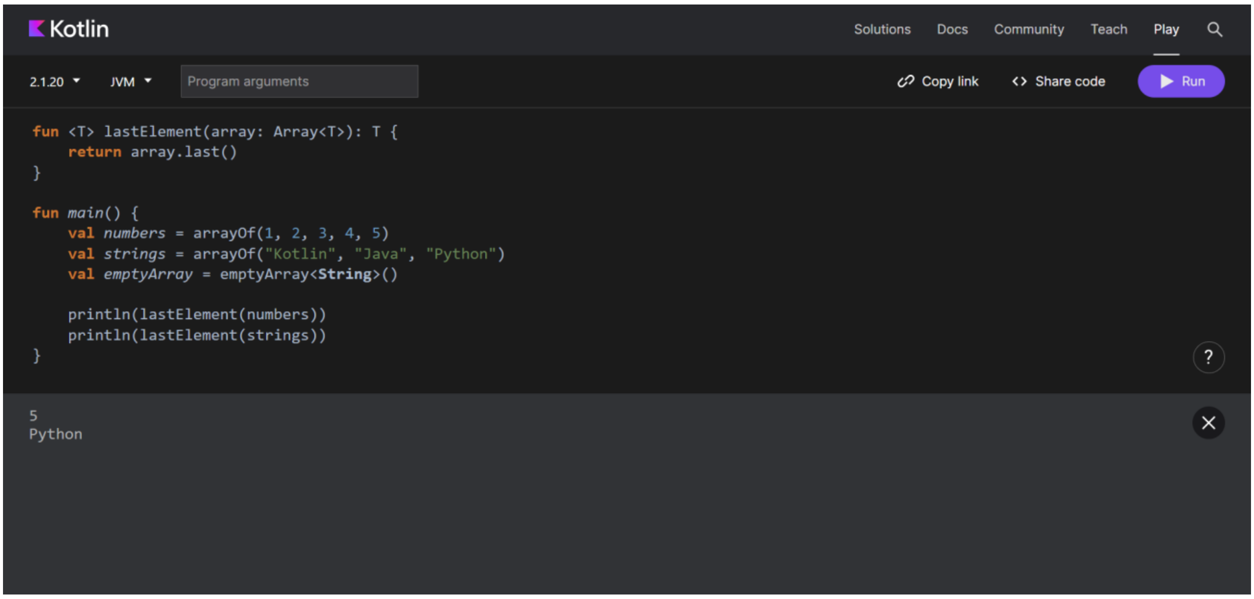
25)



```kotlin
fun concatenateStrings(str1: String, str2: String): String {
    return str1 + str2
}

fun main() {
    val hello = "Привет, "
    val world = "мир!"

    println(concatenateStrings(hello, world))
}
```

```
Привет, мир!
```

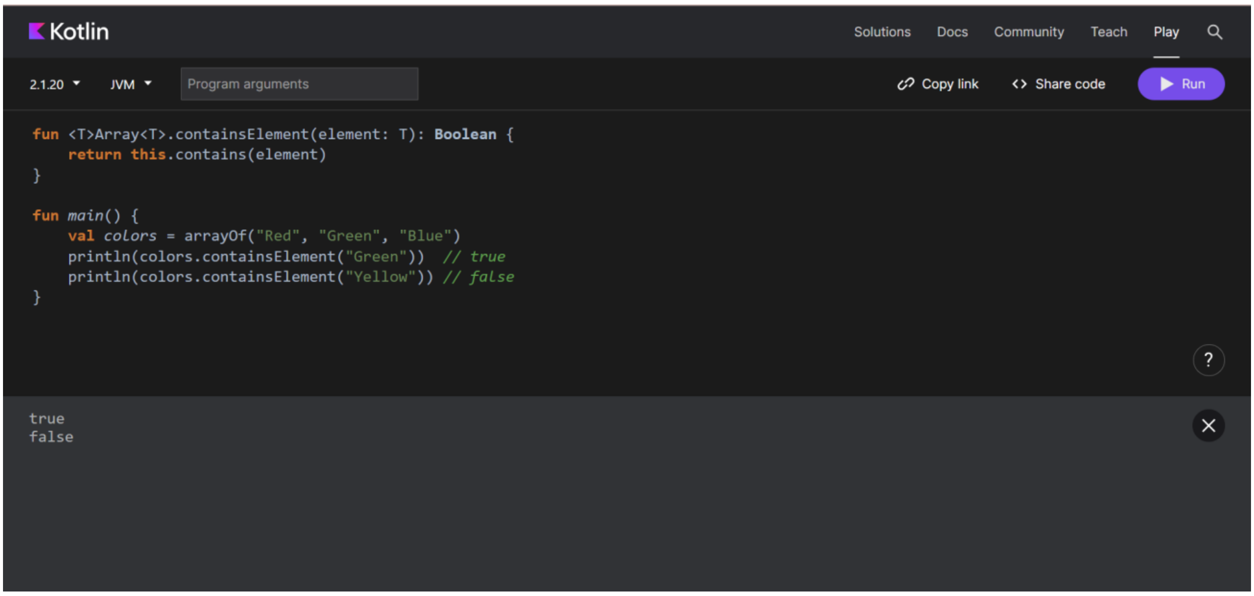
26)



```
fun <T> lastElement(array: Array<T>): T {
    return array.last()
}

fun main() {
    val numbers = arrayOf(1, 2, 3, 4, 5)
    val strings = arrayOf("Kotlin", "Java", "Python")
    val emptyArray = emptyArray<String>()

    println(lastElement(numbers))
    println(lastElement(strings))
}
```

```
5
Python
```

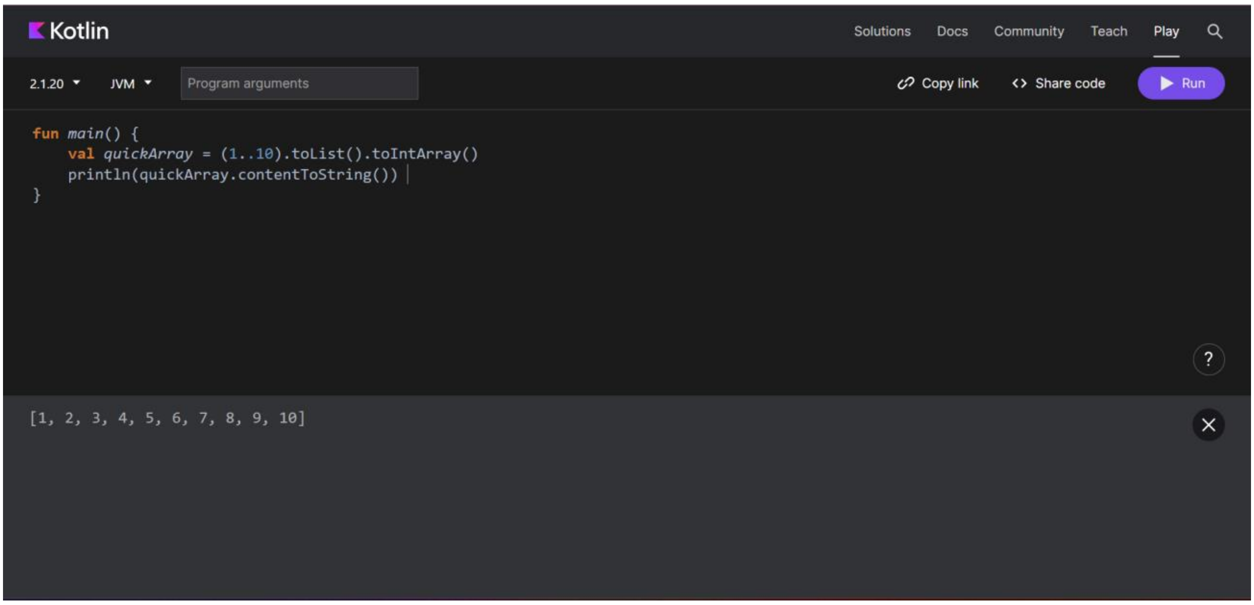
27)



```
fun <T>Array<T>.containsElement(element: T): Boolean {
    return this.contains(element)
}

fun main() {
    val colors = arrayOf("Red", "Green", "Blue")
    println(colors.containsElement("Green"))  // true
    println(colors.containsElement("Yellow")) // false
}
```

```
true
false
```

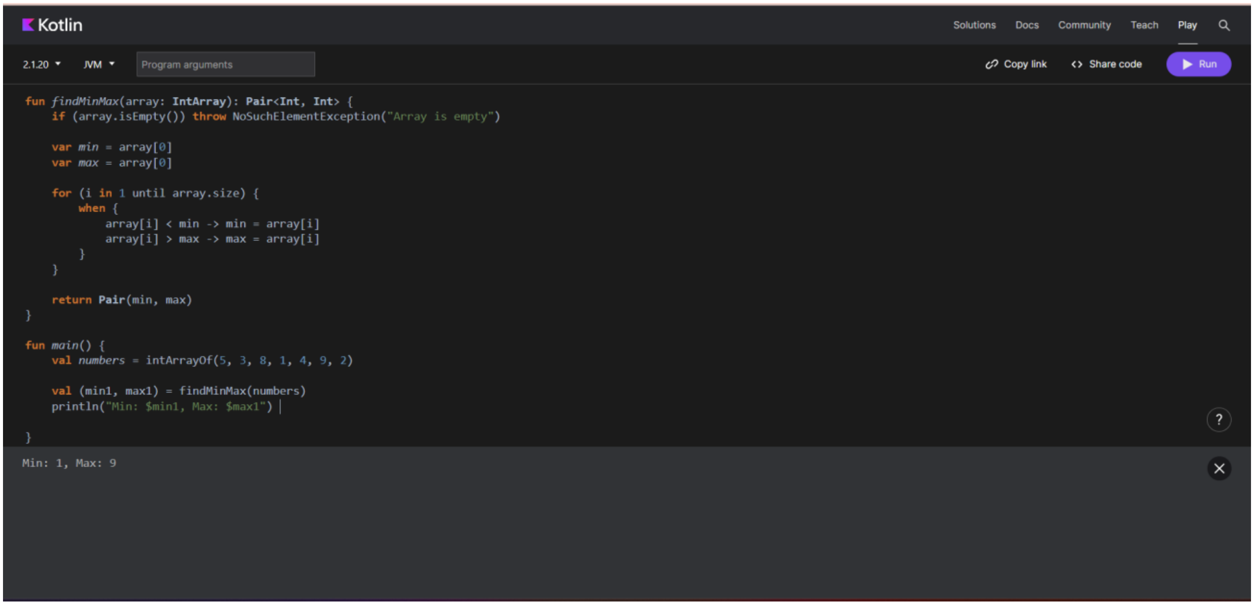
28)



```
fun main() {
    val quickArray = (1..10).toList().toIntArray()
    println(quickArray.contentToString())
}
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

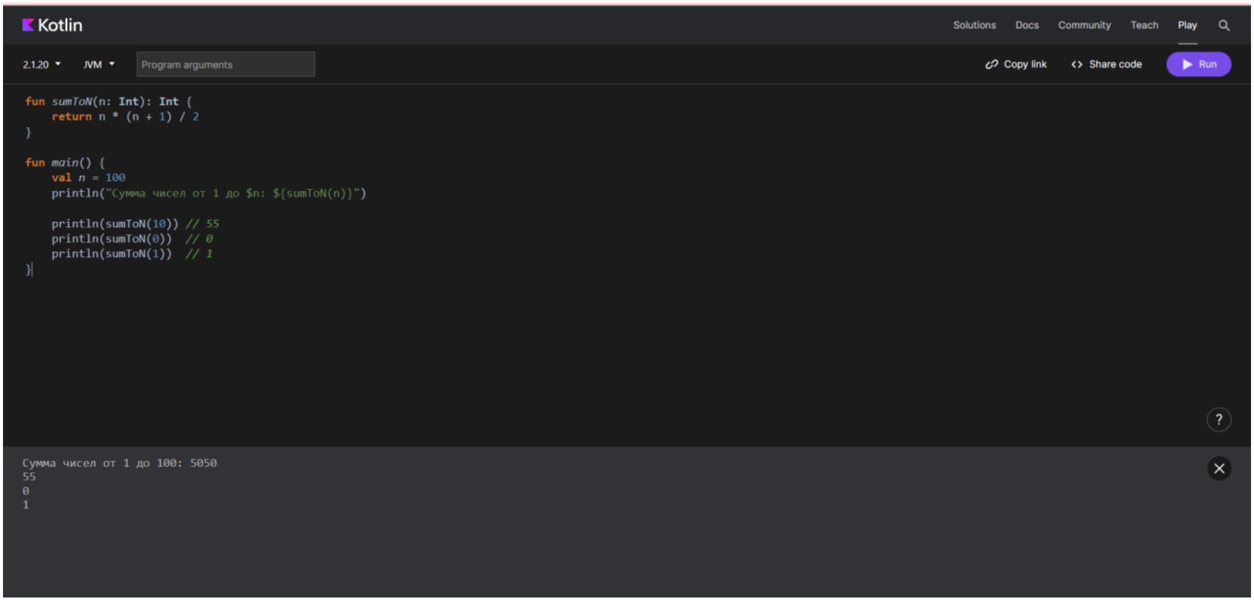
29)



```kotlin
fun findMinMax(array: IntArray): Pair<Int, Int> {
    if (array.isEmpty()) throw NoSuchElementException("Array is empty")

    var min = array[0]
    var max = array[0]

    for (i in 1 until array.size) {
        when {
            array[i] < min -> min = array[i]
            array[i] > max -> max = array[i]
        }
    }

    return Pair(min, max)
}

fun main() {
    val numbers = intArrayOf(5, 3, 8, 1, 4, 9, 2)

    val (min1, max1) = findMinMax(numbers)
    println("Min: $min1, Max: $max1")

}
```

```
Min: 1, Max: 9
```

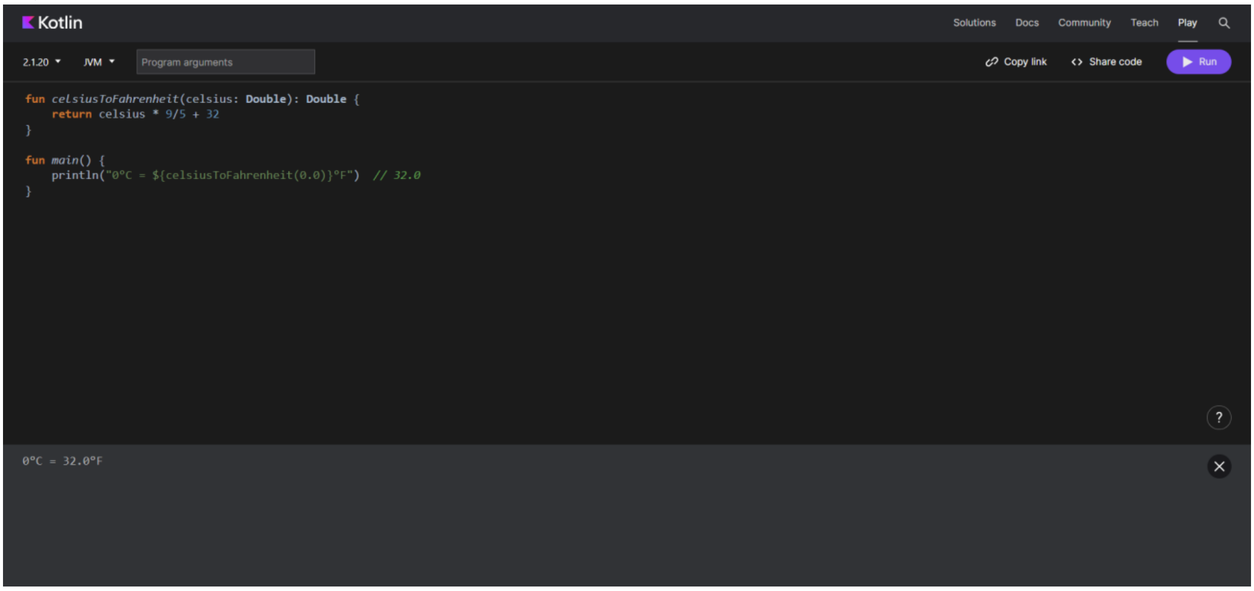
30)



```kotlin
fun sumToN(n: Int): Int {
    return n * (n + 1) / 2
}

fun main() {
    val n = 100
    println("Сумма чисел от 1 до $n: ${sumToN(n)}")

    println(sumToN(10)) // 55
    println(sumToN(0))  // 0
    println(sumToN(1))  // 1
}
```

```
Сумма чисел от 1 до 100: 5050
55
0
1
```

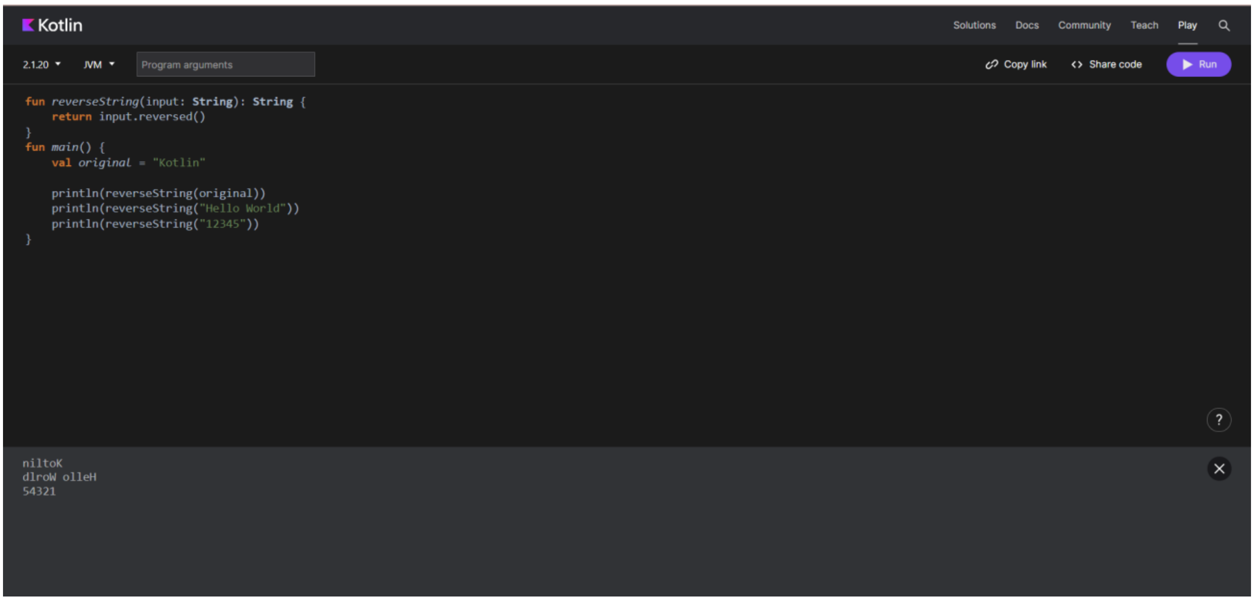
31)



```kotlin
fun celsiusToFahrenheit(celsius: Double): Double {
    return celsius * 9/5 + 32
}

fun main() {
    println("0°C = ${celsiusToFahrenheit(0.0)}°F")  // 32.0
}
```

```
0°C = 32.0°F
```

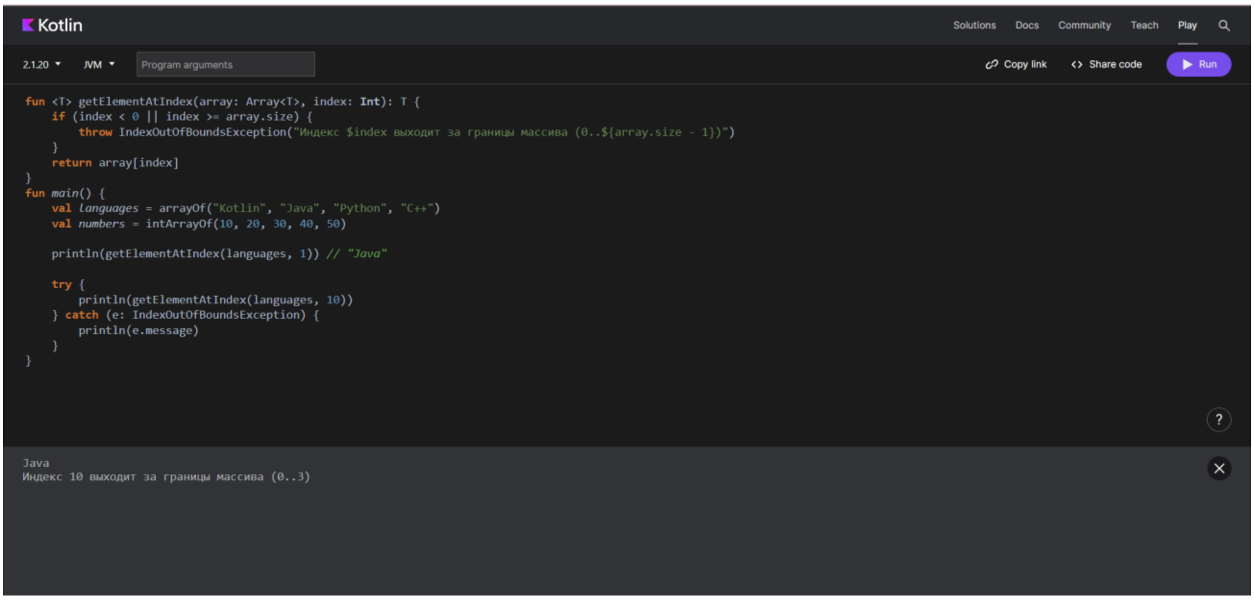
32)



```
fun reverseString(input: String): String {
    return input.reversed()
}
fun main() {
    val original = "Kotlin"

    println(reverseString(original))
    println(reverseString("Hello World"))
    println(reverseString("12345"))
}
```

```
niltoK
dlroW olleH
54321
```

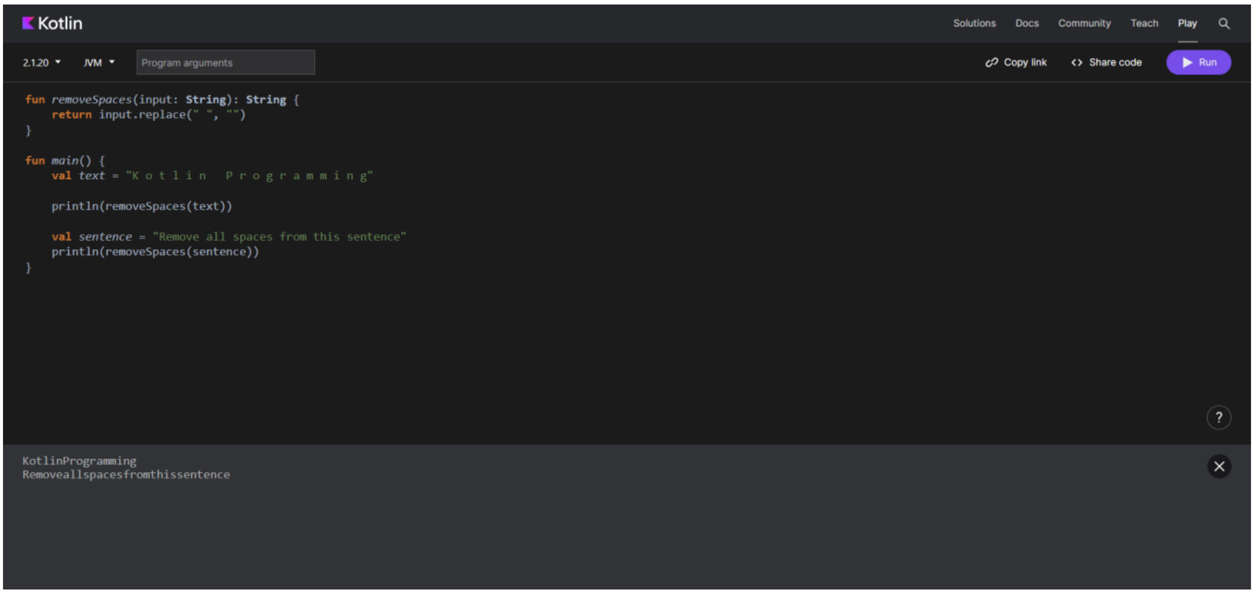
33)



```
fun <T> getElementAtIndex(array: Array<T>, index: Int): T {
    if (index < 0 || index >= array.size) {
        throw IndexOutOfBoundsException("Индекс $index выходит за границы массива (0..${array.size - 1})")
    }
    return array[index]
}
fun main() {
    val languages = arrayOf("Kotlin", "Java", "Python", "C++")
    val numbers = intArrayOf(10, 20, 30, 40, 50)

    println(getElementAtIndex(languages, 1)) // "Java"

    try {
        println(getElementAtIndex(languages, 10))
    } catch (e: IndexOutOfBoundsException) {
        println(e.message)
    }
}
```

```
Java
Индекс 10 выходит за границы массива (0..3)
```

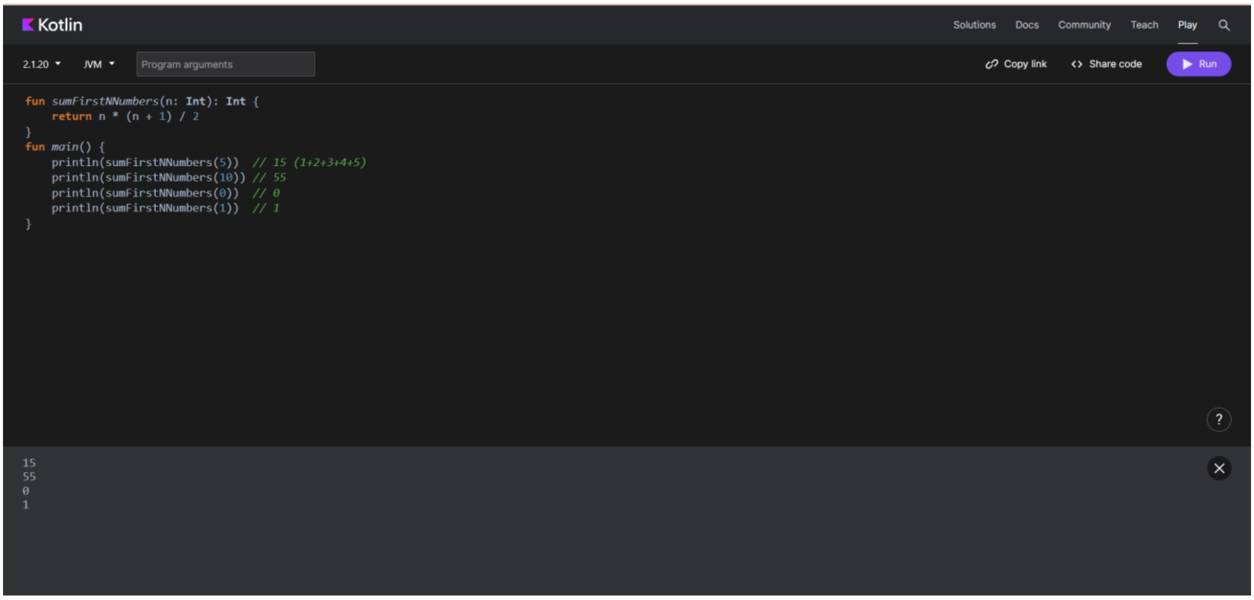
34)



```
fun removeSpaces(input: String): String {
    return input.replace(" ", "")
}

fun main() {
    val text = "K o t l i n   P r o g r a m m i n g"

    println(removeSpaces(text))

    val sentence = "Remove all spaces from this sentence"
    println(removeSpaces(sentence))
}
```

```
KotlinProgramming
Removeallspacesfromthissentence
```

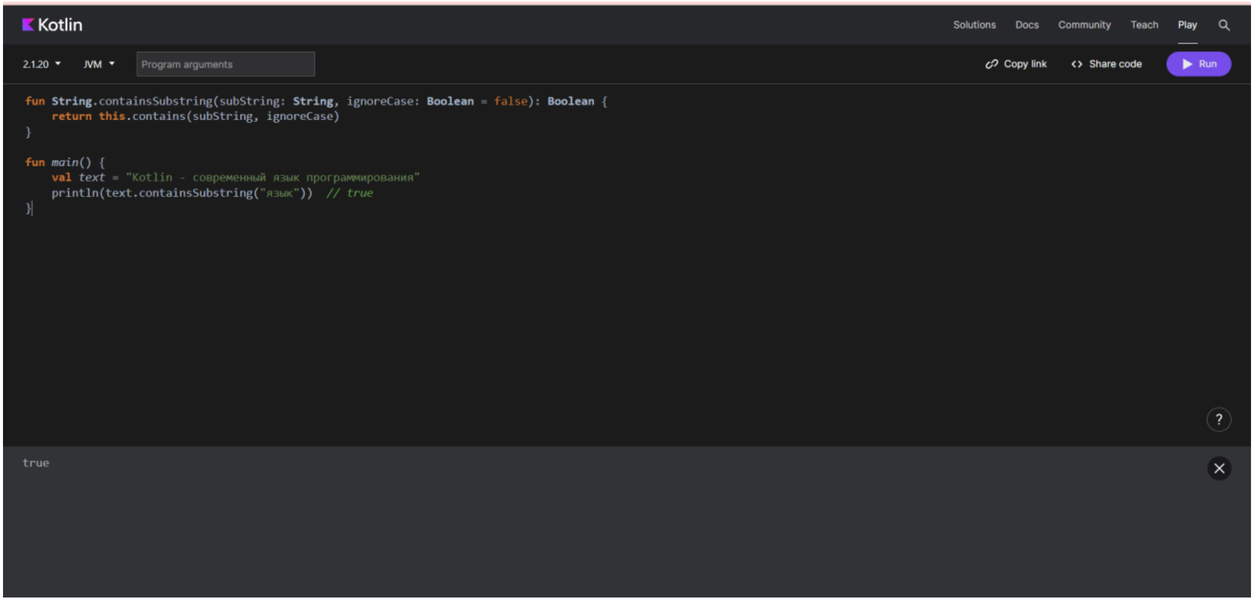
35)



```
fun sumFirstNNumbers(n: Int): Int {
    return n * (n + 1) / 2
}
fun main() {
    println(sumFirstNNumbers(5))  // 15 (1+2+3+4+5)
    println(sumFirstNNumbers(10)) // 55
    println(sumFirstNNumbers(0))  // 0
    println(sumFirstNNumbers(1))  // 1
}
```

```
15
55
0
1
```

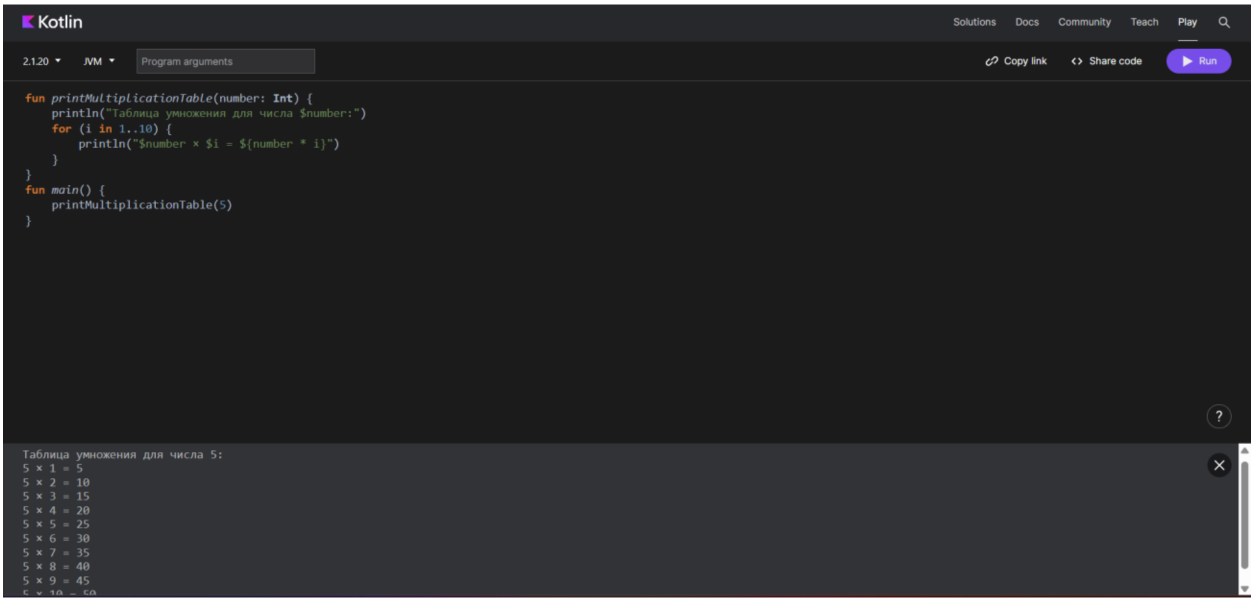
36)



```
fun String.containsSubstring(subString: String, ignoreCase: Boolean = false): Boolean {
    return this.contains(subString, ignoreCase)
}

fun main() {
    val text = "Kotlin - современный язык программирования"
    println(text.containsSubstring("язык"))  // true
}
```

```
true
```

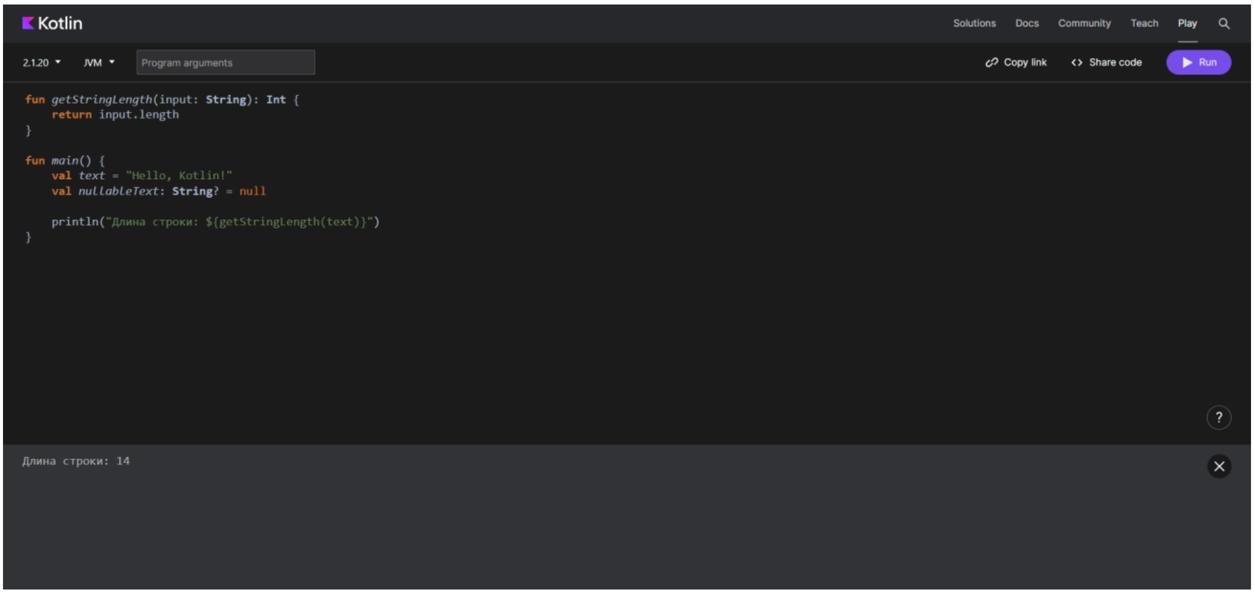
37)



```
fun printMultiplicationTable(number: Int) {
    println("Таблица умножения для числа $number:")
    for (i in 1..10) {
        println("$number × $i = ${number * i}")
    }
}
fun main() {
    printMultiplicationTable(5)
}
```

```
Таблица умножения для числа 5:
5 × 1 = 5
5 × 2 = 10
5 × 3 = 15
5 × 4 = 20
5 × 5 = 25
5 × 6 = 30
5 × 7 = 35
5 × 8 = 40
5 × 9 = 45
5 × 10 = 50
```

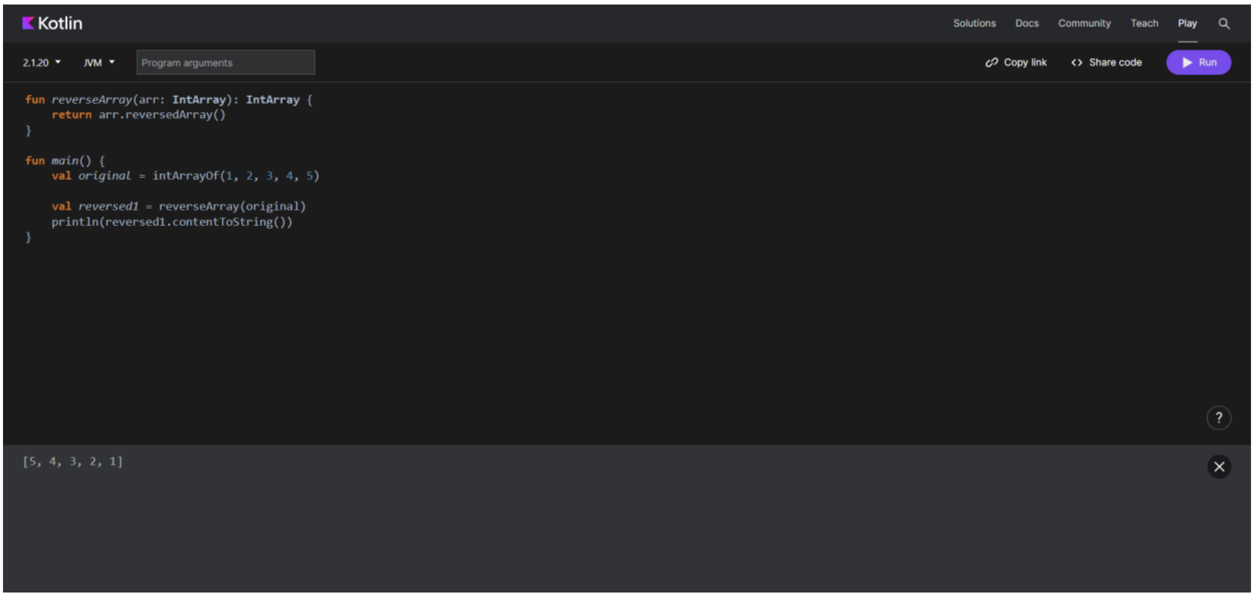
38)



```
fun getStringLength(input: String): Int {
    return input.length
}

fun main() {
    val text = "Hello, Kotlin!"
    val nullableText: String? = null

    println("Длина строки: ${getStringLength(text)}")
}
```

```
Длина строки: 14
```

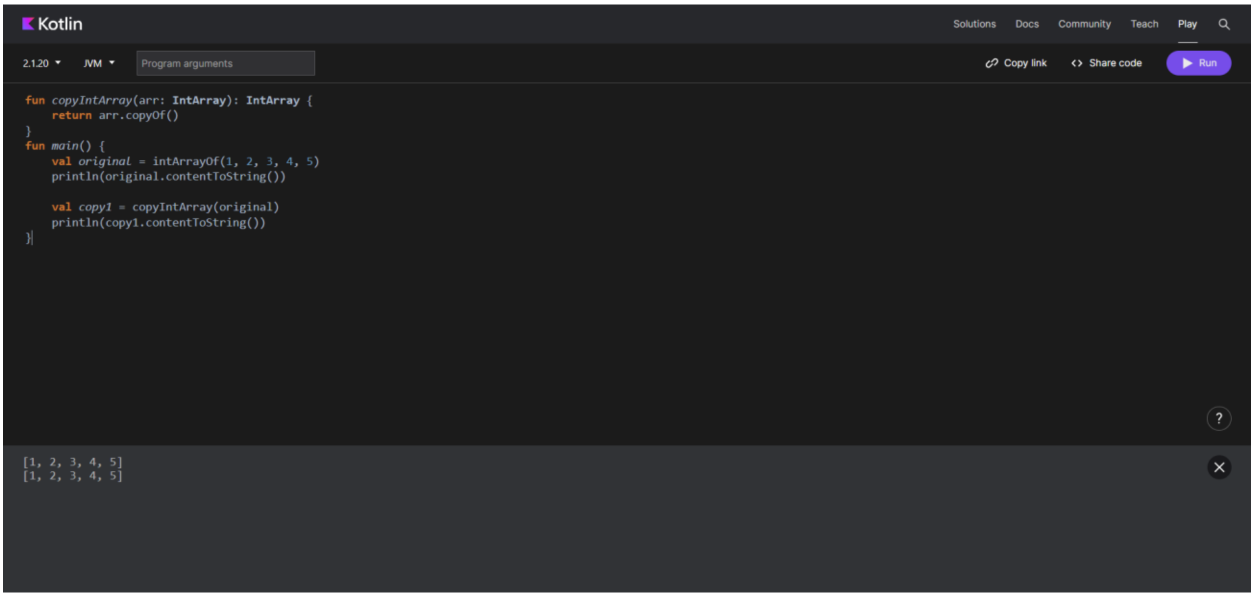
39)



```
fun reverseArray(arr: IntArray): IntArray {
    return arr.reversedArray()
}

fun main() {
    val original = intArrayOf(1, 2, 3, 4, 5)

    val reversed1 = reverseArray(original)
    println(reversed1.contentToString())
}
```

```
[5, 4, 3, 2, 1]
```

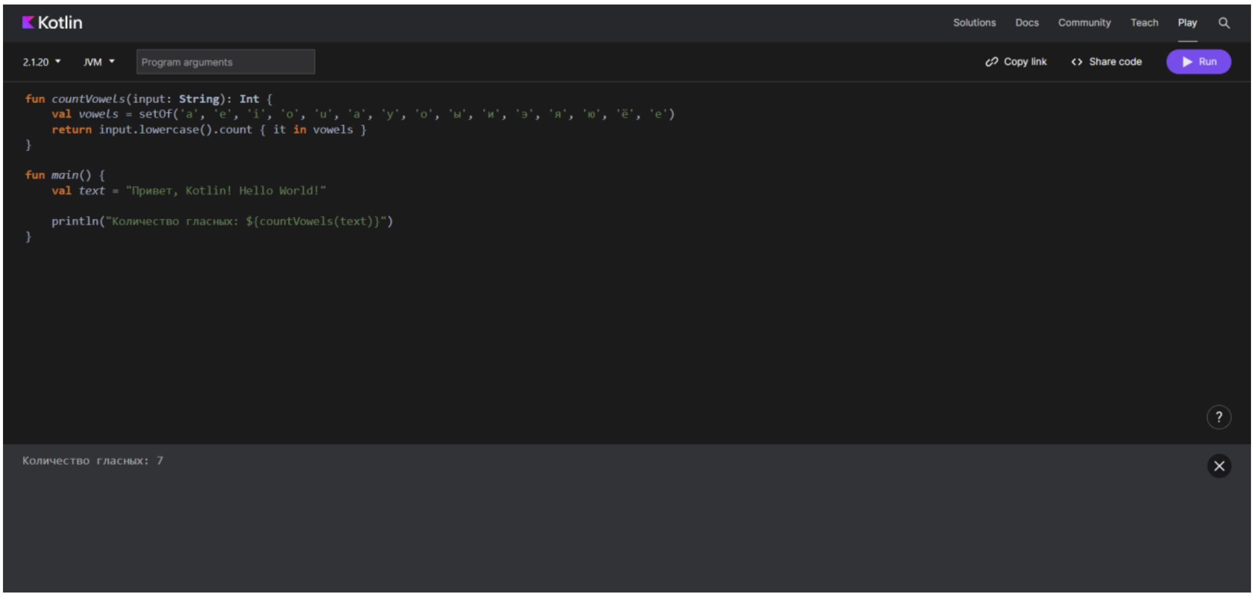
40)



```
fun copyIntArray(arr: IntArray): IntArray {
    return arr.copyOf()
}
fun main() {
    val original = intArrayOf(1, 2, 3, 4, 5)
    println(original.contentToString())

    val copy1 = copyIntArray(original)
    println(copy1.contentToString())
}
```

```
[1, 2, 3, 4, 5]
[1, 2, 3, 4, 5]
```

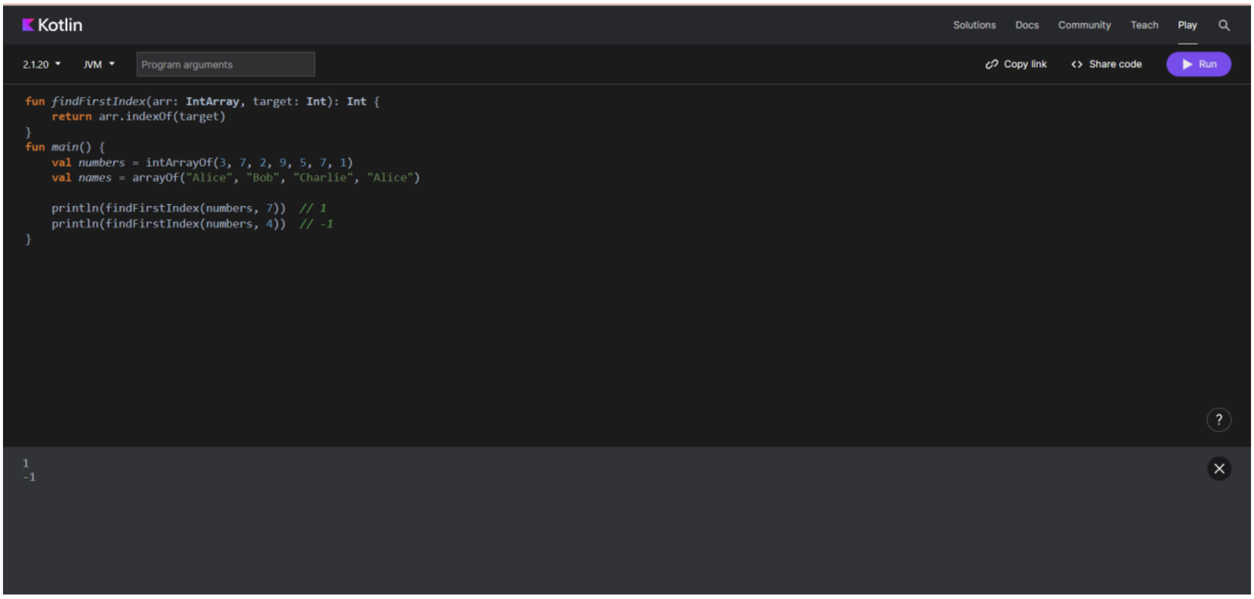
41)



```
fun countVowels(input: String): Int {
    val vowels = setOf('a', 'e', 'i', 'o', 'u', 'а', 'у', 'о', 'ы', 'и', 'э', 'я', 'ю', 'ё', 'е')
    return input.lowercase().count { it in vowels }
}

fun main() {
    val text = "Привет, Kotlin! Hello World!"

    println("Количество гласных: ${countVowels(text)}")
}
```

```
Количество гласных: 7
```

42)



```
fun findFirstIndex(arr: IntArray, target: Int): Int {
    return arr.indexOf(target)
}
fun main() {
    val numbers = intArrayOf(3, 7, 2, 9, 5, 7, 1)
    val names = arrayOf("Alice", "Bob", "Charlie", "Alice")

    println(findFirstIndex(numbers, 7))  // 1
    println(findFirstIndex(numbers, 4))  // -1
}
```

```
1
-1
```