

1)

The screenshot shows the IntelliJ IDEA interface with a project named 'LABA'. The left sidebar displays the project structure, including a 'src/main/kotlin' directory containing a file named 'phi.kt'. The main editor window shows the code for 'phi.kt':

```
1 fun sumList(numbers: List<Int>): Int {  
2     return numbers.sum()  
3 }  
4  
5 main() {  
6     val numbers = listOf(1, 2, 3, 4, 5, 6, 7)  
7     println("Сумма элементов списка: ${sumList(numbers)}")  
8 }
```

The Run console at the bottom shows the command executed: `C:\Users\atypo\jdk\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.encoding=UTF-8`, followed by the output: `Сумма элементов списка: 28`. The status bar at the bottom indicates the file is at line 5, column 45, with CRLF line endings, UTF-8 encoding, and 4 spaces for indentation.

2)

The screenshot shows the IntelliJ IDEA interface with the same 'LABA' project. The left sidebar shows the project structure, with 'src/main/kotlin' containing 'phi.kt'. The main editor window shows the code for 'phi.kt':

```
1 fun differenceBetweenMaxAndMin(numbers: List<Int>): Int {  
2     if (numbers.isEmpty()) return 0  
3     return numbers.maxOrNull()!! - numbers.minOrNull()!!  
4 }  
5  
6 main() {  
7     val numbers = listOf(10, 5, 20, 3, 8)  
8     println("Разность max и min: ${differenceBetweenMaxAndMin(numbers)}")  
9  
10    val emptyList = listOf<Int>()  
11    println("Разность для пустого списка: ${differenceBetweenMaxAndMin(emptyList)}")  
12 }
```

The Run console at the bottom shows the command executed: `C:\Users\atypo\jdk\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.encoding=UTF-8`, followed by the output: `Разность max и min: 17` and `Разность для пустого списка: 0`. The status bar at the bottom indicates the file is at line 2, column 36, with CRLF line endings, UTF-8 encoding, and 4 spaces for indentation.

3)

```

1 fun combineLists(list1: List<Int>, list2: List<Int>): List<Int> {
2     return list1 + list2
3 }
4 fun combineListsAlternative(list1: List<Int>, list2: List<Int>): List<Int> {
5     return mutableListOf<Int>().apply {
6         addAll(list1)
7         addAll(list2)
8     }
9 }
10
11 fun main() {
12     val listA = listOf(1, 2, 3)
13     val listB = listOf(4, 5, 6)
14     val combined = combineLists(listA, listB)
15     println("Объединенный список: $combined")
16 }

```

Run

```

C:\Users\atypo\jdk\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.encoding=UTF-8
Объединенный список: [1, 2, 3, 4, 5, 6]
Process finished with exit code 0

```

4)

```

1 fun isProfitable(prob: Double, prize: Double, pay: Double): Boolean {
2     return prob * prize > pay
3 }
4
5 fun isProfitableVerbose(prob: Double, prize: Double, pay: Double): Boolean {
6     if (prob * prize > pay) {
7         return true
8     } else {
9         return false
10    }
11 }
12
13 fun main() {
14     println(isProfitable(prob=0.5, prize=10.0, pay=4.0))
15     println(isProfitable(prob=0.1, prize=100.0, pay=10.0))
16     println(isProfitable(prob=2.0, prize=5.0, pay=9.0))
17 }

```

Run

```

C:\Users\atypo\jdk\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.encoding=UTF-8
true
false
true
Process finished with exit code 0

```

5)

```

1 fun checkProfit(prob: Double, prize: Double, pay: Double): Boolean {
2     return prob * prize > pay
3 }
4
5 fun main() {
6     println(checkProfit(0.2, 50.0, 9.0))
7     println(checkProfit(0.1, 30.0, 5.0))
8     println(checkProfit(0.5, 10.0, 15.0))
9 }

```

```

C:\Users\atypo\jdk\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.encoding=UTF-8
true
false
false
Process finished with exit code 0

```

6)

```

1 fun isSumLessThan100(a: Int, b: Int): Boolean {
2     return a + b < 100
3 }
4
5 fun isSumLessThan100Verbose(a: Int, b: Int): Boolean {
6     if (a + b < 100) {
7         return true
8     } else {
9         return false
10    }
11 }
12
13 fun main() {
14     println(isSumLessThan100(50, 40))
15     println(isSumLessThan100(50, 60))
16     println(isSumLessThan100(99, 0))
17     println(isSumLessThan100(100, -1))
18 }

```

```

C:\Users\atypo\jdk\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.encoding=UTF-8
true
false
true
true
Process finished with exit code 0

```

7)

```

1 fun isDivisibleBy100(number: Int): Boolean {
2     return number % 100 == 0
3 }
4
5 fun isDivisibleBy100(verbose: number: Int): Boolean {
6     if (number % 100 == 0) {
7         return true
8     }
9     return false
10 }
11
12 fun main() {
13     println(isDivisibleBy100(200)) // true
14     println(isDivisibleBy100(150)) // false
15     println(isDivisibleBy100(-300)) // true
16     println(isDivisibleBy100(0)) // true
17     println(isDivisibleBy100(99)) // false
18 }

```

Run

```

C:\Users\atypo\.jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=54569" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8
true
false
true
true
false
Process finished with exit code 0

```

LABA > src > main > kotlin > isDivisibleBy100Verbose

8)

```

1 fun checkPowerEquality(n: Int, k: Int): Boolean {
2     return when {
3         k == 0 -> n == 1 // 0-й степеня 0^0 = 1
4         k == 1 -> n == 1 // 1-й степеня n^1 = n
5         k == -1 -> n == -1 || n == 1 // (-1)^n = -1 and 1
6         k > 0 -> {
7             var result = 1
8             repeat(k) { result *= n }
9             result == n
10        }
11        else -> false
12    }
13 }
14
15 fun main() {
16     println(checkPowerEquality(0, 1))
17     println(checkPowerEquality(0, 2))
18     println(checkPowerEquality(27, 3))
19     println(checkPowerEquality(256, 4))
20     println(checkPowerEquality(-1, -1))
21     println(checkPowerEquality(0, 0))
22     println(checkPowerEquality(0, -1))
23     println(checkPowerEquality(100, 3))
24     println(checkPowerEquality(0, -8, -2))
25 }

```

Run

```

C:\Users\atypo\.jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=54569" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8
true
true
true
true
true
true
false
false
Process finished with exit code 0

```

LABA > src > main > kotlin > main

The screenshot displays the IntelliJ IDEA IDE interface. The top toolbar includes icons for file operations, search, and running code. The left sidebar shows the project structure for 'LABA', with the file 'isKtoEqual.kt' selected under 'src/main/kotlin'. The main editor window shows the following Kotlin code:

```
1 fun isKtoEqual(n: Int, k: Int): Boolean {
2     return when {
3         k == 0 -> n == 1
4         k == 1 -> n == 1
5         k == -1 -> n == -1 || n == 1
6         k > 0 -> {
7             var result = 1L
8             repeat(k) {
9                 result *= k.toLong()
10                if (result > Int.MAX_VALUE) return false
11            }
12            result == n.toLong()
13        }
14        else -> false
15    }
16 }
17
18 fun main() {
19     println(isKtoEqual(n = 1, k = 1))
20     println(isKtoEqual(n = 250, k = 4))
21     println(isKtoEqual(n = 3125, k = 5))
22     println(isKtoEqual(n = -1, k = -1))
23     println(isKtoEqual(n = 1, k = -1))
24     println(isKtoEqual(n = 1, k = 0))
25     println(isKtoEqual(n = 100, k = 3))
26     println(isKtoEqual(n = Int.MAX_VALUE, k = 2))
27 }
28
```

The bottom panel shows the 'Run' configuration for 'PhiKt' and the execution output:

```
Run
PhiKt
true
true
true
true
true
false
false
Process finished with exit code 0
```

The status bar at the bottom indicates the current file path as 'LABA > src > main > kotlin > PhiKt' and the encoding as 'UTF-8'.

The screenshot displays the IntelliJ IDEA IDE interface. The top toolbar includes icons for running (a green play button), debugging (a bug icon), and other development tools. Below the toolbar, the 'Project' view on the left shows a file tree for a project named 'LABA'. The tree structure is as follows:

- LABA C:\Users\atypol\IdeaProjects\LABA
  - gradle
  - .idea
  - kotlin
  - build
  - gradle
  - src
    - main
      - kotlin
        - Φ.kt
      - resources
    - test
    - gitignore
    - build.gradle.kts
    - gradle.properties
    - gradlew
    - gradlew.bat

The main editor window shows the contents of 'Φ.kt':1 fun repetition(txt: String, n: Int): String {  
2 return when {  
3 n <= 0 -> ""  
4 n == 1 -> txt  
5 else -> txt + repetition(txt, n - 1)  
6 }  
7 }  
8  
9 fun main() {  
10 println(repetition("aa", 7))  
11 println(repetition("kiwi", 1))  
12 println(repetition("cherry", 2))  
13 println(repetition("a", 5))  
14 println(repetition("test", 0))  
15 }

A red cursor is positioned at line 10, column 10. To the right of the code editor, there are several status icons: a green checkmark, a magnifying glass, a refresh symbol, and a share icon.

At the bottom, the 'Run' tool window is active, displaying the output of the program:C:\Users\atypol\.jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\atypol\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea\_rt.jar=56727"-Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.enm  
asababababao  
kiskiki  
cherrycherry  
aaaaa  
  
Process finished with exit code 0

The Run tab title bar indicates it is running 'Φ.kt'.

The bottom status bar shows the current path: LABA > src > main > kotlin > Φ.kt > main, along with technical details like '11:33 CRFL UTF-8 4 spaces'.

```
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())
}

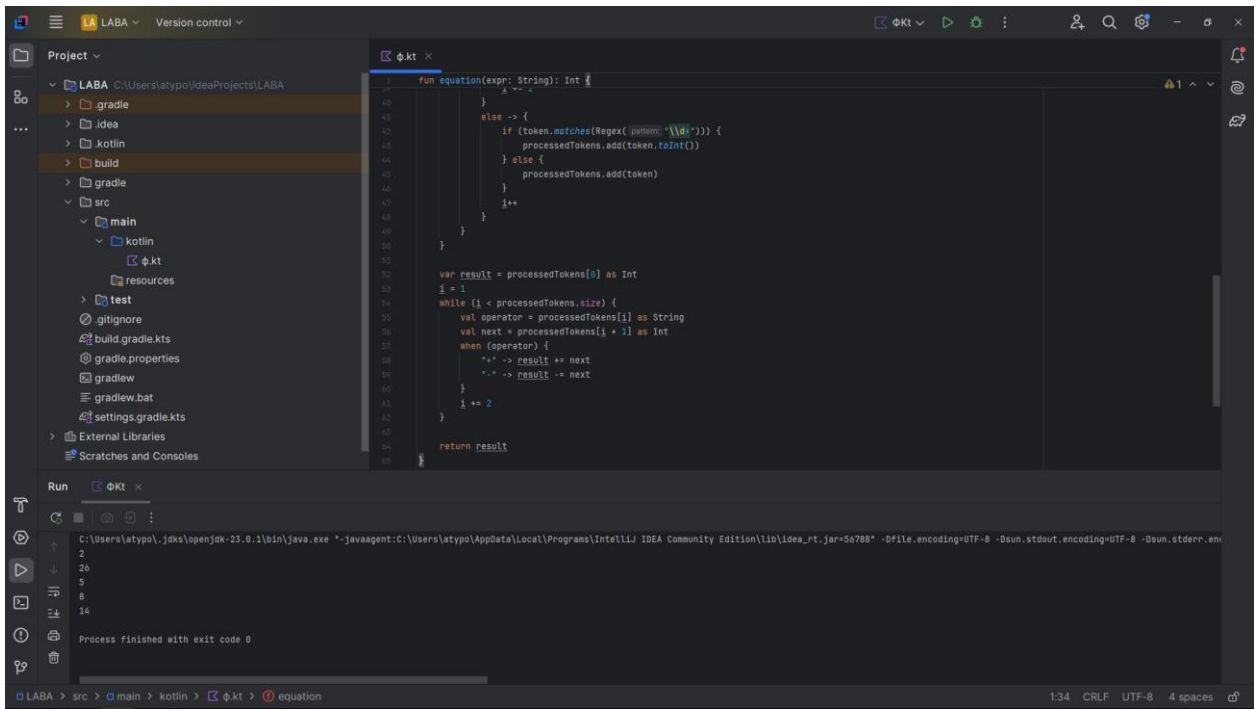
val processedTokens = mutableListOf<Any>()
var i = 0
while (i < tokens.size) {
    val token = tokens[i]
    when (token) {
        "*" -> {
            val prev = processedTokens.removeAt(processedTokens.size - 1)
as Int
            val next = tokens[i + 1].toInt()
            processedTokens.add(prev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

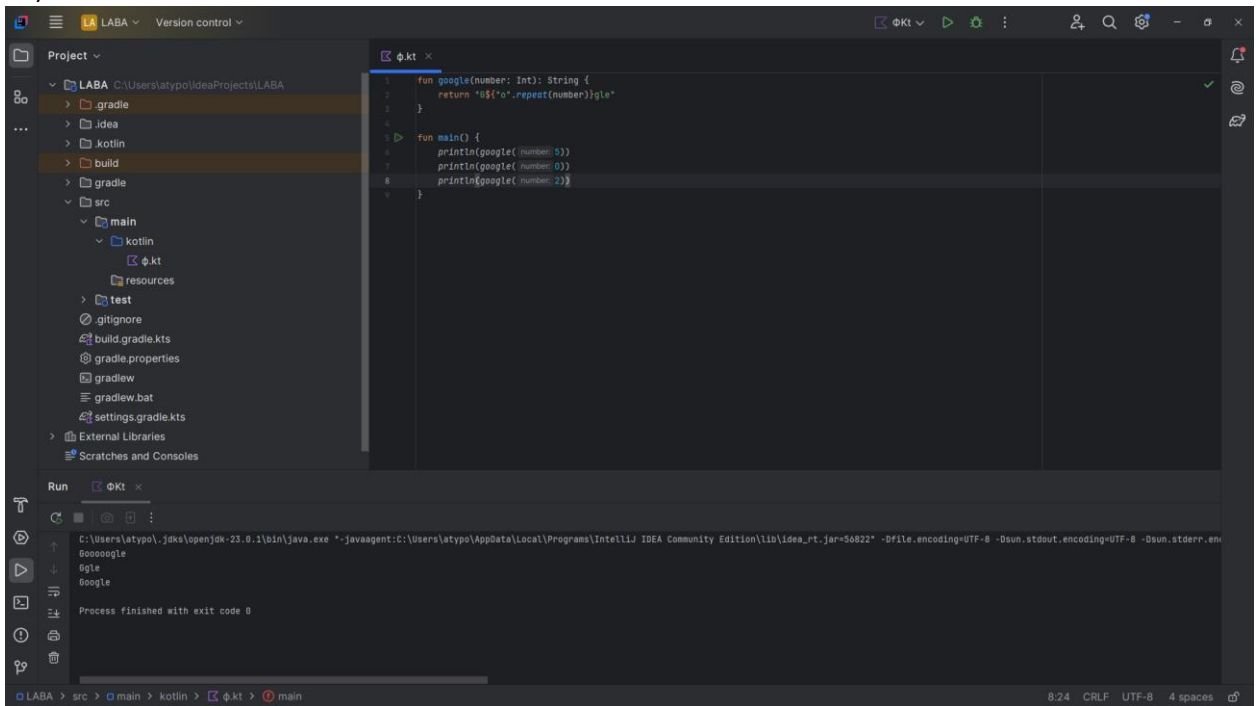
return result
}

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"))
}

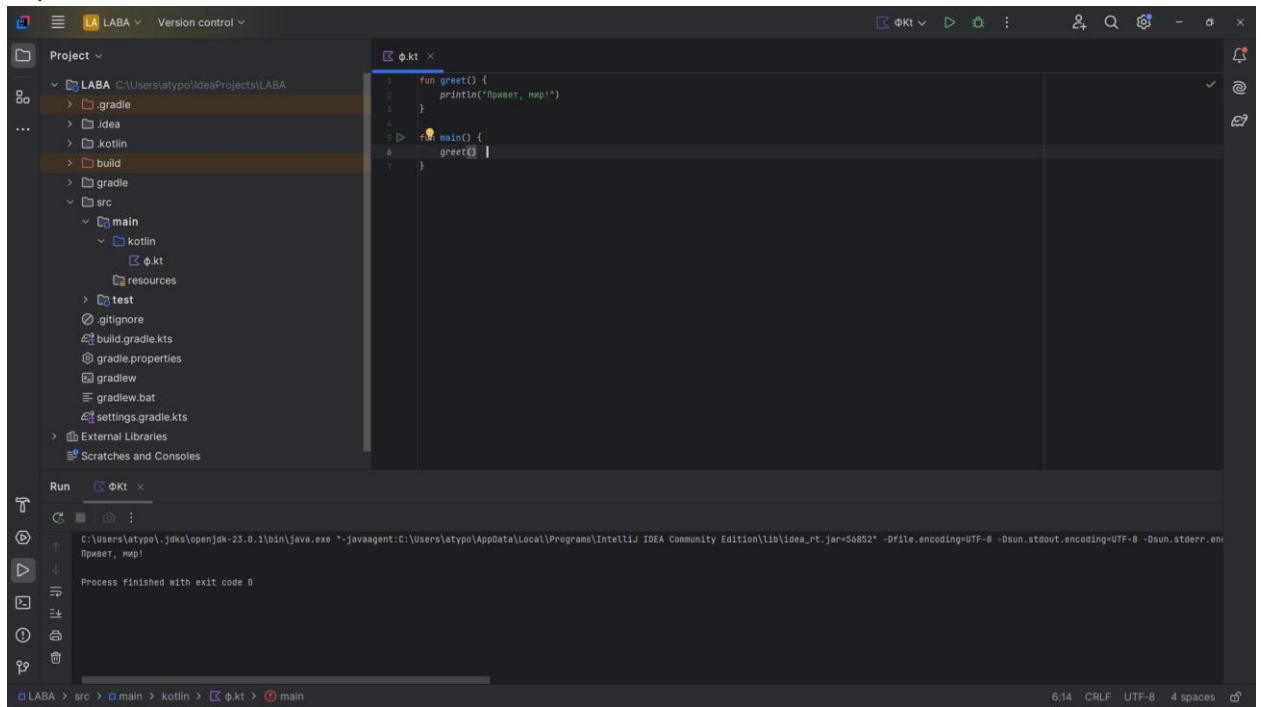
```



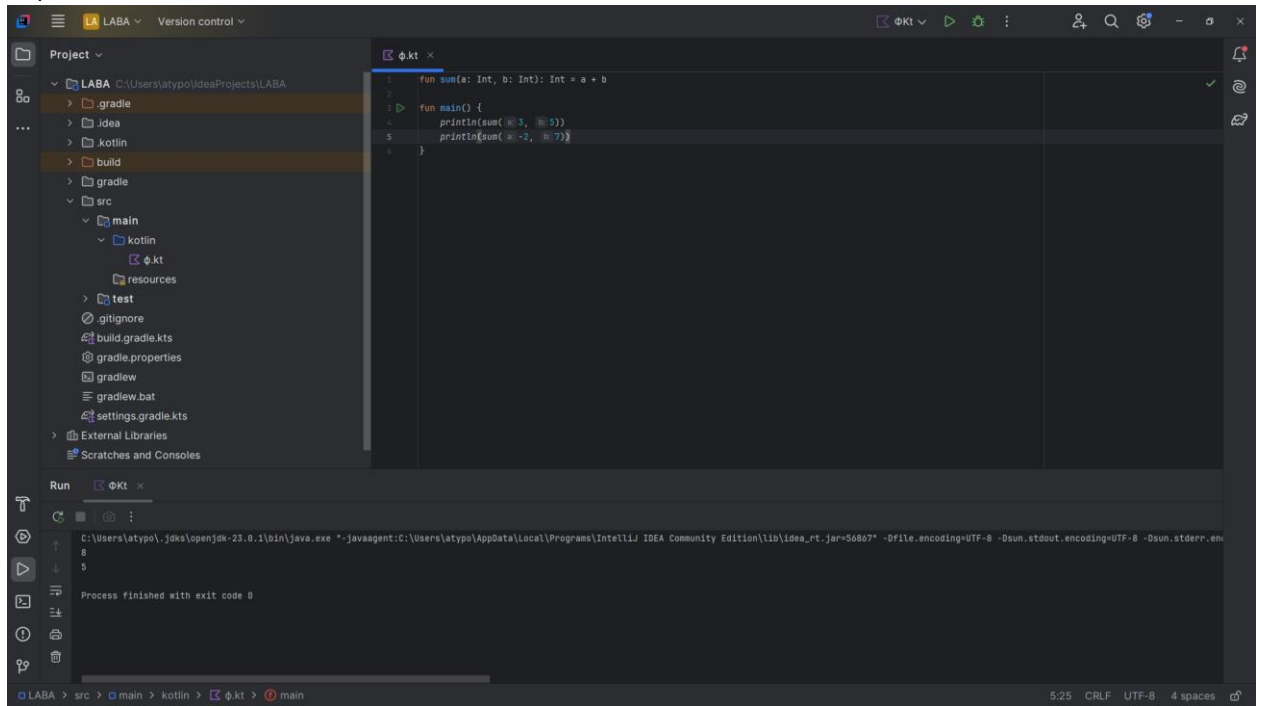
12)



13)



14)





15)

The screenshot shows the IntelliJ IDEA interface with a Kotlin file open. The code defines a function `maxOfTwo` and a `main` function to test it.

```

1 fun maxOfTwo(a: Int, b: Int): Int = if (a > b) a else b
2
3 fun main() {
4     println(maxOfTwo(3, 5))
5     println(maxOfTwo(-2, -7))
6     println(maxOfTwo(10, 15))
7 }

```

The Run console shows the output of the program:

```

C:\Users\atypo\.jds\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
-2
10
Process finished with exit code 0

```

The status bar at the bottom indicates the file is at `LABA > src > main > kotlin > φ.kt > main` with a cursor at line 6, column 31.

16)

The screenshot shows the IntelliJ IDEA interface with a Kotlin file open. The code defines a function `isEven` and a `main` function to test it.

```

1 fun isEven(number: Int): Boolean = number % 2 == 0
2
3 fun main() {
4     println(isEven(4))
5     println(isEven(7))
6     println(isEven(0))
7     println(isEven(-3))
8     println(isEven(-8))
9 }

```

The Run console shows the output of the program:

```

C:\Users\atypo\.jds\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

```

The status bar at the bottom indicates the file is at `LABA > src > main > kotlin > φ.kt > main` with a cursor at line 8, column 26.

17)

The screenshot shows the IntelliJ IDEA IDE with a Kotlin file open. The code defines a recursive factorial function and a main function to test it. The Run console shows the output of the main function.

```

1 fun factorial(n: Int): Long {
2     require(n >= 0) { "Факториал отрицательного числа не определен" }
3     return if (n <= 1) 1 else n * factorial(n - 1)
4 }
5
6 fun main() {
7     println(factorial(0))
8     println(factorial(1))
9     println(factorial(5))
10    println(factorial(10))
11 }

```

```

C:\Users\atypo\jdk\openjdk-23.0.1\bin\java.exe -javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=50952 -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8
1
1
120
3628800
Process finished with exit code 0

```

18)

The screenshot shows the IntelliJ IDEA IDE with a Kotlin file open. The code defines an isPrime function and a main function to test it. The Run console shows the output of the main function.

```

1 fun isPrime(n: Int): Boolean {
2     if (n <= 1) return false
3     if (n <= 2) return true
4     if (n % 2 == 0) return false
5
6     var i = 3
7     while (i * i <= n) {
8         if (n % i == 0) return false
9         i += 2
10    }
11    return true
12 }
13
14 fun main() {
15    println(isPrime(0))
16    println(isPrime(1))
17    println(isPrime(10))
18    println(isPrime(23))
19    println(isPrime(1))
20    println(isPrime(0))
21    println(isPrime(-5))
22 }

```

```

C:\Users\atypo\jdk\openjdk-23.0.1\bin\java.exe -javaagent:C:\Users\atypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=50952 -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8
true
true
false
true
false
false
false
false

```

19)

The screenshot shows the IntelliJ IDEA interface with a Kotlin file open. The code defines two functions: `sumArray` and `sumArrayManual`. `sumArray` uses the `sum()` extension function, while `sumArrayManual` implements the logic manually using a loop. The `main` function tests both with three different arrays. The Run console shows the program executed successfully with exit code 0.

```

1 fun sumArray(numbers: IntArray): Int = numbers.sum()
2
3 fun sumArrayManual(numbers: IntArray): Int {
4     var sum = 0
5     for (num in numbers) {
6         sum += num
7     }
8     return sum
9 }
10
11 fun main() {
12     val arr1 = IntArrayOf(1, 2, 3, 4, 5)
13     val arr2 = IntArrayOf(-1, 0, 1)
14     val arr3 = IntArrayOf()
15
16     println(sumArray(arr1))
17     println(sumArrayManual(arr2))
18     println(sumArray(arr3))
19 }

```

```

C:\Users\ltypo\.jds\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\ltypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=57810* -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.en
15
0
0
Process finished with exit code 0

```

20)

The screenshot shows the IntelliJ IDEA interface with a Kotlin file open. The code defines two functions: `findMax` and `findMaxManual`. `findMax` uses the `maxOrNull()` extension function, while `findMaxManual` implements the logic manually using a loop. The `main` function tests both with three different arrays, including an empty one. The Run console shows the program executed successfully with exit code 0, but the output indicates an exception was caught and printed.

```

1 fun findMax(numbers: IntArray): Int = numbers.maxOrNull() ?: throw IllegalArgumentException("Массив не должен быть пустым")
2
3 fun findMaxManual(numbers: IntArray): Int {
4     require(numbers.isNotEmpty()) { "Массив не должен быть пустым" }
5     var max = numbers[0]
6     for (num in numbers) {
7         if (num > max) max = num
8     }
9     return max
10 }
11
12 fun main() {
13     val arr1 = IntArrayOf(1, 2, 3, 4, 5)
14     val arr2 = IntArrayOf(-1, -5, -3)
15     val arr3 = IntArrayOf()
16
17     println(findMax(arr1))
18     println(findMaxManual(arr2))
19     println(findMax(arr3))
20
21     try {
22         findMax(IntArrayOf())
23     } catch (e: IllegalArgumentException) {
24         println(e.message)
25     }
26 }

```

```

C:\Users\ltypo\.jds\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\ltypo\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=57857* -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.en
9
-1
42
Массив не должен быть пустым
Process finished with exit code 0

```

21)

```

1 fun sortArray(numbers: IntArray): IntArray = numbers.sortedArray()
2
3 fun bubbleSort(numbers: IntArray): IntArray {
4     val sorted = numbers.copyOf()
5     for (i in sorted.indices) {
6         for (j in 0 until sorted.size - i - 1) {
7             if (sorted[j] > sorted[j + 1]) {
8                 val temp = sorted[j]
9                 sorted[j] = sorted[j + 1]
10                sorted[j + 1] = temp
11            }
12        }
13    }
14    return sorted
15 }
16
17 fun main() {
18     val arr1 = intArrayOf(1, 3, 0, 1, 2)
19     val arr2 = intArrayOf(-1, -3, 0, -5)
20     val arr3 = intArrayOf()
21
22     println(sortArray(arr1).contentToString())
23     println(bubbleSort(arr2).contentToString())
24     println(sortArray(arr3).contentToString())
25 }

```

Run console output:

```

C:\Users\laptop\jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\laptop\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=57098" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8
[1, 2, 3, 0, 1]
[-5, -3, -1, 0]
[]
Process finished with exit code 0

```

22)

```

1 fun isPalindrome(input: String): Boolean {
2     val cleanInput = input.replace("[^a-zA-Zа-яА-Я0-9]".toRegex(), "").lowercase()
3     return cleanInput == cleanInput.reversed()
4 }
5
6 fun main() {
7     val testString = "А роза упала на лапу Азора"
8     println("'${testString}' является палиндромом? ${isPalindrome(testString)}")
9 }

```

Output:

```

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

```

23)

The screenshot shows the Kotlin Playground interface. At the top, there's a header with the Kotlin logo and navigation links: Solutions, Docs, Community, Teach, Play, and a search icon. Below the header, there's a toolbar with version (2.1.20), JVM, and a dropdown menu showing 'Program arguments'. To the right of the dropdown are links for 'Copy link', 'Share code', and a 'Run' button. The main area contains the following Kotlin code:

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

Below the code, the output is displayed: "Количество символов в строке 'Hello, мир!': 11". There are also icons for help (?) and close (X) on the right side of the output area.

24)

The screenshot shows the Kotlin Playground interface. At the top, there's a header with the Kotlin logo and navigation links: Solutions, Docs, Community, Teach, Play, and a search icon. Below the header, there's a toolbar with version (2.1.20), JVM, and a dropdown menu showing 'Program arguments'. To the right of the dropdown are links for 'Copy link', 'Share code', and a 'Run' button. The main area contains the following Kotlin code:

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

Below the code, the output is displayed: "Исходная строка: 'Привет, Kotlin!'" and "В верхнем регистре: 'ПРИВЕТ, KOTLIN!'". There are also icons for help (?) and close (X) on the right side of the output area.

25)

The screenshot shows the Kotlin Playground interface. At the top, there's a header with the Kotlin logo and navigation links: Solutions, Docs, Community, Teach, Play, and a search icon. Below the header, there's a toolbar with version (2.1.20), JVM, and a dropdown menu showing 'Program arguments'. To the right of the dropdown are links for 'Copy link', 'Share code', and a 'Run' button. The main area contains the following Kotlin code:

```
fun concatenateStrings(str1: String, str2: String): String {  
    return str1 + str2  
}  
  
fun main() {  
    val hello = "Привет, "  
    val world = "мир!"  
    println(concatenateStrings(hello, world))  
}
```

Below the code, the output is displayed: "Привет, мир!". There are also icons for help (?) and close (X) on the right side of the output area.

26)

Kotlin

SolutionsDocsCommunityTeachPlay

2.1.20JVMProgram argumentsCopy linkShare codeRun

```
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)

Kotlin

SolutionsDocsCommunityTeachPlay

2.1.20JVMProgram argumentsCopy linkShare codeRun

```
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)

Kotlin

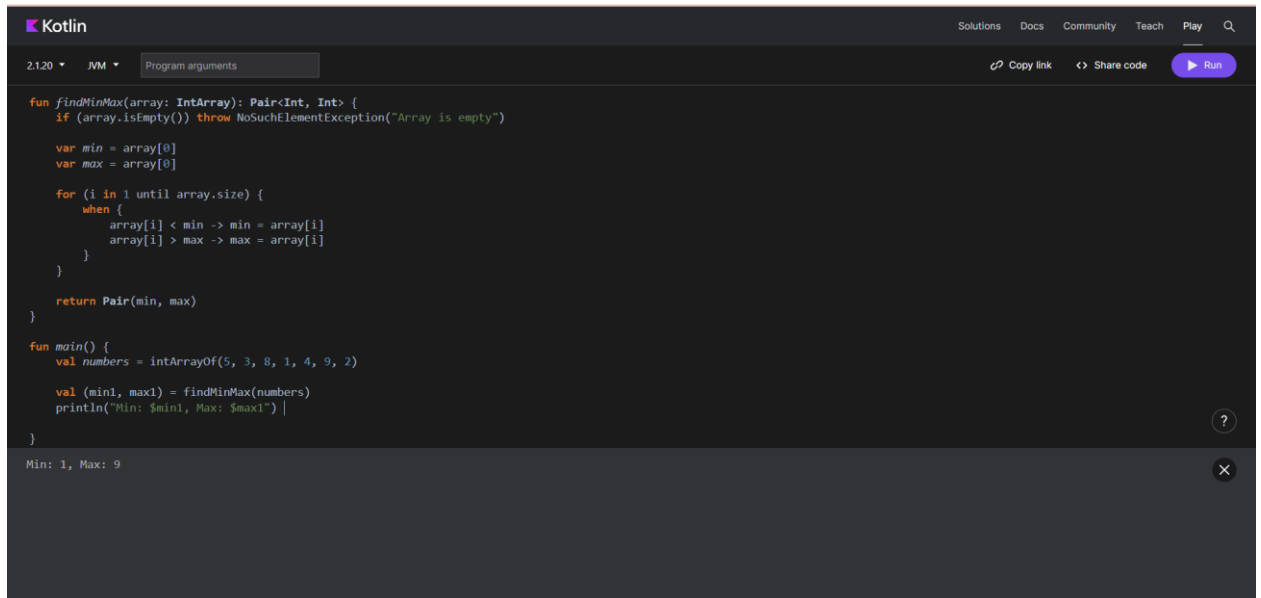
SolutionsDocsCommunityTeachPlay

2.1.20JVMProgram argumentsCopy linkShare codeRun

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

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

29)

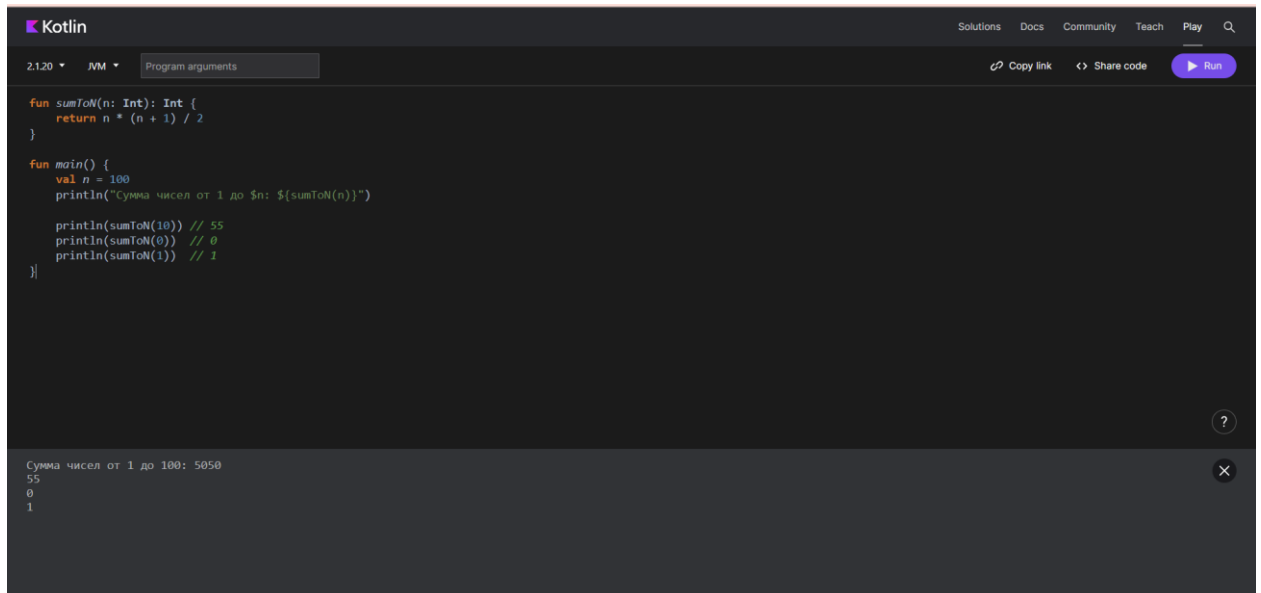


The screenshot shows the Kotlin IDE with a program to find the minimum and maximum values in an array. The code defines a function `findMinMax` that takes an `IntArray` and returns a `Pair<Int, Int>`. It uses a `for` loop to iterate through the array, updating `min` and `max` variables. The `main` function creates an array of `5, 3, 8, 1, 4, 9, 2` and prints the results.

```
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)

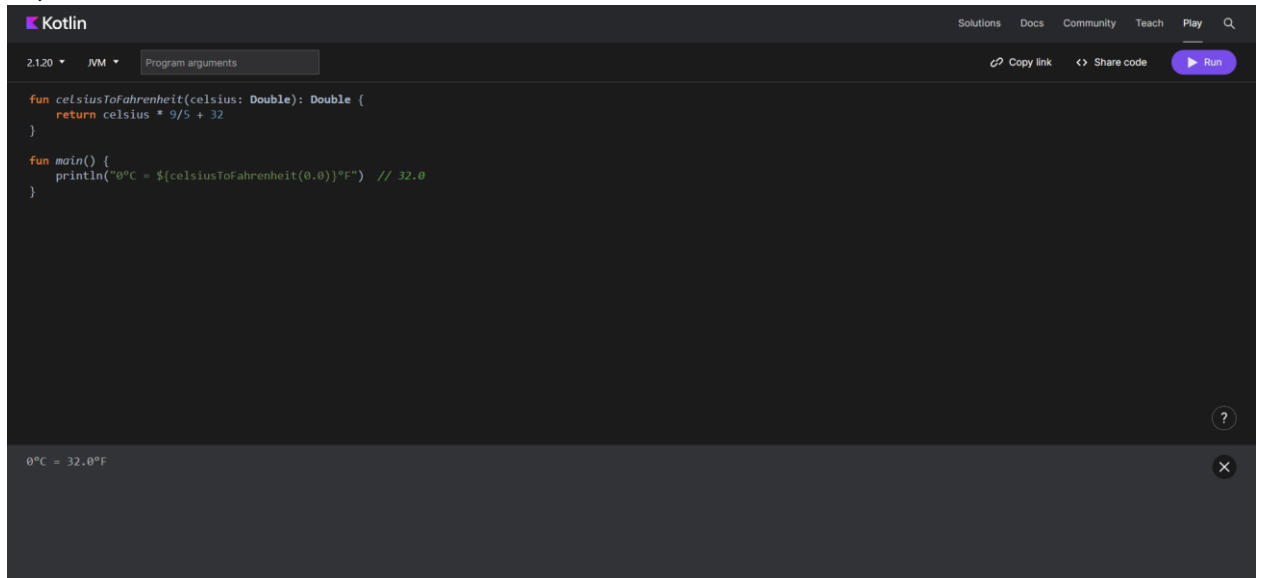


The screenshot shows the Kotlin IDE with a program to calculate the sum of numbers from 1 to `n`. The code defines a function `sumToN` that returns `n * (n + 1) / 2`. The `main` function prints the sum for `n = 100` and includes comments for `n = 0` and `n = 1`.

```
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)



The screenshot shows the Kotlin IDE with a program to convert Celsius to Fahrenheit. The code defines a function `celsiusToFahrenheit` that takes a `Double` and returns a `Double`. The `main` function prints the conversion of `0.0` Celsius to Fahrenheit.

```
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)

Kotlin

Solutions Docs Community Teach Play

2.1.20 JVM Program arguments

Copy link Share code Run

```
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)

Kotlin

Solutions Docs Community Teach Play

2.1.20 JVM Program arguments

Copy link Share code Run

```
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)

Kotlin

Solutions Docs Community Teach Play

2.1.20 JVM Program arguments

Copy link Share code Run

```
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)

Kotlin

Solutions Docs Community Teach Play

2.1.20 JVM Program arguments

Copy link Share code Run

```
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)

Kotlin

Solutions Docs Community Teach Play

2.1.20 JVM Program arguments

Copy link Share code Run

```
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)

Kotlin

Solutions Docs Community Teach Play

2.1.20 JVM Program arguments

Copy link Share code Run

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

fun main() {
    printMultiplicationTable(5)
}
```

?

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

X

38)

Kotlin

Solutions Docs Community Teach Play

2.1.20 JVM Program arguments

Copy link Share code Run

```
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

X

39)

Kotlin

Solutions Docs Community Teach Play

2.1.20 JVM Program arguments

Copy link Share code Run

```
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]

X

40)

Kotlin

Solutions Docs Community Teach Play

2.1.20 JVM Program arguments

Copy link Share code Run

```
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)

Kotlin

Solutions Docs Community Teach Play

2.1.20 JVM Program arguments

Copy link Share code Run

```
fun countVowels(input: String): Int {
    val vowels = setOf('a', 'e', 'i', 'o', 'u', 'a', 'y', 'o', 'u', 'w', 'a', 'n', 'w', 'e', 'e')
    return input.lowercase().count { it in vowels }
}

fun main() {
    val text = "Привет, Kotlin! Hello World!"
    println("Количество гласных: ${countVowels(text)}")
}
```

Количество гласных: 7

42)

Kotlin

Solutions Docs Community Teach Play

2.1.20 JVM Program arguments

Copy link Share code Run

```
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