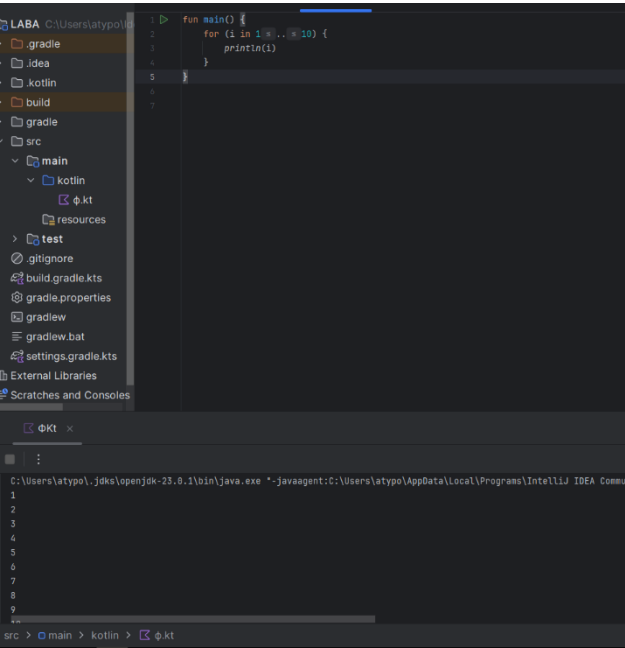
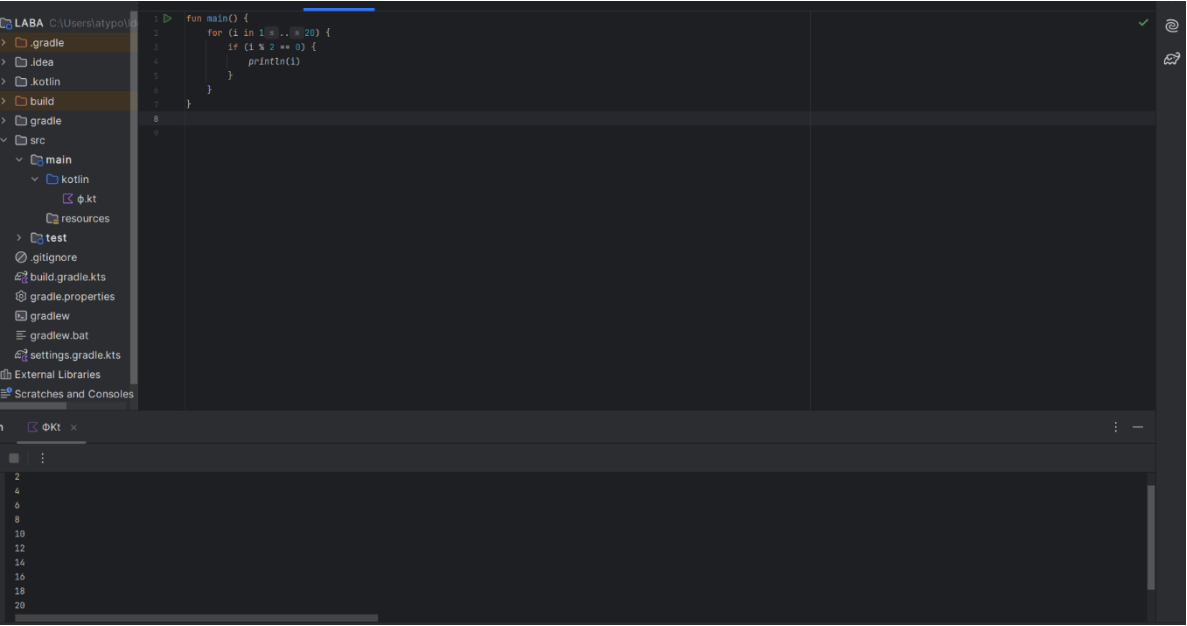


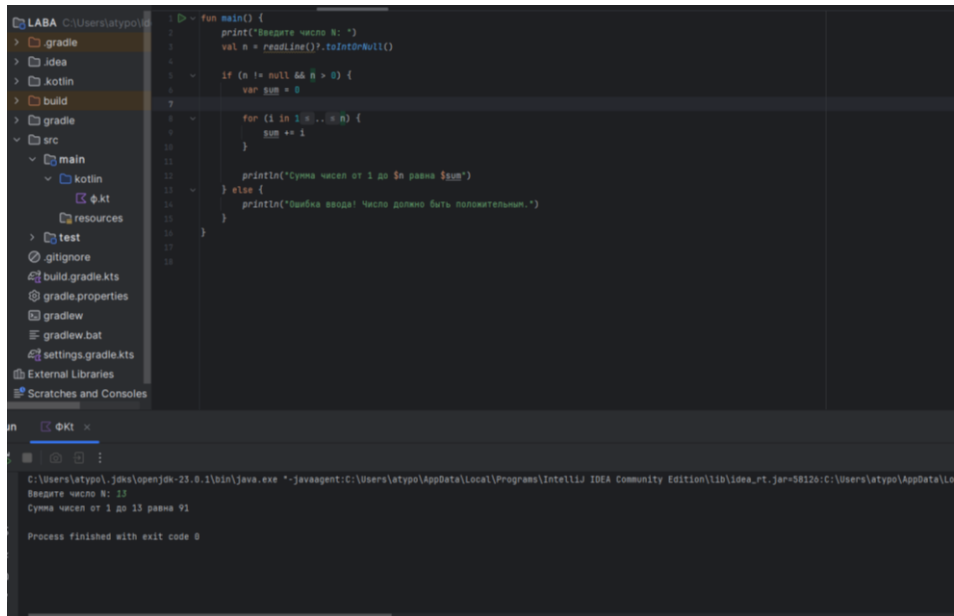
1)



2)



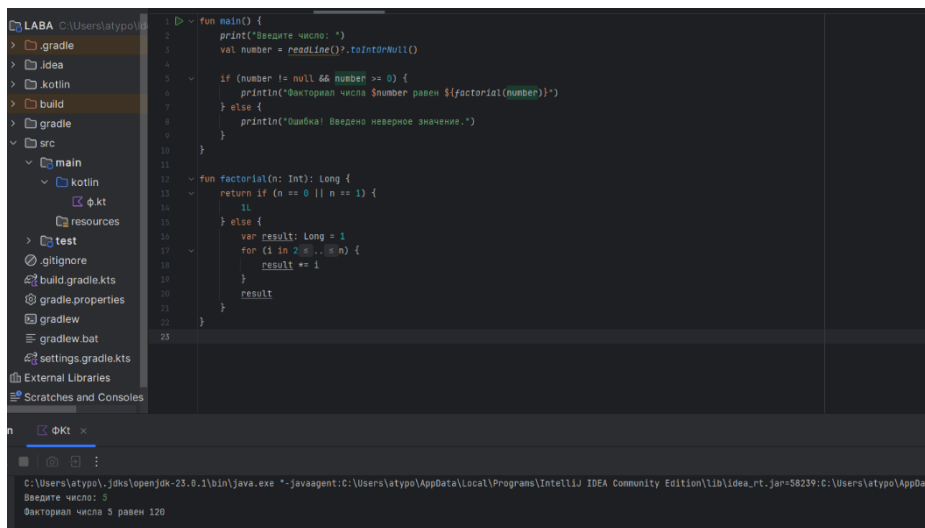
3)



```
1 fun main() {
2     print("Введите число N: ")
3     val n = readLine()?.toIntOrNull()
4
5     if (n != null && n > 0) {
6         var sum = 0
7
8         for (i in 1..n) {
9             sum += i
10        }
11
12        println("Сумма чисел от 1 до $n равна $sum")
13    } else {
14        println("Ошибка ввода! Число должно быть положительным.")
15    }
16 }
17
18
```

Process finished with exit code 0

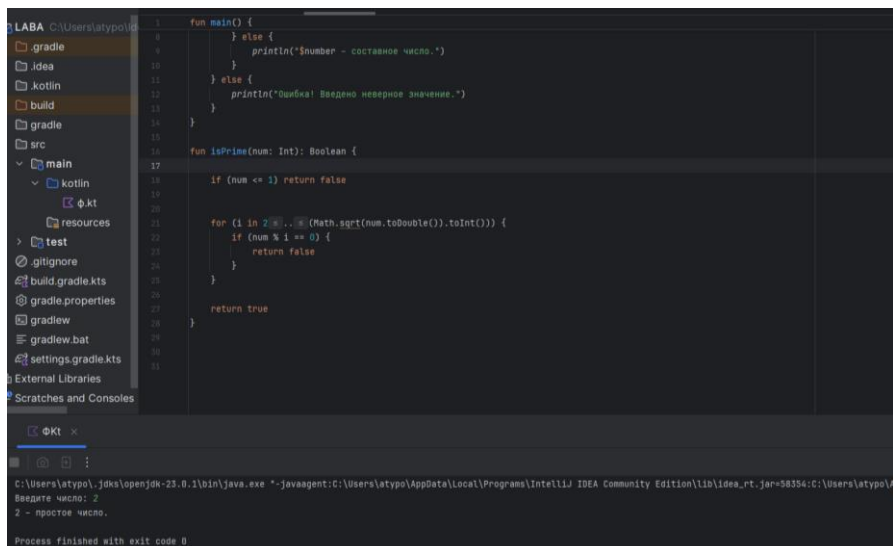
4)



```
1 fun main() {
2     print("Введите число: ")
3     val number = readLine()?.toIntOrNull()
4
5     if (number != null && number >= 0) {
6         println("Факториал числа $number равен ${factorial(number)}")
7     } else {
8         println("Ошибка! Введено неверное значение.")
9     }
10 }
11
12 fun factorial(n: Int): Long {
13     return if (n == 0 || n == 1) {
14         1L
15     } else {
16         var result: Long = 1
17         for (i in 2..n) {
18             result *= i
19         }
20         result
21     }
22 }
23
```

Process finished with exit code 0

5)



```
1 fun main() {
2     print("Введите число: ")
3     val number = readLine()?.toIntOrNull()
4
5     if (number != null && number >= 0) {
6         println("Число $number - простое или составное?")
7     } else {
8         println("Ошибка! Введено неверное значение.")
9     }
10 }
11
12 fun isPrime(num: Int): Boolean {
13     if (num <= 1) return false
14
15     for (i in 2..Math.sqrt(num.toDouble()).toInt()) {
16         if (num % i == 0) {
17             return false
18         }
19     }
20
21     return true
22 }
23
```

Process finished with exit code 0

6)

```

1 fun main() {
2     for (i in 1..10) {
3         for (j in 1..10) {
4             print("$i * $j = ${(i * j)}\t")
5         }
6         println()
7     }
8 }

```

Process finished with exit code 0

7)

```

1 fun fibonacci(n: Int): List<Int> {
2     require(n >= 0) { "Количество чисел должно быть неотрицательным." }
3
4     val sequence = mutableListOf<Int>()
5
6     when {
7         n == 0 -> return emptyList()
8         n == 1 -> sequence.add(0)
9         else -> {
10            sequence.add(0)
11            sequence.add(1)
12
13            for (i in 2..n) {
14                val nextNumber = sequence[i - 1] + sequence[i - 2]
15                sequence.add(nextNumber)
16            }
17        }
18    }
19
20    return sequence
21 }
22
23 fun main() {
24     print("Введите количество чисел Фибоначчи: ")
25     val n = readLine()?.toIntOrNull() ?: 0
26     val result = fibonacci(n)
27     if (result.isEmpty()) {
28         print("Введите $n чисел Фибоначчи: ${result.joinToString(separator = ", ")}")
29     } else {
30         print("Ошибка: введено неверное значение.")
31     }
32 }

```

Введите количество чисел Фибоначчи: 14  
 Введите 14 чисел Фибоначчи: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229, 832040, 1346269, 2178309, 3524578

Process finished with exit code 0

8)

The screenshot shows the IntelliJ IDEA IDE with a Kotlin project named LABA. The project structure on the left includes a `src/main/kotlin` directory. The main code file contains a `gcd` function and a `main` function. The `gcd` function uses a `while` loop to calculate the greatest common divisor of two integers `a` and `b`. The `main` function prompts the user for two numbers, reads them, and prints the result of the `gcd` function.

```
1 import kotlin.math.abs
2
3
4 fun gcd(a: Int, b: Int): Int {
5     var g = abs(a)
6     var g = abs(b)
7
8     while (b != 0) {
9         val temp = g
10        g = a % b
11        a = temp
12    }
13
14    return g
15 }
16
17 fun main() {
18     print("Введите первое число: ")
19     val num1 = readLine()!!.toInt()
20     print("Введите второе число: ")
21     val num2 = readLine()!!.toInt()
22
23     val result = gcd(num1, num2)
24
25     println("Наибольшая общая делитель ($num1, $num2) = $result")
26 }
27
```

The Run console at the bottom shows the execution output:

```
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=63595:C:\Users\atypol\AppData\Local\Programs\IntelliJ IDEA Community Edit
Введите первое число: 2
Введите второе число: 6
Наибольшая общая делитель (2, 6) = 2
Process finished with exit code 0
```

9)

The screenshot shows the IntelliJ IDEA IDE with the same LABA project. The main code file contains a `main` function that prompts the user for a string, reads it, reverses it using `reversed()`, and prints the result.

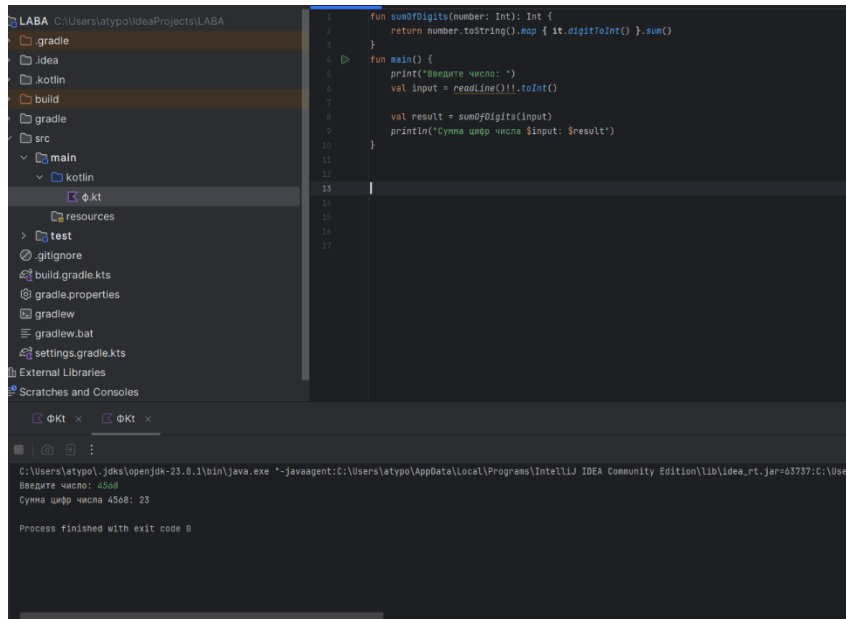
```
1 fun main() {
2     print("Введите строку: ")
3     val input = readLine()!!
4
5     val reversedInput = input.reversed()
6     println("Обратная строка: $reversedInput")
7 }
8
9
10
11
12
13
14
15
```

The Run console at the bottom shows the execution output:

```
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=63595:C:\Users\atypol\AppData\Local\Programs\IntelliJ IDEA Community Edit
Введите строку: /ruet/
Обратная строка: /trep/
Process finished with exit code 0
```

At the bottom of the IDE, the breadcrumb navigation shows the current path: `LABA > src > main > kotlin > Kotlin > main`.

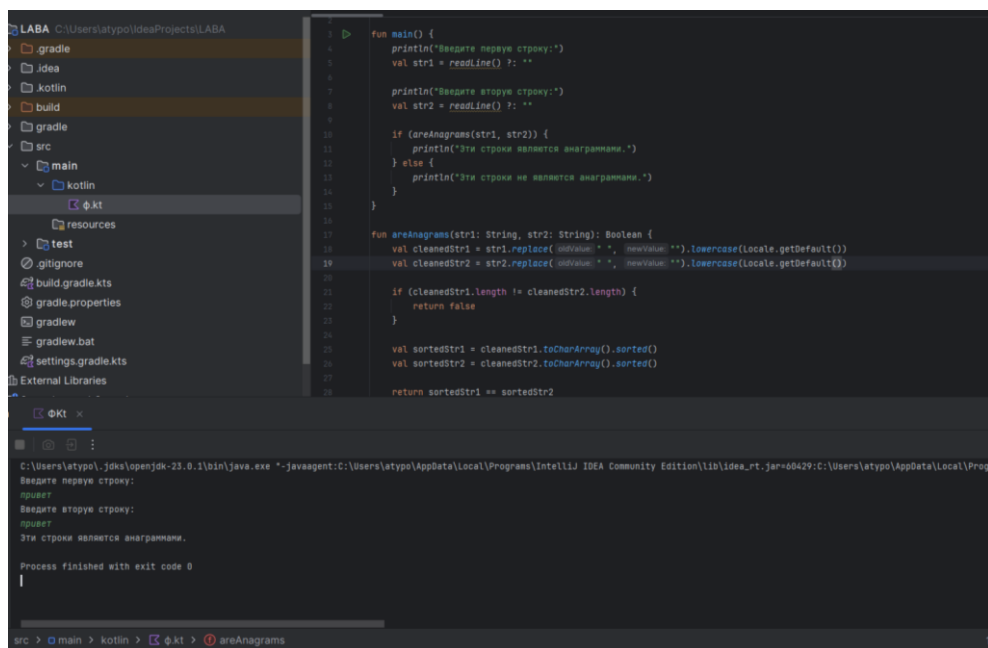
10)



```
1 fun sumOfDigits(number: Int): Int {
2     return number.toString().map { it.digitToInt() }.sum()
3 }
4
5 fun main() {
6     print("Введите число: ")
7     val input = readLine()!!.toInt()
8
9     val result = sumOfDigits(input)
10    println("Число $input равно $result")
11 }
12
13
14
15
16
17
```

Process finished with exit code 0

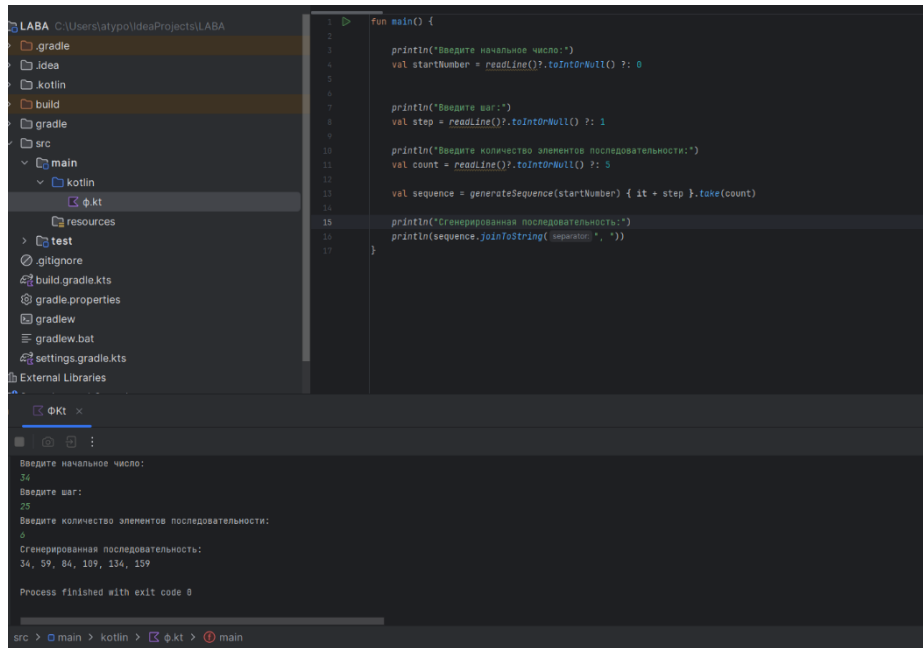
11)



```
1 fun main() {
2     println("Введите первую строку:")
3     val str1 = readLine() ?: ""
4
5     println("Введите вторую строку:")
6     val str2 = readLine() ?: ""
7
8     if (areAnagrams(str1, str2)) {
9         println("Эти строки являются анаграммами.")
10    } else {
11        println("Эти строки не являются анаграммами.")
12    }
13 }
14
15
16
17 fun areAnagrams(str1: String, str2: String): Boolean {
18     val cleanedStr1 = str1.replace(Regex("[^a-zA-Z]"), "").lowercase(Locale.getDefault())
19     val cleanedStr2 = str2.replace(Regex("[^a-zA-Z]"), "").lowercase(Locale.getDefault())
20
21     if (cleanedStr1.length != cleanedStr2.length) {
22         return false
23     }
24
25     val sortedStr1 = cleanedStr1.toCharArray().sorted()
26     val sortedStr2 = cleanedStr2.toCharArray().sorted()
27
28     return sortedStr1 == sortedStr2
29 }
```

Process finished with exit code 0

12)



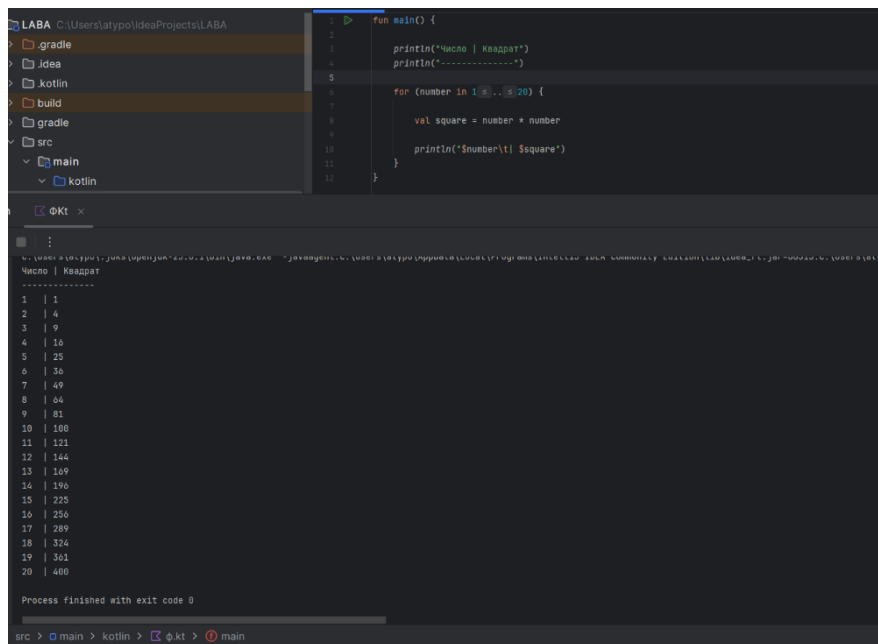
The screenshot shows an IDE with a project named LABA. The file explorer on the left shows the project structure, including a `src/main/kotlin` directory. The main editor displays the following Kotlin code:

```
1 fun main() {
2     println("Введите начальное число:")
3     val startNumber = readLine()?.toIntOrNull() ?: 0
4
5     println("Введите шаг:")
6     val step = readLine()?.toIntOrNull() ?: 1
7
8     println("Введите количество элементов последовательности:")
9     val count = readLine()?.toIntOrNull() ?: 5
10
11     val sequence = generateSequence(startNumber) { it + step }.take(count)
12
13     println("Сгенерированная последовательность:")
14     println(sequence.joinToString(separator: ", "))
15 }
16
```

The output window at the bottom shows the following text:

```
Введите начальное число:
34
Введите шаг:
25
Введите количество элементов последовательности:
5
Сгенерированная последовательность:
34, 59, 84, 109, 134, 159
Process finished with exit code 0
```

13)



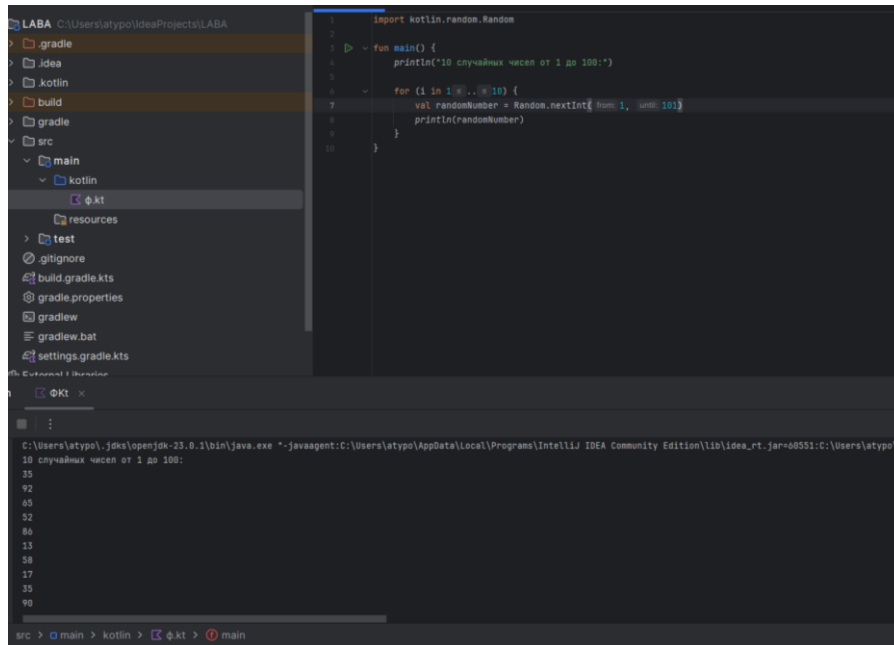
The screenshot shows an IDE with a project named LABA. The file explorer on the left shows the project structure, including a `src/main/kotlin` directory. The main editor displays the following Kotlin code:

```
1 fun main() {
2     println("Число | Квадрат")
3     println("-----")
4
5     for (number in 1..20) {
6         val square = number * number
7         println("$number | $square")
8     }
9 }
10
```

The output window at the bottom shows the following text:

```
Число | Квадрат
-----
1 | 1
2 | 4
3 | 9
4 | 16
5 | 25
6 | 36
7 | 49
8 | 64
9 | 81
10 | 100
11 | 121
12 | 144
13 | 169
14 | 196
15 | 225
16 | 256
17 | 289
18 | 324
19 | 361
20 | 400
Process finished with exit code 0
```

14)



The screenshot shows the IntelliJ IDEA IDE with a Kotlin project named 'LABA'. The file explorer on the left shows the project structure, including 'src/main/kotlin' and 'resources'. The main editor displays a Kotlin file with the following code:

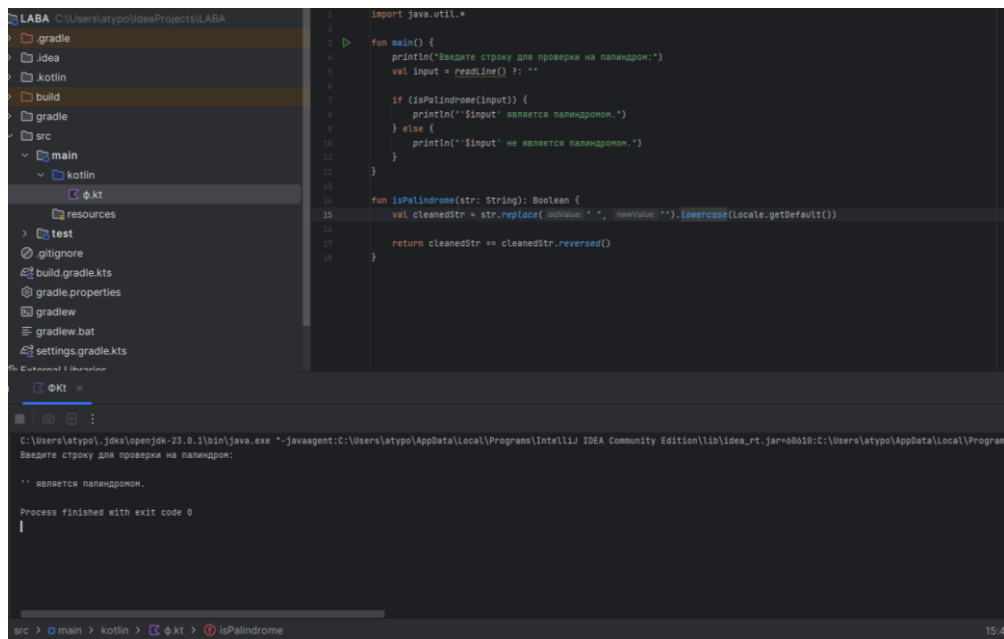
```
1 import kotlin.random.Random
2
3 fun main() {
4     println("10 случайных чисел от 1 до 100:")
5
6     for (i in 1..10) {
7         val randomNumber = Random.nextInt(from = 1, to = 101)
8         println(randomNumber)
9     }
10 }
```

The bottom panel shows the output of the program:

```
C:\Users\latypov\jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\latypov\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=40551:C:\Users\latypov\AppData\Local\Programs\IntelliJ IDEA Community Edition\bin\idea_rt.jar" 40551
10 случайных чисел от 1 до 100:
35
92
65
52
84
13
58
17
35
90
```

The status bar at the bottom indicates the current file is 'main' in the 'kotlin' package.

15)



The screenshot shows the IntelliJ IDEA IDE with a Kotlin project named 'LABA'. The file explorer on the left shows the project structure, including 'src/main/kotlin' and 'resources'. The main editor displays a Kotlin file with the following code:

```
1 import java.util.*
2
3 fun main() {
4     println("Введите строку для проверки на палиндром:")
5     val input = readLine() ?: ""
6
7     if (isPalindrome(input)) {
8         println("'input' является палиндромом.")
9     } else {
10        println("'input' не является палиндромом.")
11    }
12 }
13
14 fun isPalindrome(str: String): Boolean {
15     val cleanedStr = str.replace("\\s+", "").toLowerCase(Locale.getDefault())
16
17     return cleanedStr == cleanedStr.reversed()
18 }
```

The bottom panel shows the output of the program:

```
C:\Users\latypov\jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Users\latypov\AppData\Local\Programs\IntelliJ IDEA Community Edition\lib\idea_rt.jar=40610:C:\Users\latypov\AppData\Local\Programs\IntelliJ IDEA Community Edition\bin\idea_rt.jar" 40610
Введите строку для проверки на палиндром:
** является палиндромом.
Process finished with exit code 0
```

The status bar at the bottom indicates the current file is 'isPalindrome' in the 'kotlin' package.

16)

The screenshot shows the IntelliJ IDEA interface with a project named 'mobilPriloz'. The 'Project' view on the left shows the directory structure: 'src' > 'laba5' > '16z.kt'. The editor displays the following Kotlin code:

```
1 package laba5
2
3 fun sumOfSquares(n: Int): Int {
4     return (1..n).sumOf { it * it }
5 }
6
7 fun main() {
8     val n = 5
9     val result = sumOfSquares(n)
10    println("Сумма квадратов чисел от 1 до $n: $result")
11 }
```

The 'Run' view at the bottom shows the execution command and output:

```
"C:\Program Files\Microsoft\jdk-17.0.9-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\lib\idea_rt.jar-62310:C:\Program Files\JetBrains\I..."
Сумма квадратов чисел от 1 до 5: 55
Process finished with exit code 0
```

The status bar at the bottom indicates the file is '16z.kt' in the 'src > laba5' directory, with a cursor at line 10, column 43.

17)

The screenshot shows the IntelliJ IDEA interface with the same project 'mobilPriloz'. The 'Project' view shows the directory structure: 'src' > 'laba5' > '17z.kt'. The editor displays the following Kotlin code:

```
1 package laba5
2
3 fun printCharacters(input: String) {
4     for (char in input) {
5         println(char)
6     }
7 }
8
9 fun main() {
10    val text = "Hello, Kotlin!"
11    printCharacters(text)
12 }
```

The 'Run' view at the bottom shows the execution command and output:

```
"C:\Program Files\Microsoft\jdk-17.0.9-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\lib\idea_rt.jar-62310:C:\Program Files\JetBrains\I..."
H
e
l
l
o
,

K
o
t
l
i
n
!
Process finished with exit code 0
```

The status bar at the bottom indicates the file is '17z.kt' in the 'src > laba5' directory, with a cursor at line 12, column 2.



18)

```

package laba5

fun printStaircase(n: Int) {
    for (i in 1..n) {
        repeat(n - i + 1) {
            print("# ")
        }
        println()
    }
}

fun main() {
    println("Введите высоту лестницы (N):")
    val N = readln().toInt()
    printStaircase(N)
}

```

Run \_18z.kt x

Введите высоту лестницы (N):  
4

#####

Process finished with exit code 0

19)

```

package laba5

fun main() {
    val numbers = arrayOf(45, 23, 11, 99, 78, 67, 56)
    println("Исходный массив: ${numbers.joinToString(separator = ", ")}")

    for (i in 0..numbers.size - 1) {
        for (j in 0..numbers.size - i - 1) {
            if (numbers[j] > numbers[j + 1]) {
                val temp = numbers[j]
                numbers[j] = numbers[j + 1]
                numbers[j + 1] = temp
            }
        }
    }

    println("Отсортированный массив: ${numbers.joinToString(separator = ", ")}")
}

```

Run \_19z.kt x

"C:\Program Files\Microsoft\jdk-17.0.9-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\lib\idea\_rt.jar=62569:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\bin" -jar C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\bin\idea\_rt.jar 62569

Исходный массив: 45, 23, 11, 99, 78, 67, 56

Отсортированный массив: 11, 23, 45, 56, 67, 78, 99

Process finished with exit code 0

20)

```

fun main() {
    println("Введите начальное значение диапазона:")
    val start = readLine()!!.toInt()
    println("Введите конечное значение диапазона:")
    val end = readLine()!!.toInt()

    val primes = findPrimesInRange(start, end)
    println("Простые числа в диапазоне от $start до $end:")
    println(primes.joinToString(", "))
}

fun findPrimesInRange(start: Int, end: Int): List<Int> {
    if (start < 2) {
        println("Начальное значение должно быть больше или равно 2.")
        return emptyList()
    }
}

```

```

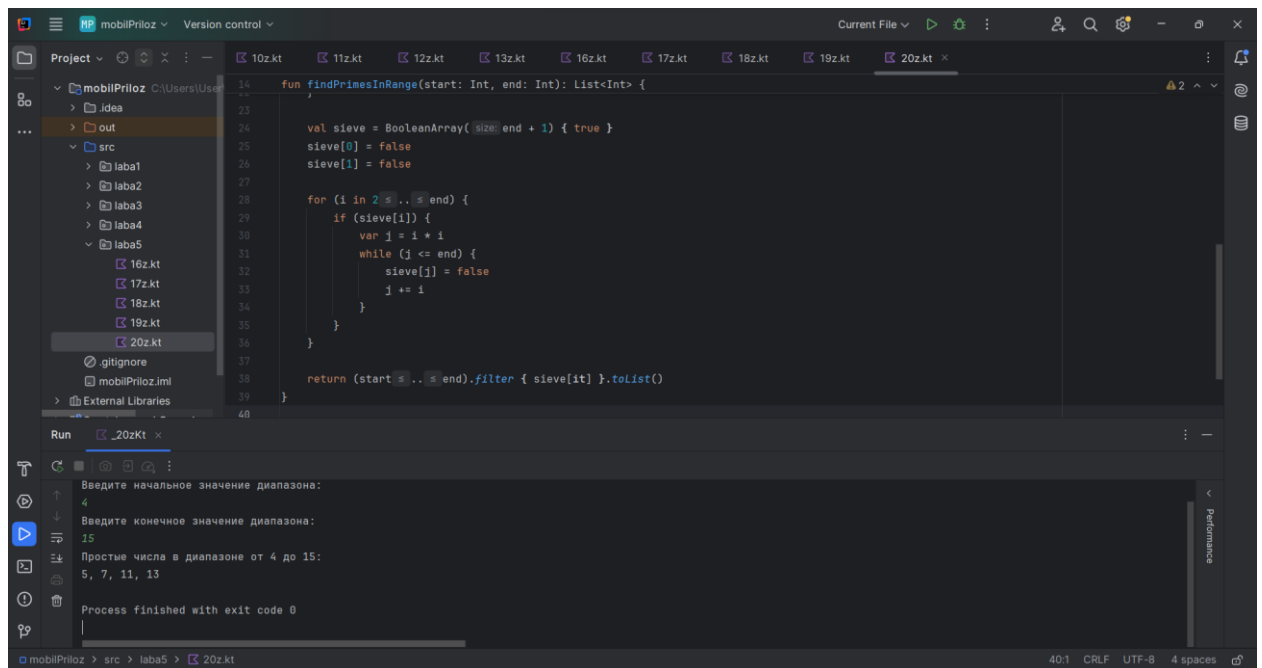
    if (end < start) {
        println("Конечное значение должно быть больше или равно начальному.")
        return emptyList()
    }

    val sieve = BooleanArray(end + 1) { true }
    sieve[0] = false
    sieve[1] = false

    for (i in 2..end) {
        if (sieve[i]) {
            var j = i * i
            while (j <= end) {
                sieve[j] = false
                j += i
            }
        }
    }

    return (start..end).filter { sieve[it] }.toList()
}

```



```

21) import java.util.Calendar

fun main() {
    println("Введите год:")
    val year = readLine()!!.toInt()
    println("Введите месяц (1-12):")
    val month = readLine()!!.toInt()

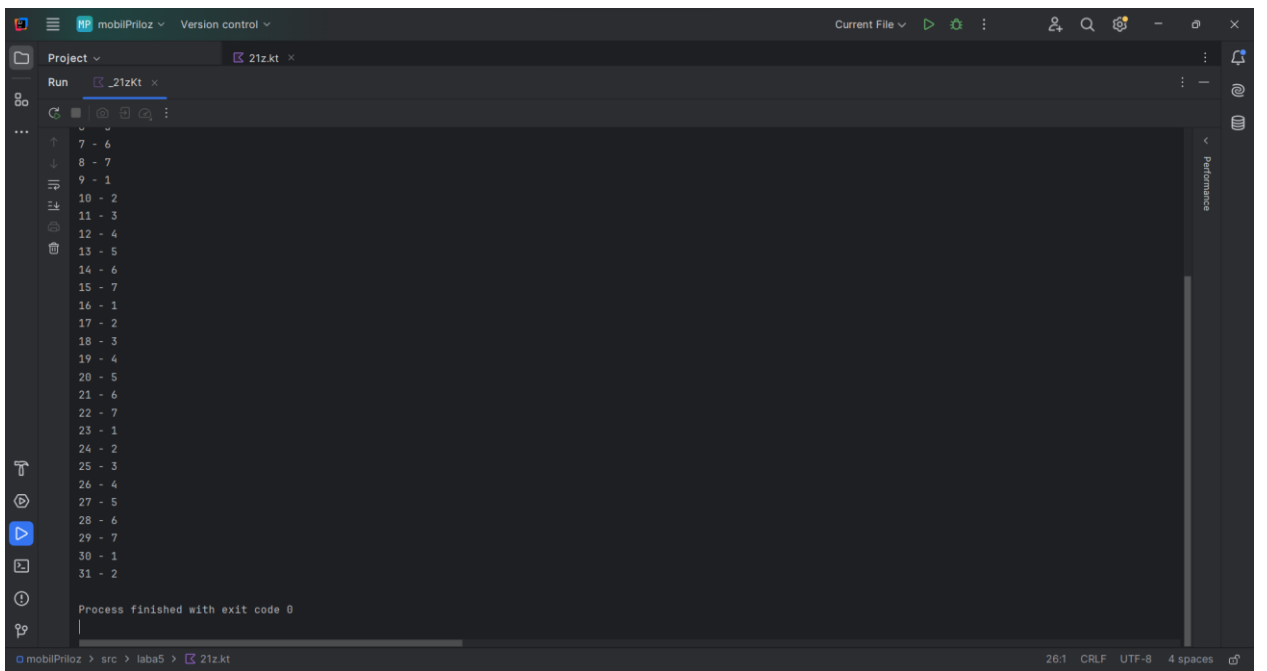
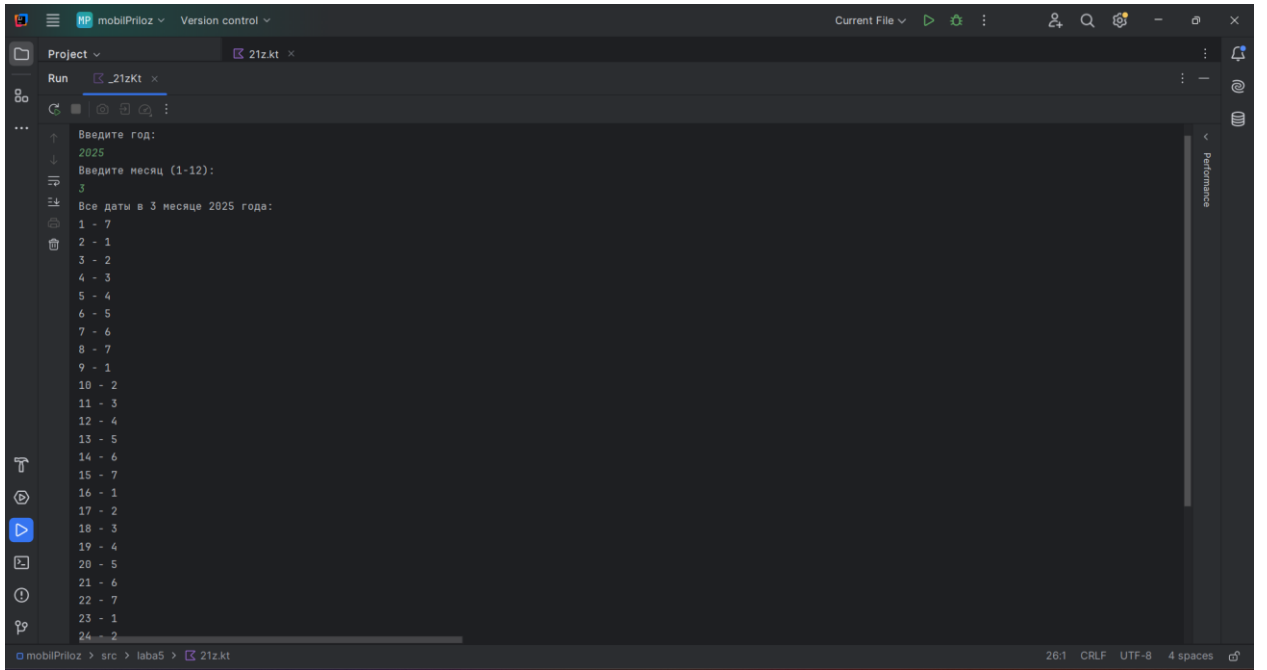
    val calendar = Calendar.getInstance()
    calendar.set(Calendar.YEAR, year)
    calendar.set(Calendar.MONTH, month - 1) // Месяцы в Calendar начинаются с
0
    calendar.set(Calendar.DAY_OF_MONTH, 1)

    val daysInMonth = calendar.getActualMaximum(Calendar.DAY_OF_MONTH)

    println("Все даты в $month месяце $year года:")
    for (day in 1..daysInMonth) {

```

```
calendar.set(Calendar.DAY_OF_MONTH, day)
val dayOfWeek = calendar.get(Calendar.DAY_OF_WEEK)
val dayOfMonth = calendar.get(Calendar.DAY_OF_MONTH)
println("$dayOfMonth - $dayOfWeek")
}
}
```



22)

```

package laba5
import java.util.Random

fun main() {
    val random = Random()
    val targetNumber = random.nextInt(100) + 1 // Генерация случайного числа от 1 до 100
    var guess: Int
    var attempts = 0

    println("Добро пожаловать в игру 'Угадай число!'")
    println("Попробуйте угадать число от 1 до 100.")

    do {
        println("Введите ваше предположение:")
        guess = readLine()!!.toInt()
        attempts++

        if (guess < targetNumber) {
            println("Ваше число меньше. Попробуйте еще раз.")
        } else if (guess > targetNumber) {
            println("Ваше число больше. Попробуйте еще раз.")
        }
    } while (guess != targetNumber)

    println("Поздравляем! Вы угадали число $targetNumber за $attempts попыток.")
}

```

```

"C:\Program Files\Microsoft\jdk-17.0.9-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\lib\idea_rt.jar=63324:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\bin" -Didea.config.path=C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\config -Didea.system.path=C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\lib -Didea.version=2024.3.2
Добро пожаловать в игру 'Угадай число!'
Попробуйте угадать число от 1 до 100.
Введите ваше предположение:
50
Ваше число больше. Попробуйте еще раз.
Введите ваше предположение:
30
Ваше число больше. Попробуйте еще раз.
Введите ваше предположение:
10
Ваше число меньше. Попробуйте еще раз.
Введите ваше предположение:
20
Ваше число больше. Попробуйте еще раз.
Введите ваше предположение:
14
Ваше число больше. Попробуйте еще раз.
Введите ваше предположение:
11
Ваше число меньше. Попробуйте еще раз.
Введите ваше предположение:
12
Ваше число меньше. Попробуйте еще раз.
Введите ваше предположение:
13
Поздравляем! Вы угадали число 13 за 8 попыток.
Process finished with exit code 0

```

```

23) fun main() {
    var continueOperation = true

    while (continueOperation) {
        println("Введите первую цифру:")
        val firstNumber = readLine()!!.toInt()

        println("Введите вторую цифру:")
        val secondNumber = readLine()!!.toInt()

        println("Выберите операцию: сложение или умножение?")
        val operation = readLine()!!.lowercase()

        when (operation) {
            "сложение" -> {
                val result = firstNumber + secondNumber
                println("Результат сложения: $result")
            }
        }
    }
}

```

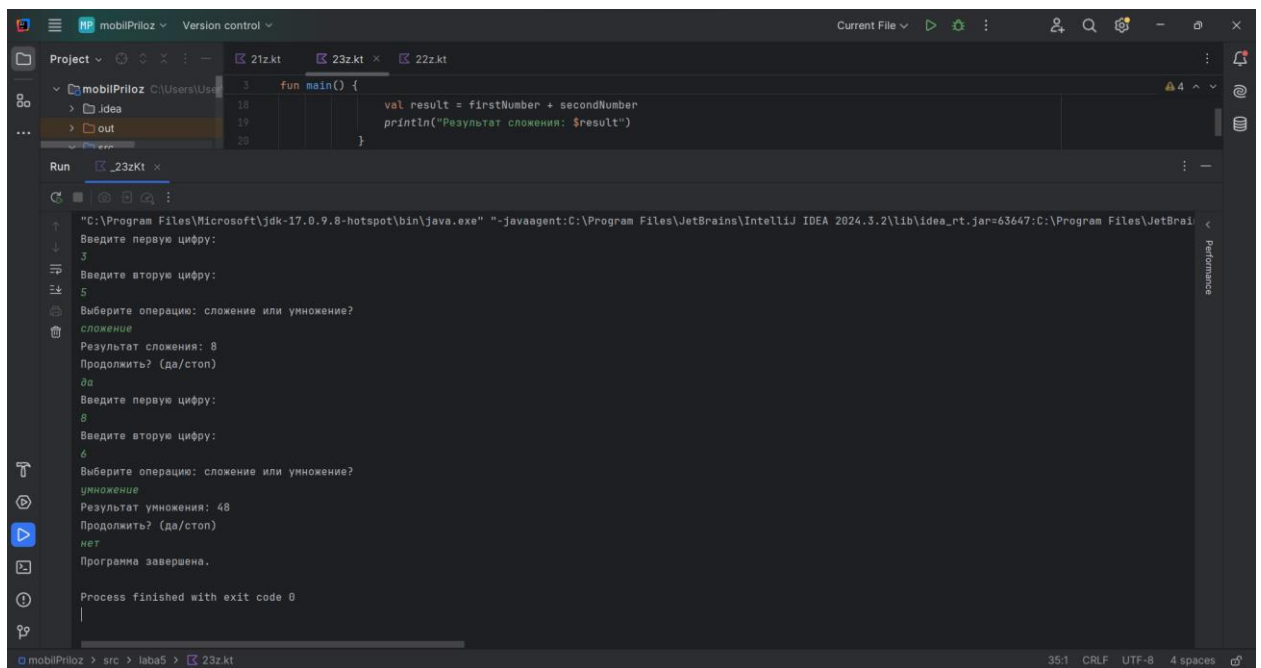
```

    }
    "умножение" -> {
        val result = firstNumber * secondNumber
        println("Результат умножения: $result")
    }
    else -> println("Неверная операция. Попробуйте снова.")
}

println("Продолжить? (да/стоп)")
val input = readLine()!!.lowercase()
continueOperation = input == "да"
}

println("Программа завершена.")
}

```



```

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

    println("Исходная матрица:")
    printMatrix(matrix)

    val transposedMatrix = transposeMatrix(matrix)

    println("\nТранспонированная матрица:")
    printMatrix(transposedMatrix)
}

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

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

```

```

    }

    return transposedMatrix
}

fun printMatrix(matrix: Array<IntArray>) {
    for (row in matrix) {
        for (element in row) {
            print("$element ")
        }
        println()
    }
}

```

The screenshot shows the IntelliJ IDEA interface with the Run window open. The Run configuration is named "24z.kt". The output of the program is as follows:

```

"C:\Program Files\Microsoft\jdk-17.0.9-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\lib\idea_rt.jar=63868:C:\Program Files\JetBrains\I...
Исходная матрица:
1 2 3
4 5 6
7 8 9

Транспонированная матрица:
1 4 7
2 5 8
3 6 9

Process finished with exit code 0

```

25)

The screenshot shows the IntelliJ IDEA interface with the Run window open. The Run configuration is named "25z.kt". The output of the program is as follows:

```

package laba5

fun main() {
    println("Кубы чисел от 1 до 10:")
    for (i in 1..10) {
        val cube = i * i * i
        println("$i^3 = $cube")
    }
}

```

```

Кубы чисел от 1 до 10:
1^3 = 1
2^3 = 8
3^3 = 27
4^3 = 64
5^3 = 125
6^3 = 216
7^3 = 343
8^3 = 512
9^3 = 729
10^3 = 1000

Process finished with exit code 0

```

26)

```

1 package laba5
2
3 fun main() {
4     println("Введите число N:")
5     val n = readLine()!!.toInt()
6
7     var sumEven = 0
8     var sumOdd = 0
9
10    for (i in 1..n) {
11        if (i % 2 == 0) {
12            sumEven += i
13        } else {
14            sumOdd += i
15        }
16    }
17
18    println("Сумма четных чисел от 1 до $n: $sumEven")
19    println("Сумма нечетных чисел от 1 до $n: $sumOdd")
20 }
21

```

Run console output:

```

Введите число N:
8
Сумма четных чисел от 1 до 8: 20
Сумма нечетных чисел от 1 до 8: 16
Process finished with exit code 0

```

27)

```

1 package laba5
2
3 fun main() {
4     println("Введите число N:")
5     val n = readLine()!!.toInt()
6
7     for (i in 1..n) {
8         for (j in 1..i) {
9             print(j)
10        }
11        println()
12    }
13 }
14

```

Run console output:

```

Введите число N:
8
1
12
123
1234
12345
123456
1234567
12345678
Process finished with exit code 0

```

28)

The screenshot shows the IntelliJ IDEA IDE with a project named 'mobilPriloz'. The 'Project' view on the left shows a directory structure with 'src' containing 'laba5' and several Kotlin files. The '28z.kt' file is open in the editor, showing the following Kotlin code:

```
1 package laba5
2
3 fun main() {
4     println("Введите количество чисел N:")
5     val n = readLine()!!.toInt()
6
7     println("Введите $n чисел:")
8     val numbers = mutableListOf<Int>()
9     repeat(n) {
10         numbers.add(readLine()!!.toInt())
11     }
12
13     numbers.sort()
14
15     println("Сортированные числа:")
16     println(numbers.joinToString(separator: ", "))
17 }
18
```

The 'Run' view at the bottom shows the execution of the program. The user input is 4, and the output is the sorted list of numbers: 3, 12, 32, 45.

```
Введите количество чисел N:
4
Введите 4 чисел:
3
45
32
12
Сортированные числа:
3, 12, 32, 45
```

29)

The screenshot shows the IntelliJ IDEA IDE with the same project 'mobilPriloz'. The '29z.kt' file is open in the editor, showing the following Kotlin code:

```
1 package laba5
2
3 fun main() {
4     println("Введите число N:")
5     val n = readLine()!!.toInt()
6
7     var sum = 0.0
8
9     for (i in 1..n) {
10         sum += 1.0 / i
11     }
12
13     println("Сумма ряда 1 + 1/2 + 1/3 + ... + 1/$n = $sum")
14 }
15
```

The 'Run' view at the bottom shows the execution of the program. The user input is 5, and the output is the sum of the series: 2.2833333333333333.

```
"C:\Program Files\Microsoft\jdk-17.0.9-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\lib\idea_rt.jar=65026:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\bin" -jar C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\bin\idea_rt.jar 65026
Введите число N:
5
Сумма ряда 1 + 1/2 + 1/3 + ... + 1/5 = 2.2833333333333333
Process finished with exit code 0
```



30)

The screenshot displays the IntelliJ IDEA IDE interface. The top toolbar includes icons for file operations, search, and settings. The 'Project' sidebar on the left lists files from 17z.kt to 30z.kt, with 30z.kt selected. The main editor shows the following Kotlin code:

```
1 package laba5
2
3 fun main() {
4     println("Введите целое число:")
5     val number = readLine()!!.toInt()
6
7     val binaryRepresentation = number.toString(radix: 2)
8
9     println("Двоичное представление числа $number: $binaryRepresentation")
10 }
11
```

Below the editor, the 'Run' tab is active, showing the execution output:

```
"C:\Program Files\Microsoft\jdk-17.0.9-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\lib\idea_rt.jar=65095:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\bin" -Didea.config.path=C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\conf -Didea.copyright.notification=false -Didea.log.path=C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\log -Didea.platform.prefix=IntelliJ -jar C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\bin\idea_rt.jar 65095
Введите целое число:
12
Двоичное представление числа 12: 1100
Process finished with exit code 0
```

The status bar at the bottom indicates the file path 'mobilPriloz > src > laba5 > 30z.kt' and encoding details '11:1 CRLF UTF-8 4 spaces'.