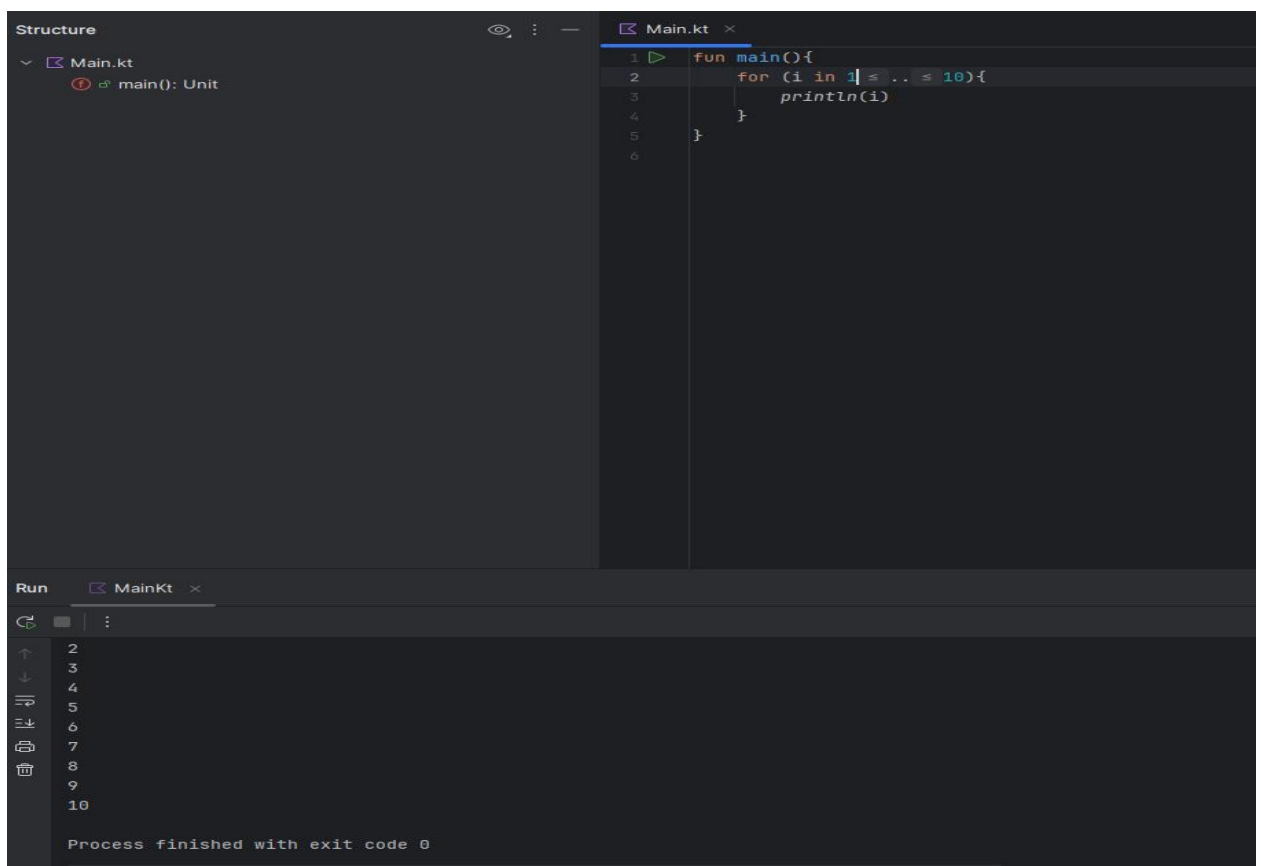


## Практическая работа № 5.

Выполнили: Андрухова и Загородняя.

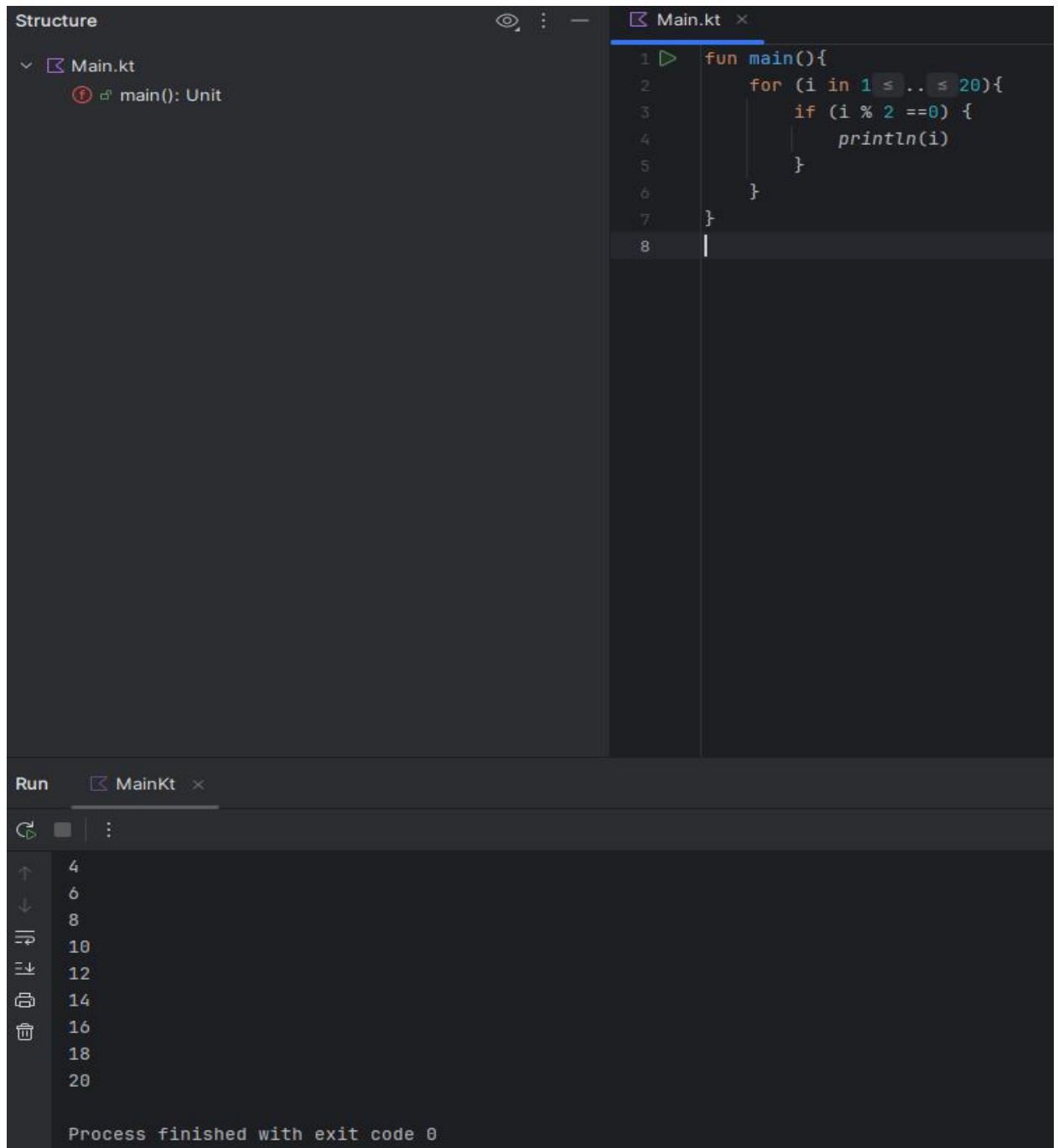
```
1. fun main(){  
    for (i in 1..10){  
        println(i)  
    }  
}
```



```
2. fun main(){  
    for (i in 1..20){  
        if (i % 2 == 0) {  
            println(i)  
        }  
    }  
}
```

}

}



3. fun main(){

print("Введите число N: ")

val n = readLine()?.toIntOrNull() ?: return

var sum = 0

```

for (i in 1..n){

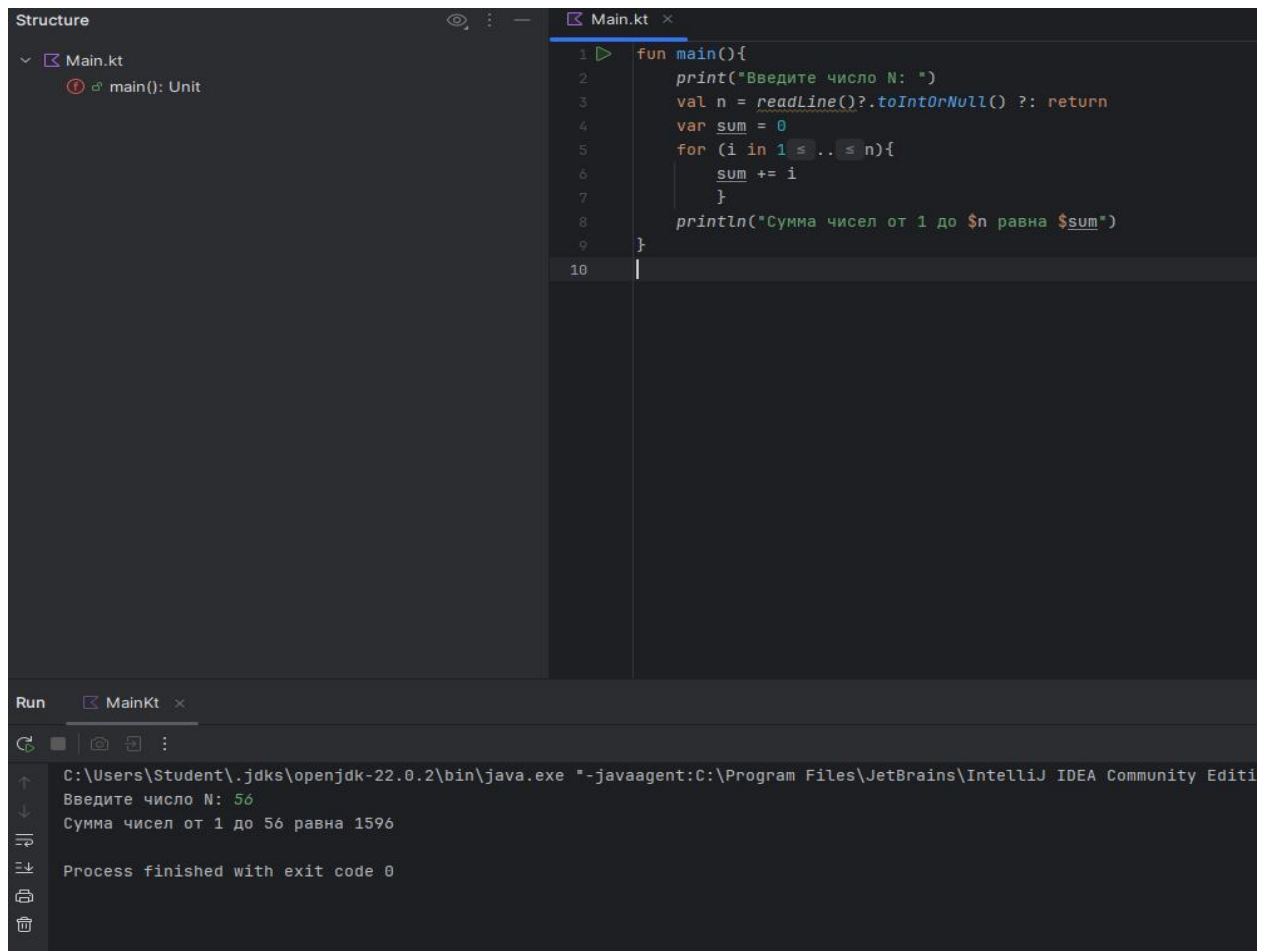
    sum += i

}

println("Сумма чисел от 1 до $n равна $sum")

}

```



```

4. fun main(){

    print("Введите число: ")

    val number = readLine()?.toIntOrNull() ?: return

    var factorial= 1L

    for (i in 1..number){

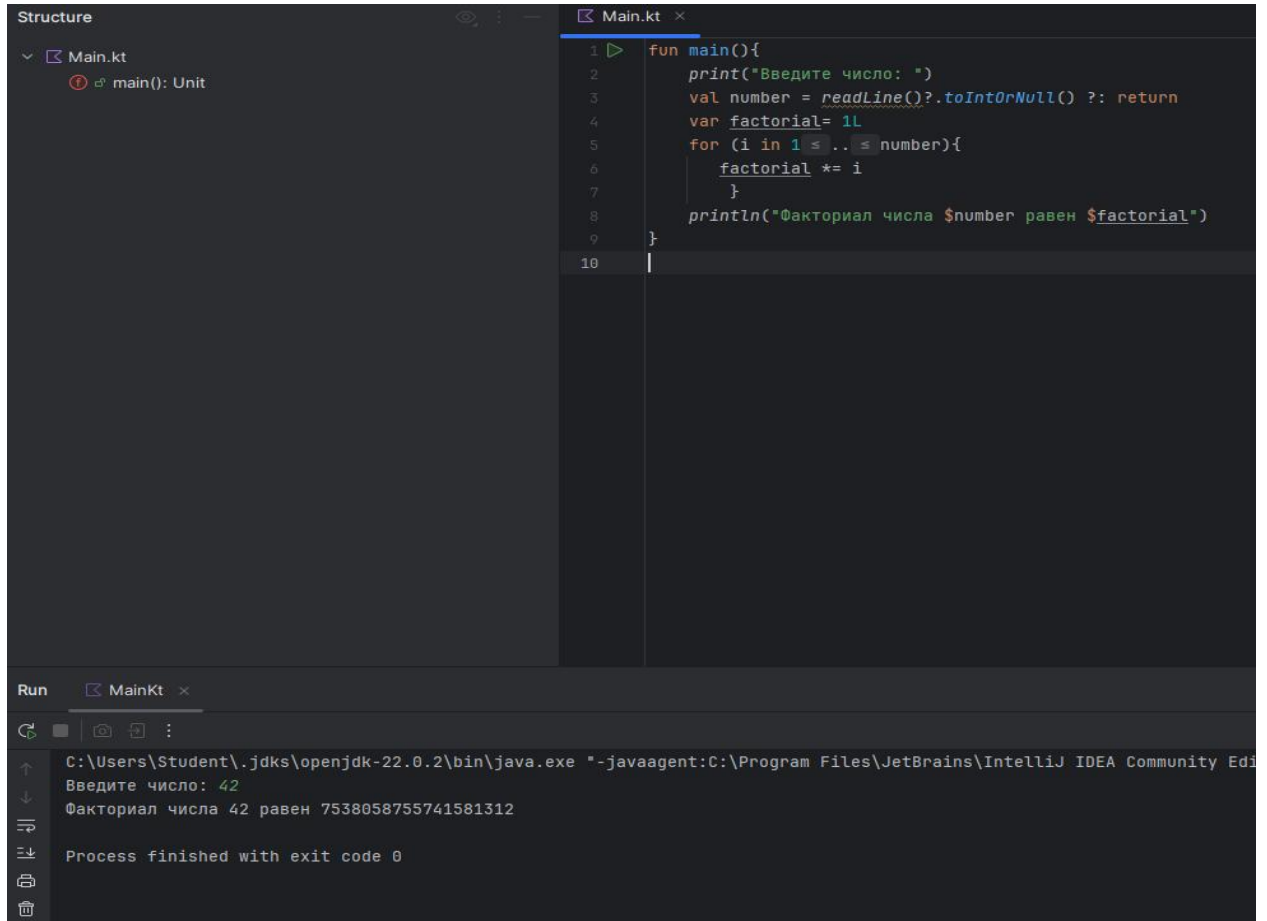
        factorial *= i
    }
}

```

}

println("Факториал числа \$number равен \$factorial")

}



5. fun isPrime (number: Int): Boolean {

if (number <= 1) return false

if (number <= 3) return true

if (number % 2 == 0 || number % 3 == 0) return false

var i = 5

while (i \* i <= number) {

if (number % i == 0 || number % (i+2) == 0) return false

i += 6

```

    }

    return true

}

fun main() {

    print("Введите число: ")

    val number =

        readLine()?. toIntOrNull()?: return

    if (isPrime(number)) {

        println("$number - простое число.")

    } else {

        println("$number - составное число." )

    }

}

```

The screenshot displays the IntelliJ IDEA IDE with a Kotlin file named `Main.kt`. The code defines a function `isPrime` that checks if a number is prime and a `main` function that prompts the user for input and prints the result.

**Structure View:**

- Main.kt
  - isPrime(Int): Boolean
  - main(): Unit

**Main.kt Code:**

```

1 fun isPrime (number: Int): Boolean {
2     if (number <= 1) return false
3     if (number <= 3) return true
4     if (number % 2 == 0 || number % 3 == 0) return false
5     var i = 5
6     while (i * i <= number) {
7         if (number % i == 0 || number % (i+2) == 0) return false
8         i += 6
9     }
10    return true
11 }
12
13 fun main() {
14     print("Введите число: ")
15     val number =
16         readLine()?. toIntOrNull()?: return
17     if (isPrime(number)) {
18         println("$number - простое число.")
19     } else {
20         println("$number - составное число." )
21     }
22 }

```

**Run View:**

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe --javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2
Введите число: 55
55 - составное число.

Process finished with exit code 0

```

```
6. fun main(){  
    for (i in 1..10){  
        for (j in 1..10){  
            print("${i * j}\t")  
        }  
        println()  
    }  
}
```

The screenshot shows an IDE with a dark theme. On the left, the 'Structure' panel shows a file named 'Main.kt' containing a function 'main(): Unit'. The main editor displays the following Kotlin code:

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

Below the editor, the 'Run' panel shows the command executed: `C:\Users\Student\jdk\openjdk-22.0.2\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea_rt.jar -Didea.launcher.debug -Didea.launcher.debug=CONSOLE -jar C:\Users\Student\IdeaProjects\Kotlin\MainKt.jar`. The output is a 10x10 grid of numbers, representing the multiplication table from 1 to 10.

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

```

7. fun fibonacci(n: Int): List<Int> {

    val sequence = mutableListOf(1,1)

    for (i in 2 until n){

        sequence.add(sequence[i - 1] + sequence[i - 2])

    }

    return sequence.take(n).toList()

}

fun main() {

    print("введите количество чисел Фибоначчи: ")

    val n = readLine()?.toIntOrNull() ?: return

    val fibSequence = fibonacci(n)

    println(fibSequence.joinToString(","))

}

```

The screenshot displays the IntelliJ IDEA IDE with a Kotlin file named `Main.kt`. The left sidebar shows the `Structure` view with two members: `fibonacci(Int): List<Int>` and `main(): Unit`. The main editor shows the following code:

```

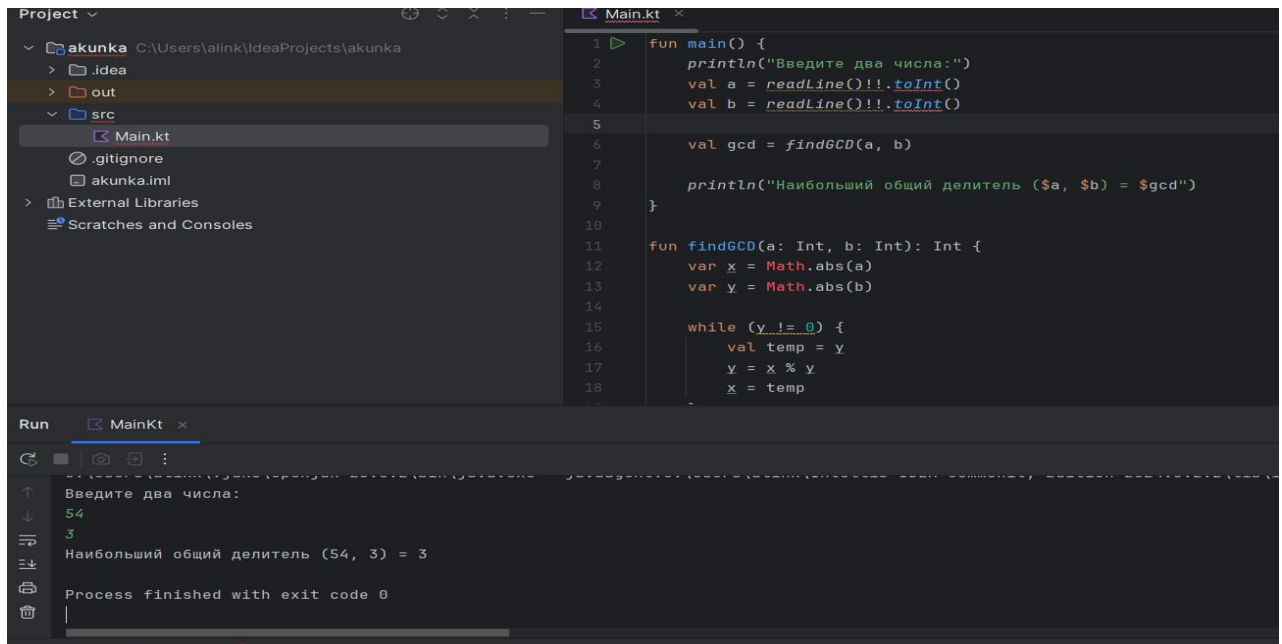
1 fun fibonacci(n: Int): List<Int> {
2     val sequence = mutableListOf(1,1)
3     for (i in 2 until n){
4         sequence.add(sequence[i - 1] + sequence[i - 2])
5     }
6     return sequence.take(n).toList()
7 }
8
9 fun main() {
10    print("введите количество чисел Фибоначчи: ")
11    val n = readLine()?.toIntOrNull() ?: return
12    val fibSequence = fibonacci(n)
13    println(fibSequence.joinToString(","))
14 }

```

At the bottom, the `Run` tab shows the execution of `MainKt`. The command line is `C:\Users\Student\jdk\openjdk-22.0.2\bin\java.exe --javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 202`. The output shows the prompt "введите количество чисел Фибоначчи: " followed by the input "6", and the resulting Fibonacci sequence "1,1,2,3,5,8". The process finished with exit code 0.

```
8.fun main() {  
  
    println("Введите два числа:")  
  
    val a = readLine()!!.toInt()  
  
    val b = readLine()!!.toInt()  
  
    val gcd = findGCD(a, b)  
  
    println("Наибольший общий делитель ($a, $b) = $gcd")  
  
}  
  
fun findGCD(a: Int, b: Int): Int {  
  
    var x = Math.abs(a)  
  
    var y = Math.abs(b)  
  
    while (y != 0) {  
  
        val temp = y  
  
        y = x % y  
  
        x = temp  
  
    }  
  
    return x  
  
}
```





```
9.fun main() {

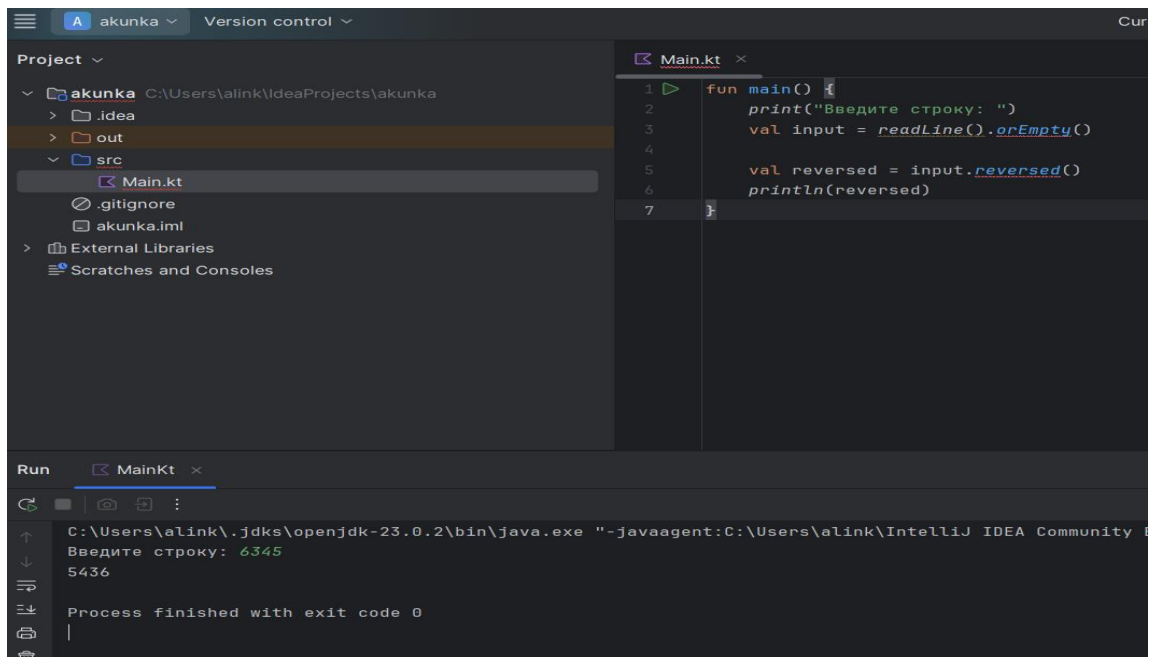
    print("Введите строку: ")

    val input = readLine().orEmpty()

    val reversed = input.reversed()

    println(reversed)

}
```



```

10.fun digitSum(number: Int): Int {

    var num = number

    var sum = 0

    while (num > 0) {

        sum += num % 10

        num /= 10

    }

    return sum

}

fun main() {

    print("Введите число: ")

    val number = readLine()?.toIntOrNull() ?: return

    val sum = digitSum(number)

    println("Сумма цифр числа $number равна $sum")

}

```

The screenshot displays the IntelliJ IDEA IDE. On the left, the 'Project' view shows the file structure of a project named 'akunka', with 'Main.kt' selected under the 'src' directory. The main editor window shows the Kotlin code for 'Main.kt', which implements a function 'digitSum' to calculate the sum of digits of a number and a 'main' function to interact with the user. The code is as follows:

```

1 fun digitSum(number: Int): Int {
2
3     while (num > 0) {
4         sum += num % 10
5         num /= 10
6     }
7
8     return sum
9
10 }
11
12
13 fun main() {
14     print("Введите число: ")
15     val number = readLine()?.toIntOrNull() ?: return
16
17     val sum = digitSum(number)
18     println("Сумма цифр числа $number равна $sum")
19 }

```

At the bottom, the 'Run' console shows the execution of the program. It prompts the user to enter a number, and the user has entered '2689'. The program outputs 'Сумма цифр числа 2689 равна 25' and then indicates that the process finished with exit code 0.

```

1 fun areAnagrams(s1: String, s2: String): Boolean {

    if (s1.length != s2.length) return false

    val charCounts = IntArray(256)

    for (c in s1) {

        charCounts[c.code]++

    }

    for (c in s2) {

        charCounts[c.code]--

    }

    for (count in charCounts) {

        if (count != 0) return false

    }

    return true

}

fun main() {

    print("Введите первую строку: ")

    val s1 = readLine()!!

    print("Введите вторую строку: ")

    val s2 = readLine()!!

    if (areAnagrams(s1, s2)) {

        println("Строки '$s1' и '$s2' являются анаграммами.")

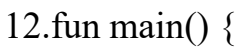
    } else {

        println("Строки '$s1' и '$s2' не являются анаграммами.")

    }

}

```

$$\}$$


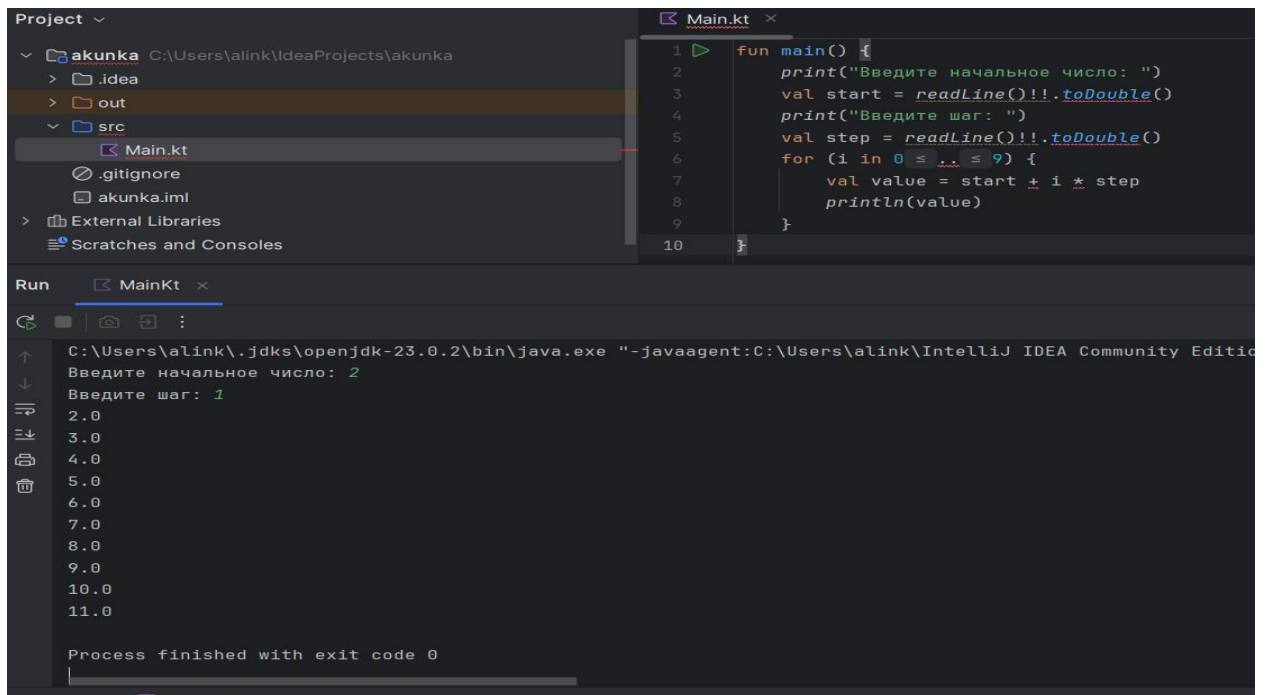
```
val start = readLine()!!.toDouble()
```

```
val step = readLine()!!.toDouble()
```

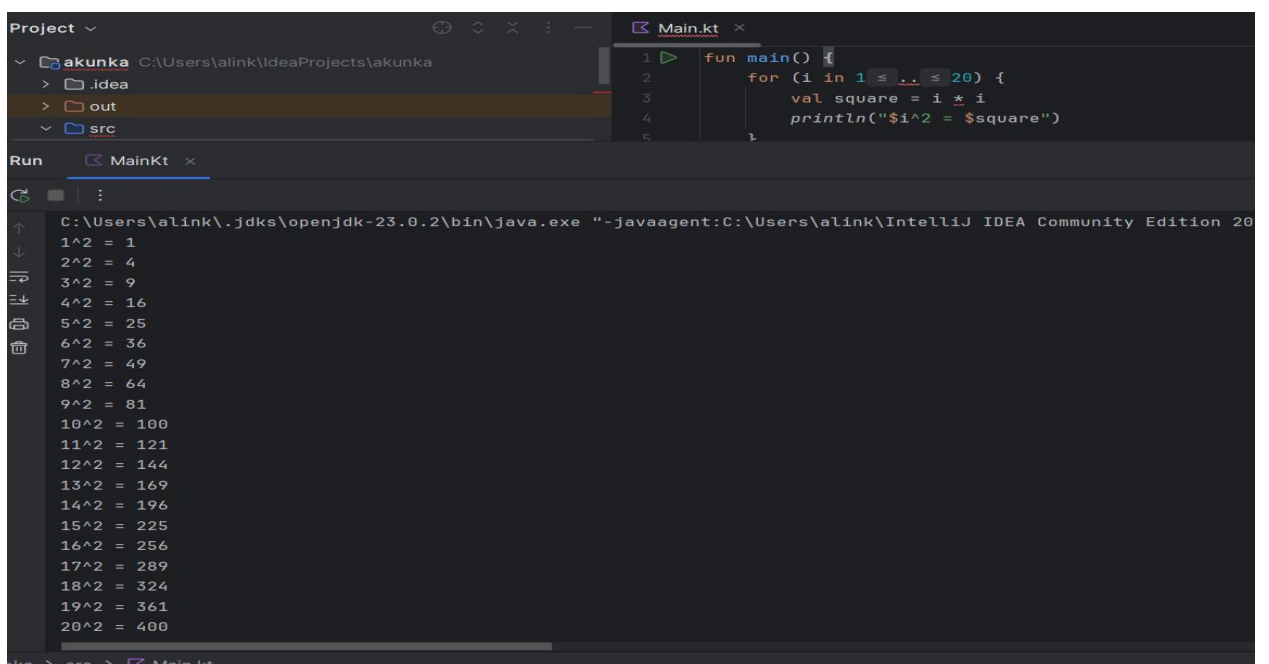
```
val value = start + i * step
```

$$\}$$

}



```
13.fun main() {  
  
    for (i in 1..20) {  
  
        val square = i * i  
  
        println("$i^2 = $square")  
  
    }  
  
}
```



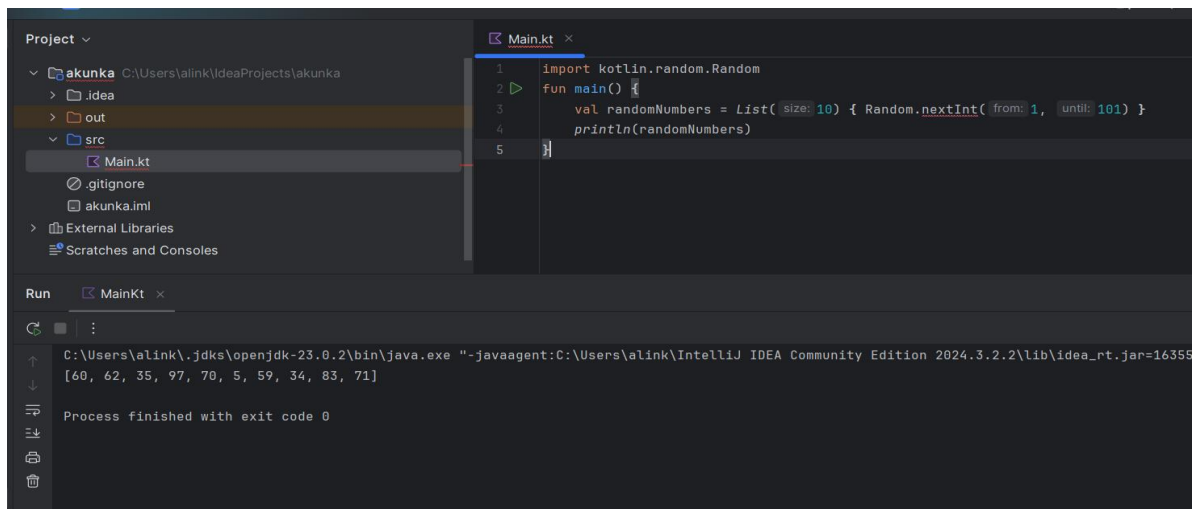
```
14.import kotlin.random.Random
```

```
fun main() {
```

```
    val randomNumbers = List(10) { Random.nextInt(1, 101) }
```

```
    println(randomNumbers)
```

```
}
```



```
15.fun isPalindrome(s: String): Boolean {
```

```
    val cleanedS = s.filter { it.isLetterOrDigit() }.lowercase()
```

```
    return cleanedS == cleanedS.reversed()
```

```
}
```

```
fun main() {
```

```
    print("Введите строку: ")
```

```
    val input = readLine()!!
```

```
    if (isPalindrome(input)) {
```

```
        println("$input' - палиндром.")
```

```

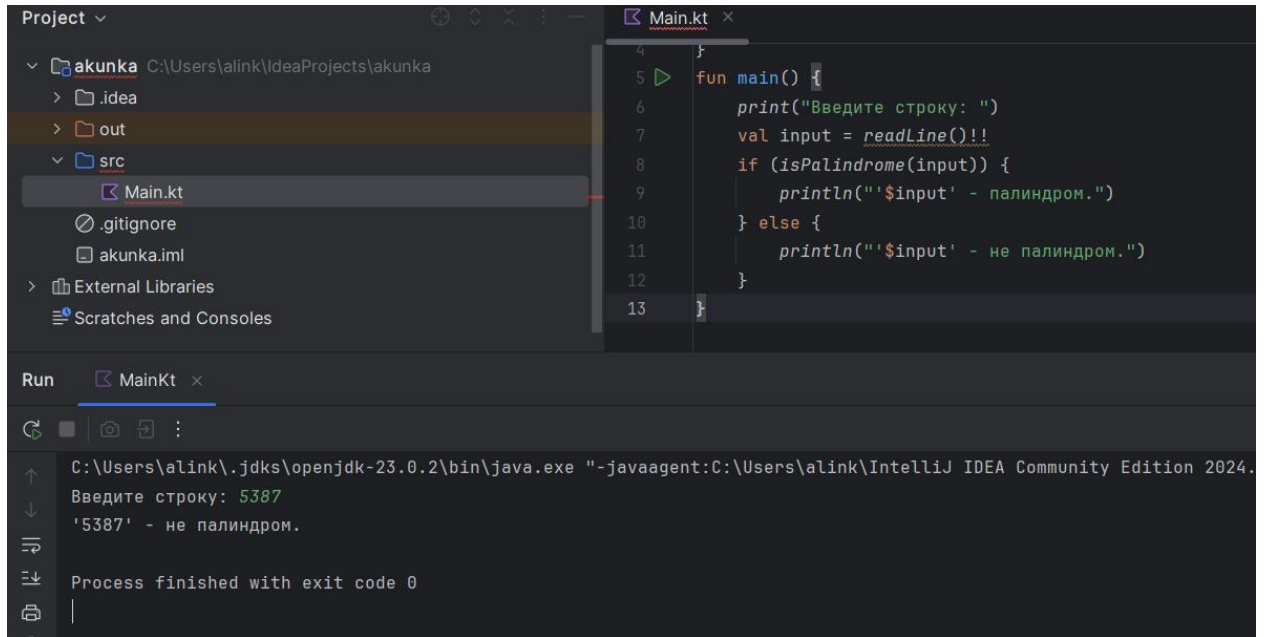
    } else {

        println("$input' - не палиндром.")

    }

}

```



16.fun sigmaSquares(N: Int): Long {

var sum = 0L

for (i in 1..N) {

sum += (i \* i).toLong()

}

return sum

}

fun main() {

print("Введите число N: ")

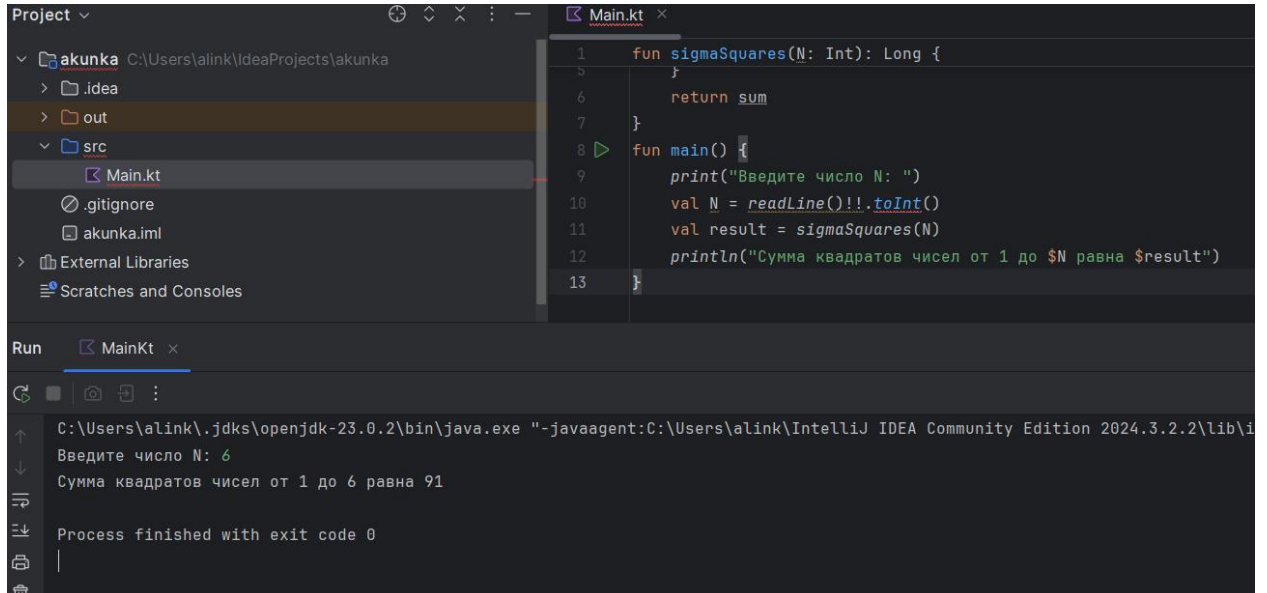
val N = readLine()!!.toInt()

```

        val result = sigmaSquares(N)

        println("Сумма квадратов чисел от 1 до $N равна $result")
    }
}

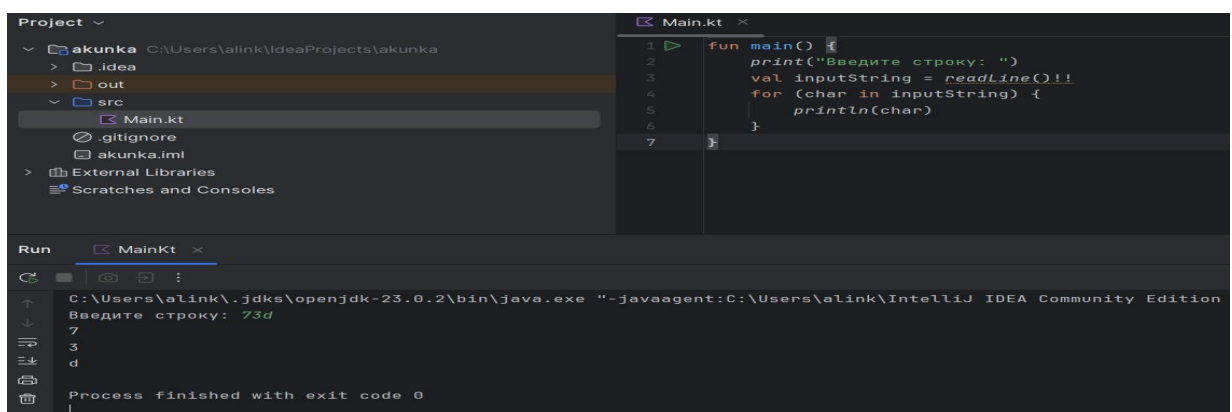
```



```

17.fun main() {
    print("Введите строку: ")
    val inputString = readLine()!!
    for (char in inputString) {
        println(char)
    }
}

```





```

18.fun main() {

    print("Введите высоту лестницы: ")

    val height = readLine()!!.toInt()

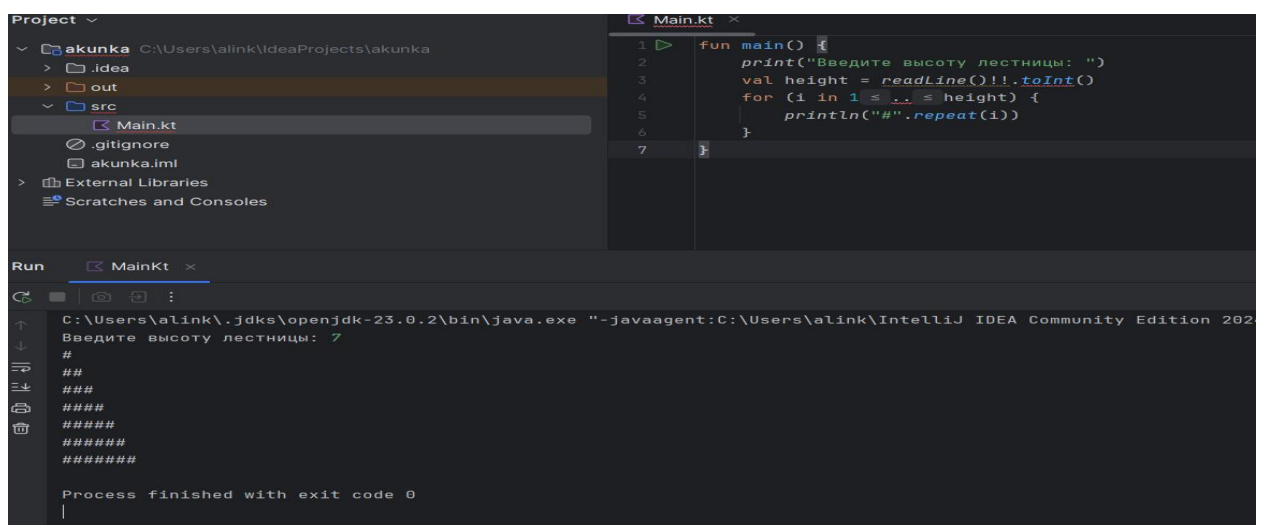
    for (i in 1..height) {

        println("#".repeat(i))

    }

}

```



```

19.fun main() {

    print("Введите количество элементов массива: ")

    val size = readLine()!!.toInt()

    val array = Array(size) { readLine()!!.toInt() }

    for (i in 0 until size - 1) {

        for (j in i + 1 until size) {

            if (array[i] > array[j]) {

                val temp = array[i]

                array[i] = array[j]

```

```

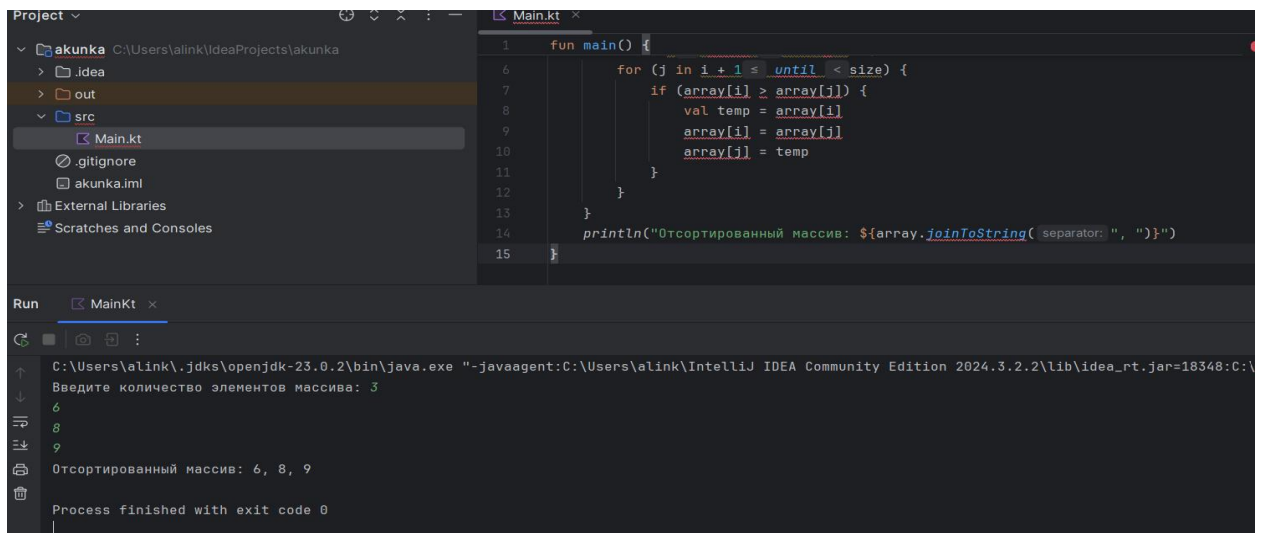
        array[j] = temp
    }

}

}

println("Отсортированный массив: ${array.joinToString(", ")}")
}

```



20. fun isPrime(n: Int): Boolean {

if (n <= 1) return false

if (n == 2 || n == 3) return true

if (n % 2 == 0 || n % 3 == 0) return false

var i = 5

while (i \* i <= n) {

if (n % i == 0 || n % (i + 2) == 0) return false

i += 6

}

```

        return true
    }

fun main() {

    print("Введите начальное число диапазона: ")

    val start = readLine()!!.toInt()

    print("Введите конечное число диапазона: ")

    val end = readLine()!!.toInt()

    println("Простые числа в диапазоне от $start до $end:")

    for (num in start..end) {

        if (isPrime(num)) {

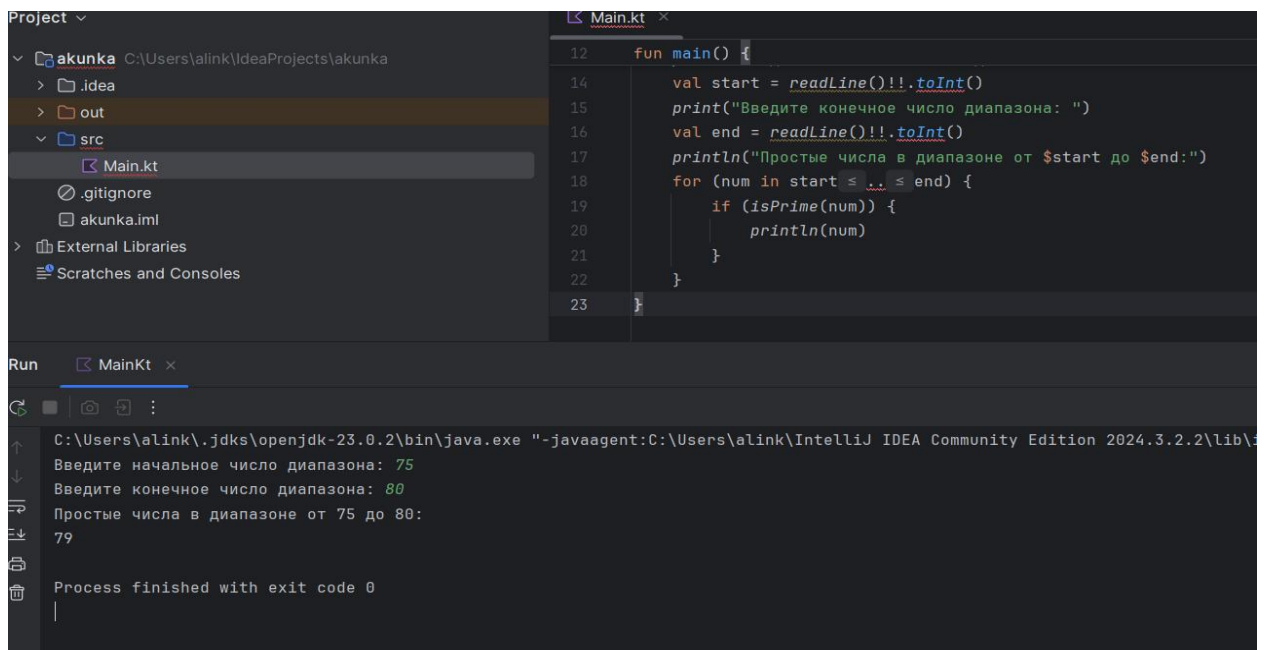
            println(num)

        }

    }

}

```



```
21.import java.time.LocalDate
```

```
import java.time.YearMonth
```

```
import java.util.Scanner
```

```
fun main() {
```

```
    val scanner = Scanner(System.`in`)
```

```
    println("Введите год (например, 2025):")
```

```
    val year = scanner.nextInt()
```

```
    println("Введите месяц (1-12):")
```

```
    val month = scanner.nextInt()
```

```
    if (month in 1..12) {
```

```
        val yearMonth = YearMonth.of(year, month)
```

```
        val daysInMonth = yearMonth.lengthOfMonth()
```

```
        println("Даты в $month/$year:")
```

```
        for (day in 1..daysInMonth) {
```

```
            val date = LocalDate.of(year, month, day)
```

```
            println(date)
```

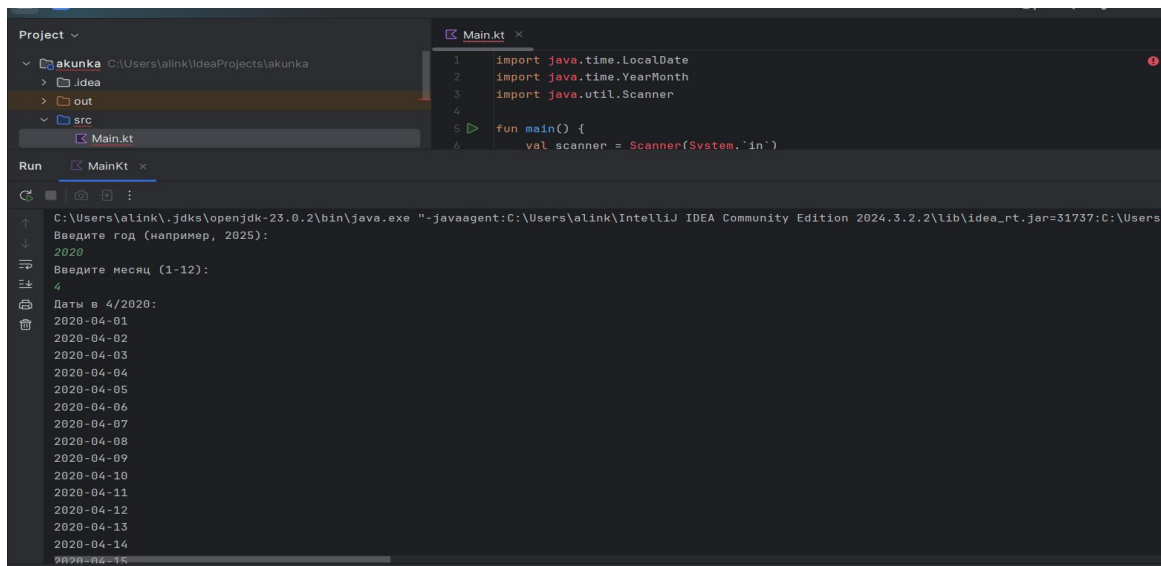
```
        }
```

```
    } else {
```

```
        println("Некорректный месяц. Пожалуйста, введите число от 1 до 12.")
```

```
    }
```

```
}
```



```
22.import kotlin.random.Random
```

```
import java.util.Scanner
```

```
fun main() {
```

```
    val scanner = Scanner(System.`in`)
```

```
    val randomNumber = Random.nextInt(1, 101) // Случайное число от 1 до 100
```

```
    var guess: Int? = null
```

```
    var attempts = 0
```

```
    println("Угадайте число от 1 до 100!")
```

```
    while (guess != randomNumber) {
```

```
        println("Введите ваше предположение:")
```

```
        guess = scanner.nextInt()
```

```
        attempts++
```

```
        when {
```

```
            guess < randomNumber -> println("Слишком низко! Попробуйте еще раз.")
```

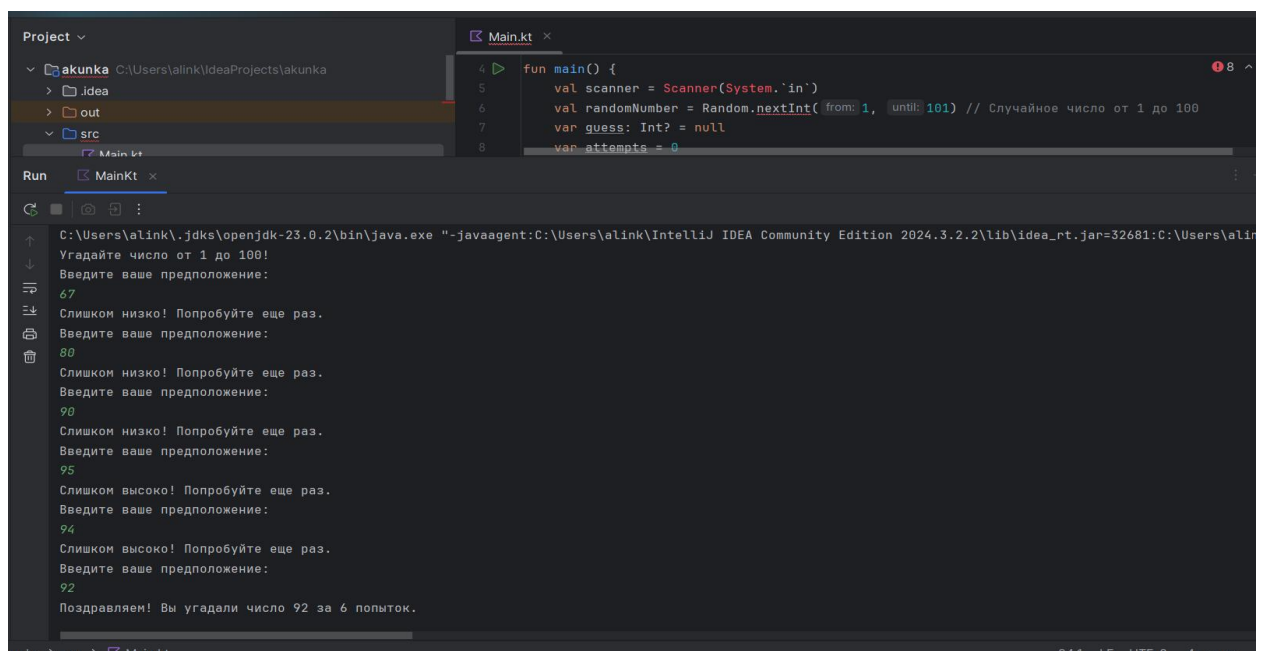
```
guess > randomNumber -> println("Слишком высоко! Попробуйте еще раз.")
```

```
else -> println("Поздравляем! Вы угадали число $randomNumber за $attempts попыток.")
```

```
}
```

```
}
```

```
}
```



```
23.import java.util.Scanner
```

```
fun main() {
```

```
    val scanner = Scanner(System.`in`)
```

```
    while (true) {
```

```
        println("Введите две цифры (или введите 'стоп' для выхода):")
```

```
        val input1 = scanner.nextLine()
```

```
        if (input1 == "стоп") break
```

```
        val input2 = scanner.nextLine()
```

```
if (input2 == "стоп") break

try {

    val number1 = input1.toDouble()

    val number2 = input2.toDouble()

    println("Введите операцию (сложение или умножение):")

    val operation = scanner.nextLine()

    when (operation) {

        "сложение" -> println("Результат сложения: ${number1 + number2}")

        "умножение" -> println("Результат умножения: ${number1 *
number2}")

        else -> println("Неизвестная операция. Пожалуйста, введите
'сложение' или 'умножение'.")

    }

} catch (e: NumberFormatException) {

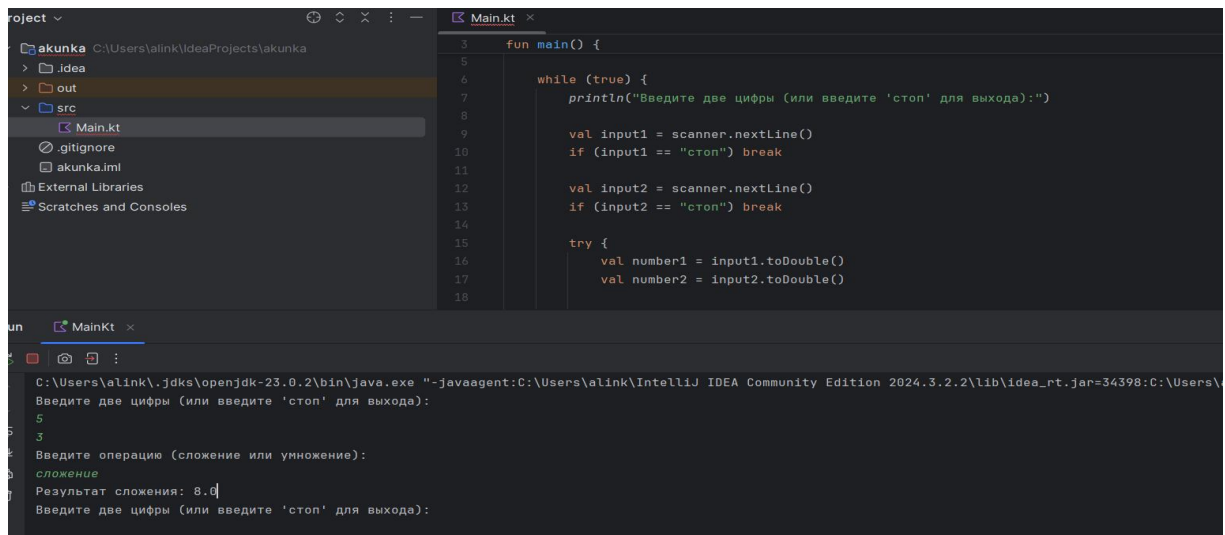
    println("Пожалуйста, введите корректные цифры.")

}

}

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("Транспонированная матрица:")

printMatrix(transposedMatrix)

}

fun transposeMatrix(matrix: Array<IntArray>): Array<IntArray> {

val rows = matrix.size

val cols = matrix[0].size

val transposed = Array(cols) { IntArray(rows) }



```

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

    return transposed
}

fun printMatrix(matrix: Array<IntArray>) {
    for (row in matrix) {
        println(row.joinToString(" "))
    }
}

```

The screenshot displays the IntelliJ IDEA IDE with a project named 'akunka'. The 'src' directory contains a file 'Main.kt'. The code in 'Main.kt' defines two functions: 'transposeMatrix' and 'printMatrix'. The 'transposeMatrix' function takes a 2D array of integers and returns its transpose. The 'printMatrix' function prints a 2D array of integers, with elements separated by spaces. The 'Run' button is visible, and the output console shows the execution results.

```

Project
└─ akunka C:\Users\alink\IdeaProjects\akunka
   └─ src
      └─ Main.kt
         .gitignore
         akunka.iml
   External Libraries
   Scratches and Consoles

Run MainKt x
C:\Users\alink\.jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Users\alink\IntelliJ IDEA Community Edition 2024.3.2.2\lib\idea_rt.jar=5574
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.fun main() {

    println("Кубы чисел от 1 до 10:")

    for (i in 1..10) {

        val cube = i * i * i

        println("$i^3 = $cube")

    }

}

```

```

Project
akunka C:\Users\alink\IdeaProjects\akunka
  .idea
  out
  src
    Main.kt
  .gitignore
  akunka.iml
Main.kt
1 fun main() {
2     println("Кубы чисел от 1 до 10:")
3     for (i in 1..10) {
4         val cube = i * i * i
5         println("$i^3 = $cube")
6     }
7 }
8

Run MainKt
C:\Users\alink\jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Users\alink\IntelliJ IDEA Community Edition 2024.3.2.2\lib\idea_rt.jar=56196:C:
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.fun main() {

    println("Введите число N:")

    val N = readLine()?.toIntOrNull()

    if (N != null && N > 0) {

        var sumEven = 0

        var sumOdd = 0

        for (i in 1..N) {

            if (i % 2 == 0) {

```

```

        sumEven += i

    } else {

        sumOdd += i

    }

}

println("Сумма четных чисел от 1 до $N: $sumEven")

println("Сумма нечетных чисел от 1 до $N: $sumOdd")

} else {

    println("Пожалуйста, введите положительное целое число.")

}

}

```

The screenshot displays the IntelliJ IDEA IDE. The left sidebar shows the project structure for 'akunka', with the 'src' directory expanded and 'Main.kt' selected. The main editor window shows the following Kotlin code in 'Main.kt':

```

1 fun main() {
2     println("Введите число N:")
3     val N = readLine()?.toIntOrNull()
4
5     if (N != null && N >= 0) {
6         var sumEven = 0
7         var sumOdd = 0
8
9         for (i in 1..N) {
10            if (i % 2 == 0) {
11                sumEven += i
12            } else {
13                sumOdd += i
14            }
15        }
16    }
17 }

```

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

```

C:\Users\alink\jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Users\alink\IntelliJ IDEA Community Edition 2024.3.2.2\lib\i
Введите число N:
6
Сумма четных чисел от 1 до 6: 12
Сумма нечетных чисел от 1 до 6: 9
Process finished with exit code 0

```

```

27.fun main() {

    println("Введите число N:")

    val N = readLine()?.toIntOrNull()

    if (N != null && N > 0) {

```

```

    for (i in 1..N) {

        for (j in 1..(N - i)) {

            print(" ")

        }

        for (k in 1..i) {

            print("$k ")

        }

        println()

    }

} else {

    println("Пожалуйста, введите положительное целое число.")

}

}

```

The screenshot displays the IntelliJ IDEA IDE. The top-left pane shows the project structure for 'akunka', with 'Main.kt' selected under the 'src' directory. The top-right pane shows the code of 'Main.kt', which implements a function to print a triangular pattern of numbers. The bottom pane shows the 'Run' output, where the program prompts for a number 'N' (8) and prints the corresponding triangular pattern of numbers from 1 to 8.

```

1 fun main() {
11     // Печатаем числа от 1 до i
12     for (k in 1..i) {
13         print("$k ")
14     }
15     // Переход на следующую строку
16     println()
17 }
18 } else {
19     println("Пожалуйста, введите положительное целое число.")
20 }
21 }
22

```

Run MainKt x

```

C:\Users\alink\jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Users\alink\IntelliJ IDEA Community Edition 2024.3.2.2\lib\idea_rt.jar=6658
Введите число N:
8
    1
   1 2
  1 2 3
 1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
Process finished with exit code 0

```

```

28.fun main() {

    println("Введите количество чисел (N):")

    val N = readLine()?.toIntOrNull()

    if (N != null && N > 0) {

        val numbers = mutableListOf<Int>()

        println("Введите $N чисел:")

        for (i in 1..N) {

            val number = readLine()?.toIntOrNull()

            if (number != null) {

                numbers.add(number)

            } else {

                println("Некорректный ввод. Пожалуйста, введите целое число.")

                return

            }

        }

        numbers.sort()

        println("Числа в порядке возрастания:")

        for (num in numbers) {

            print("$num ")

        }

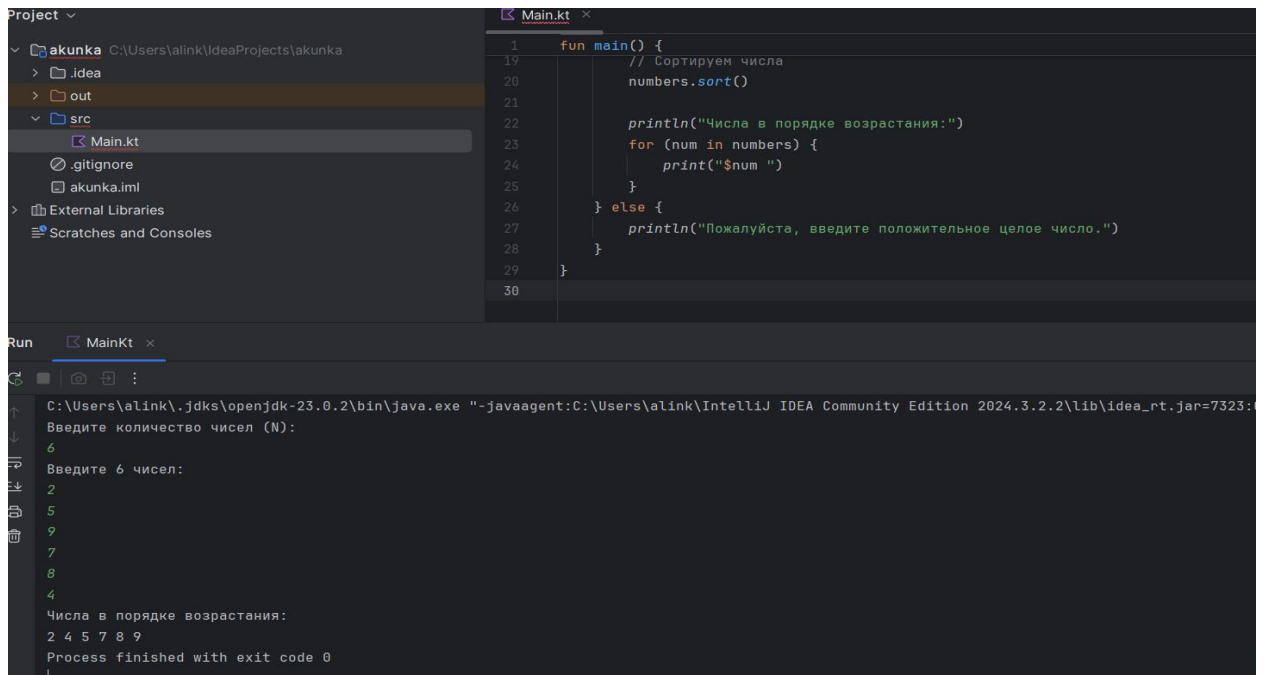
    } else {

        println("Пожалуйста, введите положительное целое число.")

    }
}

```

}



29.fun main() {

println("Введите значение N:")

val N = readLine()?.toIntOrNull()

if (N != null && N > 0) {

var sum = 0.0

for (i in 1..N) {

sum += 1.0 / i

}

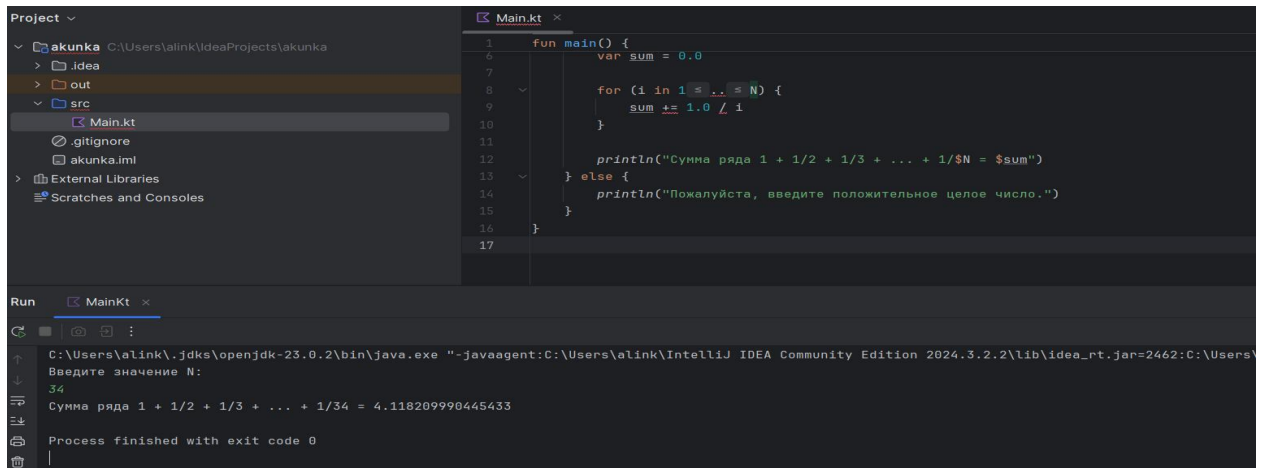
println("Сумма ряда  $1 + 1/2 + 1/3 + \dots + 1/N = \$sum$ ")

} else {

println("Пожалуйста, введите положительное целое число.")

}

}



```
30.fun main() {
```

```
    println("Введите целое число:")
```

```
    val number = readLine()?.toIntOrNull()
```

```
    if (number != null) {
```

```
        val binaryString = number.toString(2)
```

```
        println("Число $number в двоичной системе: $binaryString")
```

```
    } else {
```

```
        println("Пожалуйста, введите корректное целое число.")
```

```
    }
```

```
}
```

