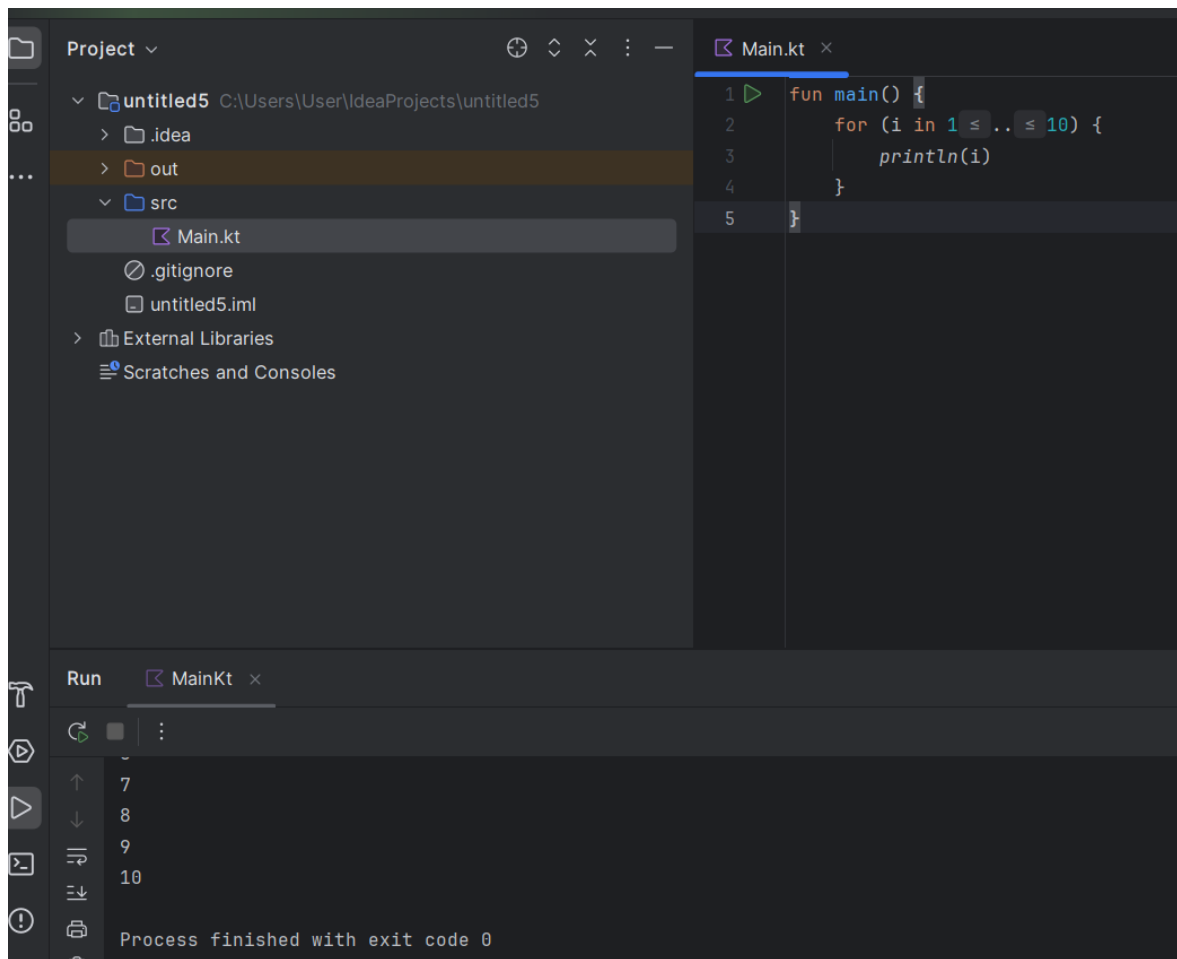
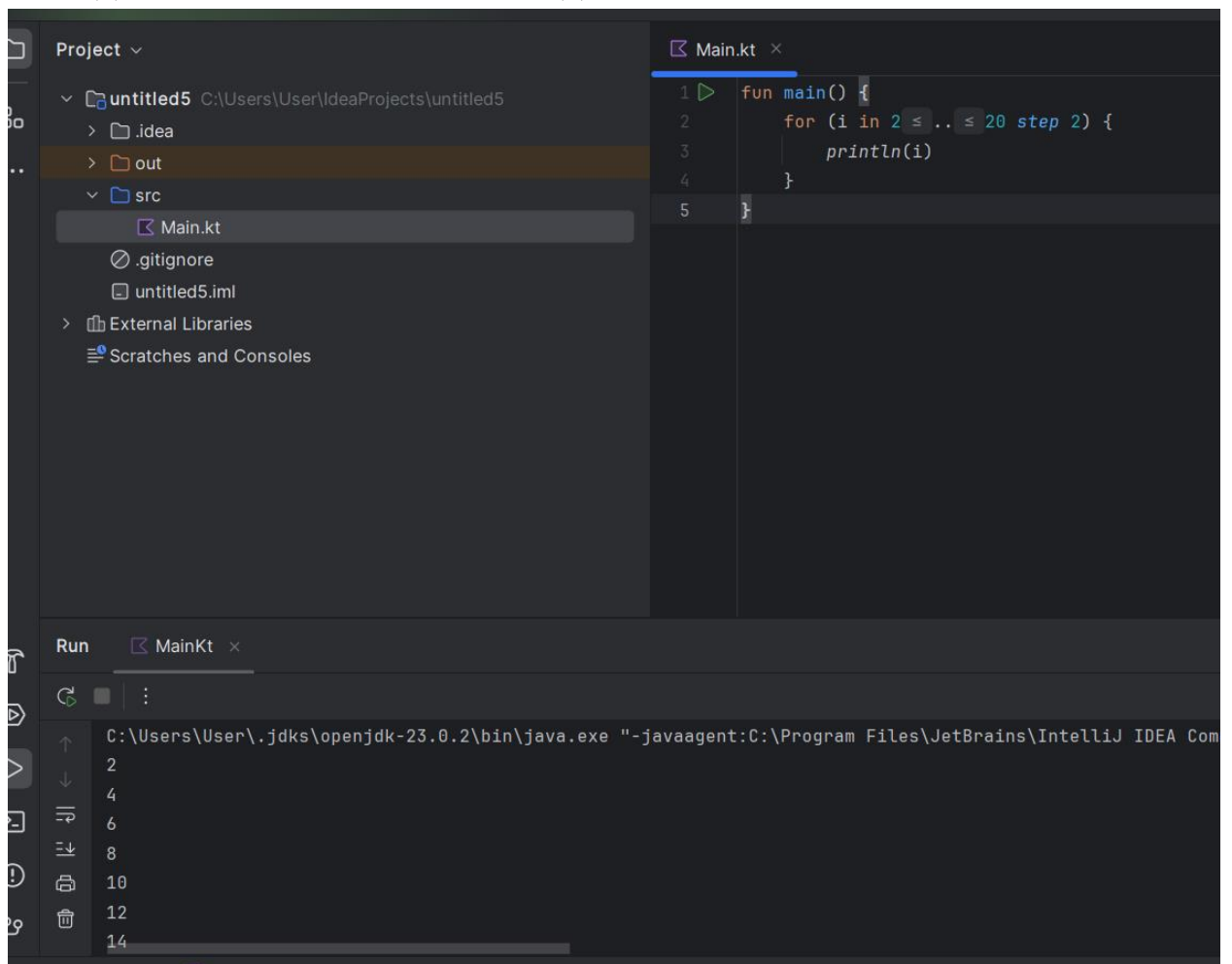


Практическая работа №5

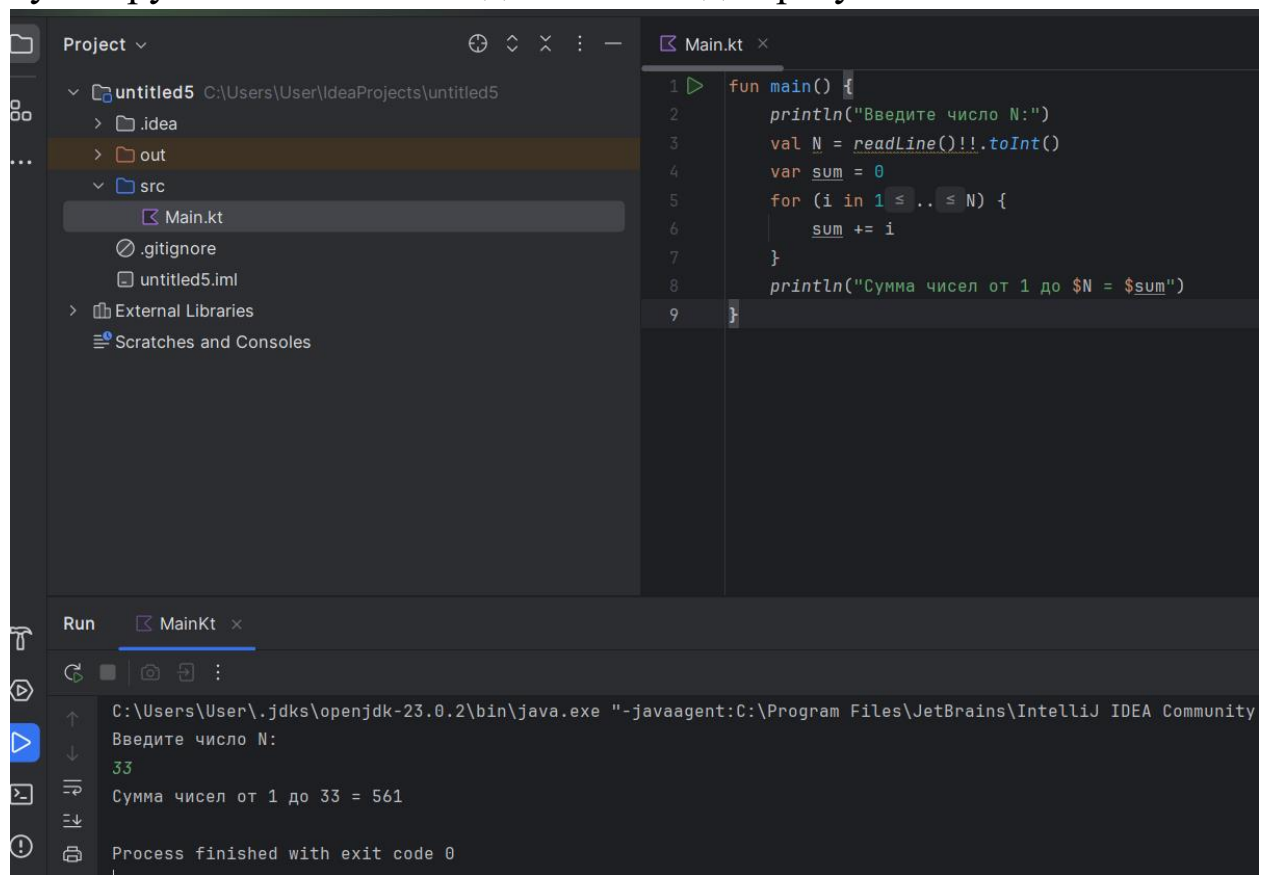
1. Вывод чисел от 1 до 10: Напишите программу, которая выводит числа от 1 до 10.



2. Вывод четных чисел от 1 до 20: Напишите программу, которая выводит все четные числа от 1 до 20.



3. Сумма чисел от 1 до N: Пользователь вводит число N, программа суммирует все числа от 1 до N и выводит результат.



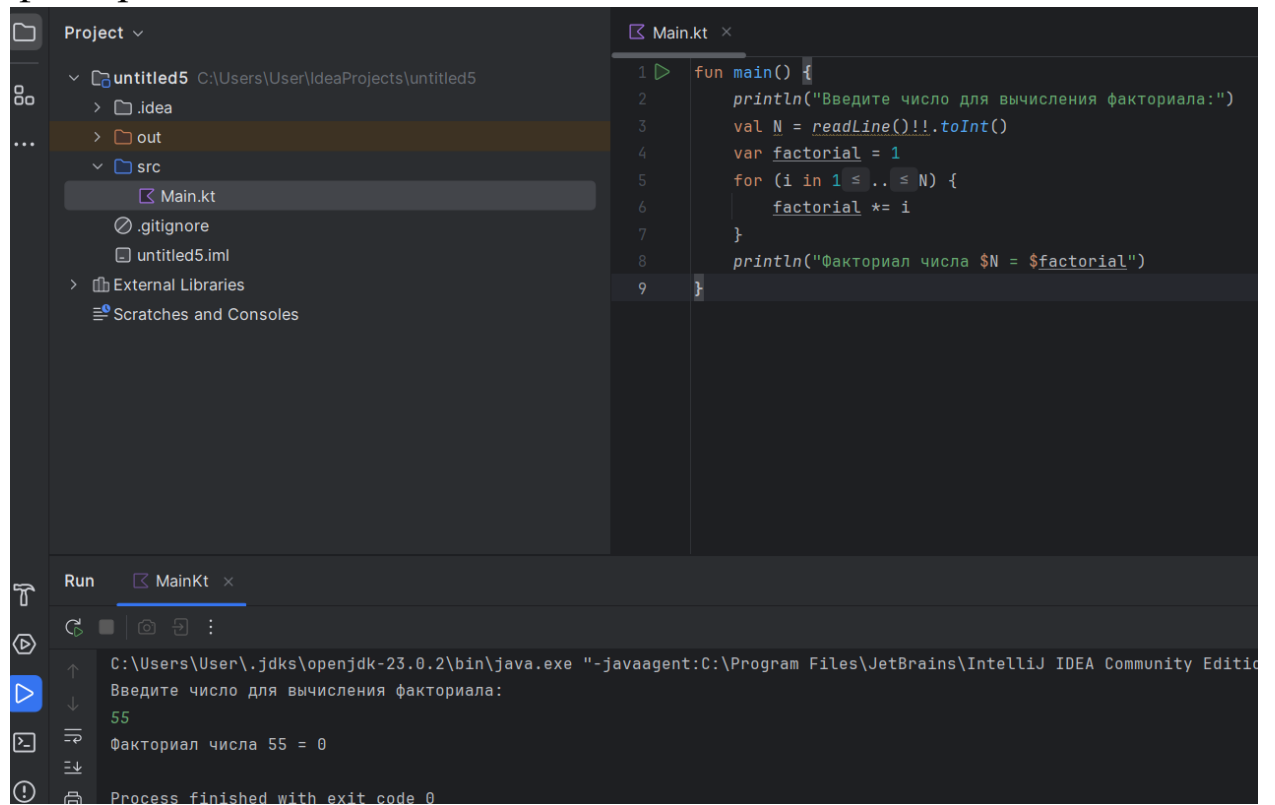
The screenshot displays the IntelliJ IDEA IDE interface. The left sidebar shows the project structure for 'untitled5', with the 'src' directory containing 'Main.kt'. The main editor window shows the following Kotlin code in 'Main.kt':

```
1 fun main() {  
2     println("Введите число N:")  
3     val N = readLine()!!.toInt()  
4     var sum = 0  
5     for (i in 1..N) {  
6         sum += i  
7     }  
8     println("Сумма чисел от 1 до $N = $sum")  
9 }
```

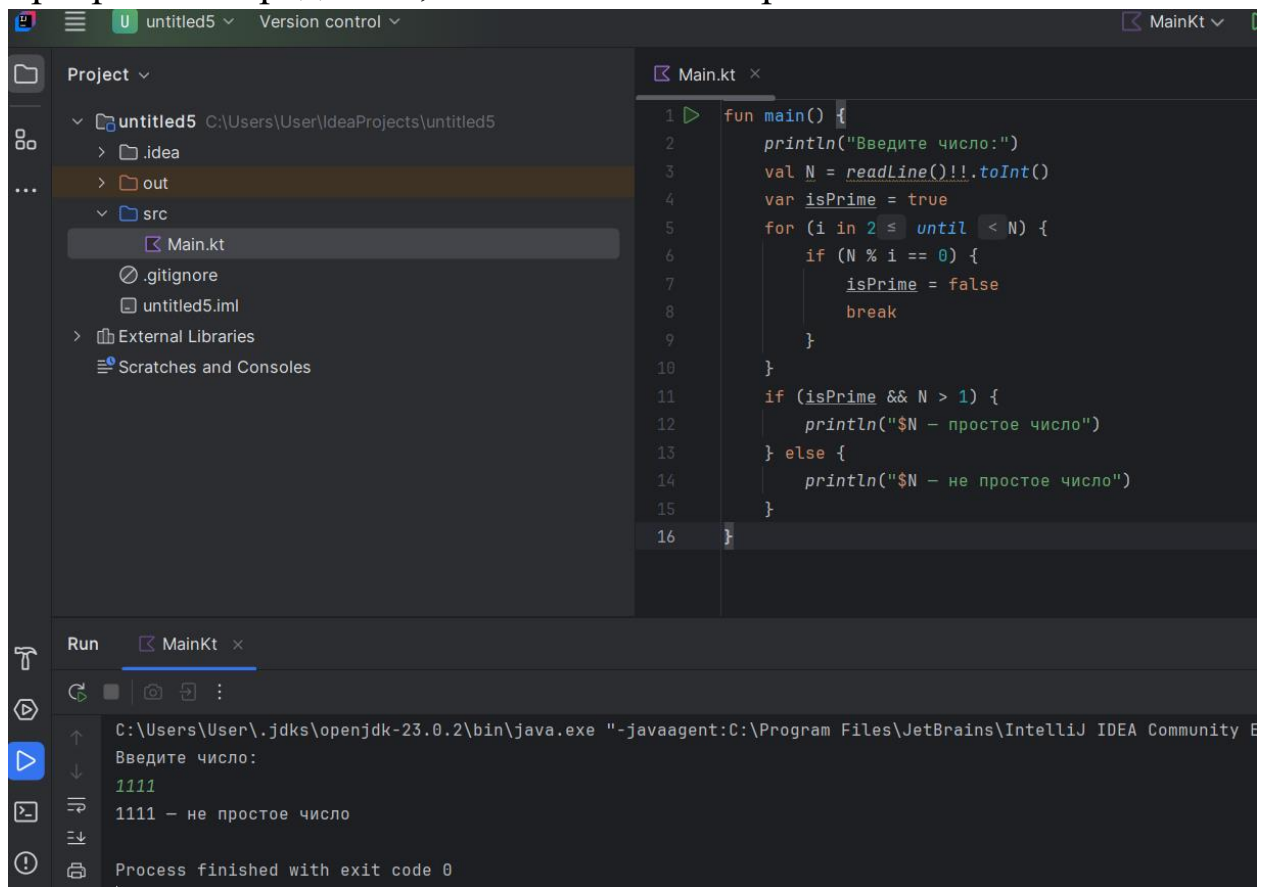
Below the editor, the 'Run' tab is active, showing the execution output for 'MainKt'. The output indicates that the program was executed using the Java runtime, prompted for input, received the value '33', and calculated the sum of numbers from 1 to 33 as 561. The process finished with exit code 0.

```
Run MainKt x  
C:\Users\User\.jds\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community  
Введите число N:  
33  
Сумма чисел от 1 до 33 = 561  
Process finished with exit code 0
```

4. Факториал числа: Напишите программу, которая вычисляет факториал введенного пользователем числа.



5. Проверка числа на простоту: Пользователь вводит число, программа определяет, является ли оно простым.



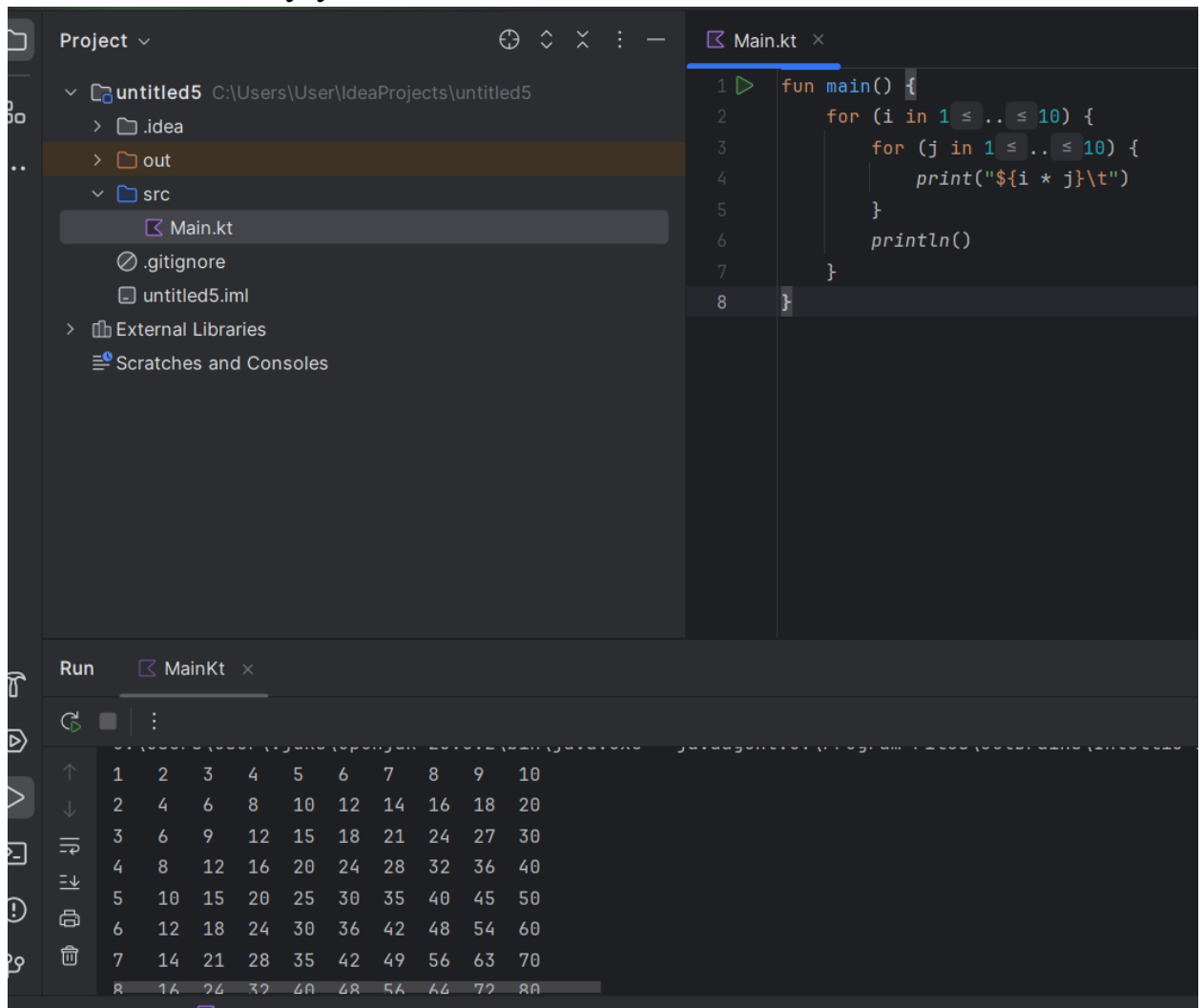
The screenshot displays the IntelliJ IDEA IDE interface. The left sidebar shows the project structure for 'untitled5', with the 'src' directory expanded to show 'Main.kt'. The main editor window displays the following Kotlin code:

```
1 fun main() {  
2     println("Введите число:")  
3     val N = readLine()!!.toInt()  
4     var isPrime = true  
5     for (i in 2..until(N)) {  
6         if (N % i == 0) {  
7             isPrime = false  
8             break  
9         }  
10    }  
11    if (isPrime && N > 1) {  
12        println("$N - простое число")  
13    } else {  
14        println("$N - не простое число")  
15    }  
16 }
```

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

```
C:\Users\User\.jids\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community E  
Введите число:  
1111  
1111 - не простое число  
Process finished with exit code 0
```

6. Вывод таблицы умножения: Напишите программу, которая выводит таблицу умножения от 1 до 10.



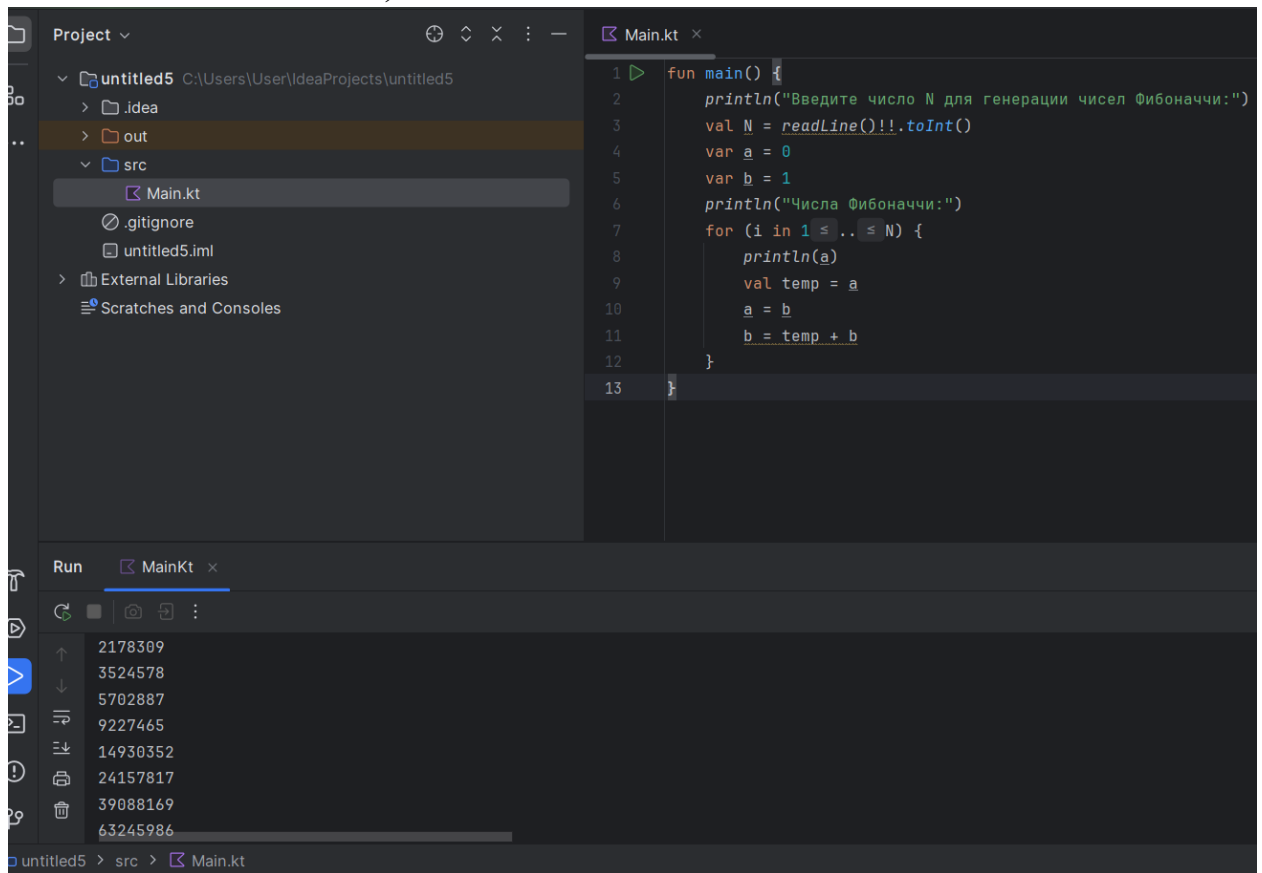
The screenshot shows an IDE with a project named 'untitled5'. The source file 'Main.kt' contains the following Kotlin code:

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

The 'Run' tab at the bottom shows the output of the program, which is a 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. Фибоначчи: Сгенерируйте первые N чисел Фибоначчи (где N вводит пользователь).



The screenshot shows an IDE with a project named 'untitled5'. The 'src' directory contains a file 'Main.kt'. The code in 'Main.kt' is as follows:

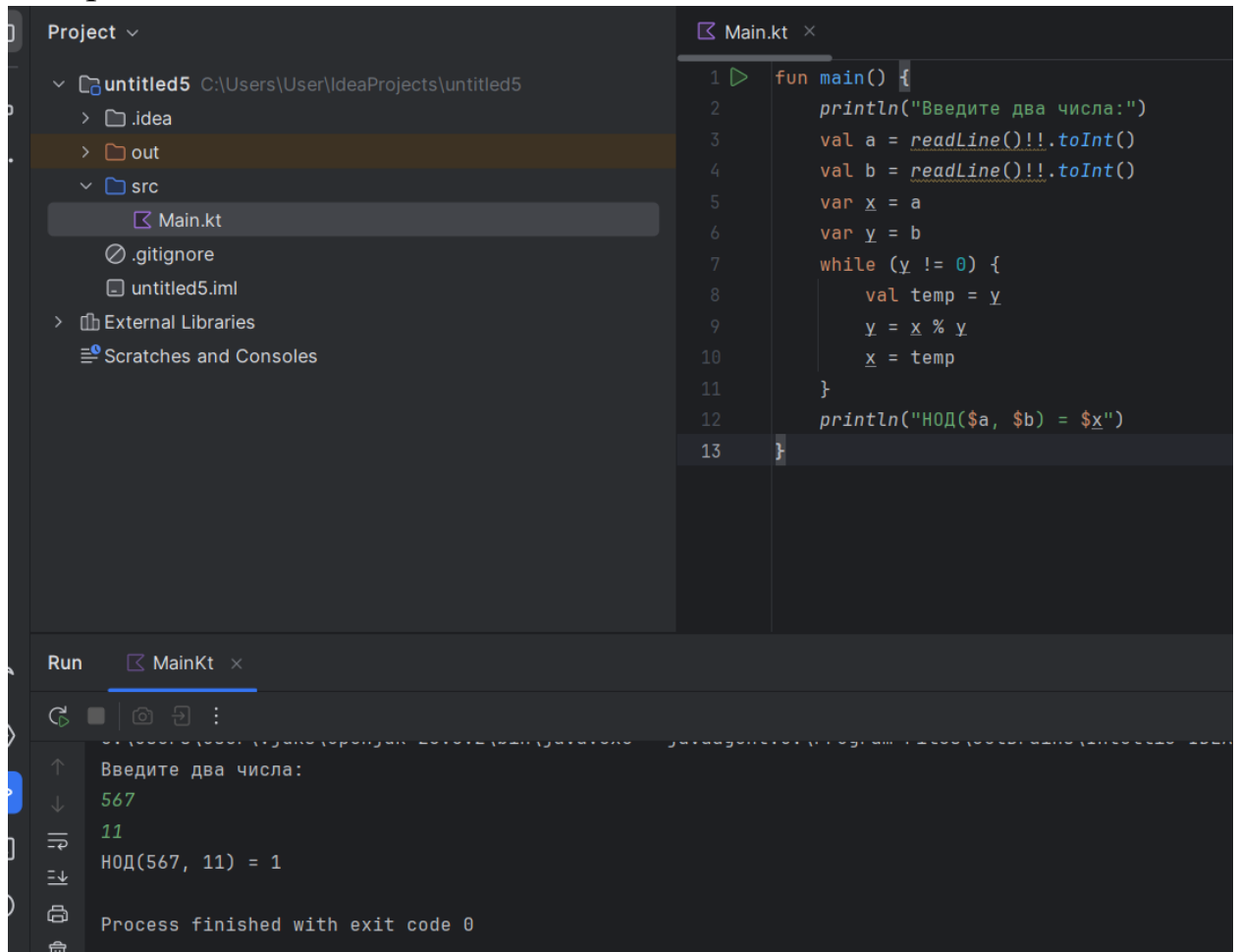
```
1 fun main() {  
2     println("Введите число N для генерации чисел Фибоначчи:")  
3     val N = readLine()!!.toInt()  
4     var a = 0  
5     var b = 1  
6     println("Числа Фибоначчи:")  
7     for (i in 1..N) {  
8         println(a)  
9         val temp = a  
10        a = b  
11        b = temp + b  
12    }  
13 }
```

The 'Run' tab shows the output of the program:

```
2178309  
3524578  
5702887  
9227465  
14930352  
24157817  
39088169  
63245986
```

The status bar at the bottom indicates the current file is 'untitled5 > src > Main.kt'.

8. Наибольший общий делитель (НОД): Напишите программу, которая находит НОД двух введенных чисел с использованием алгоритма Евклида.



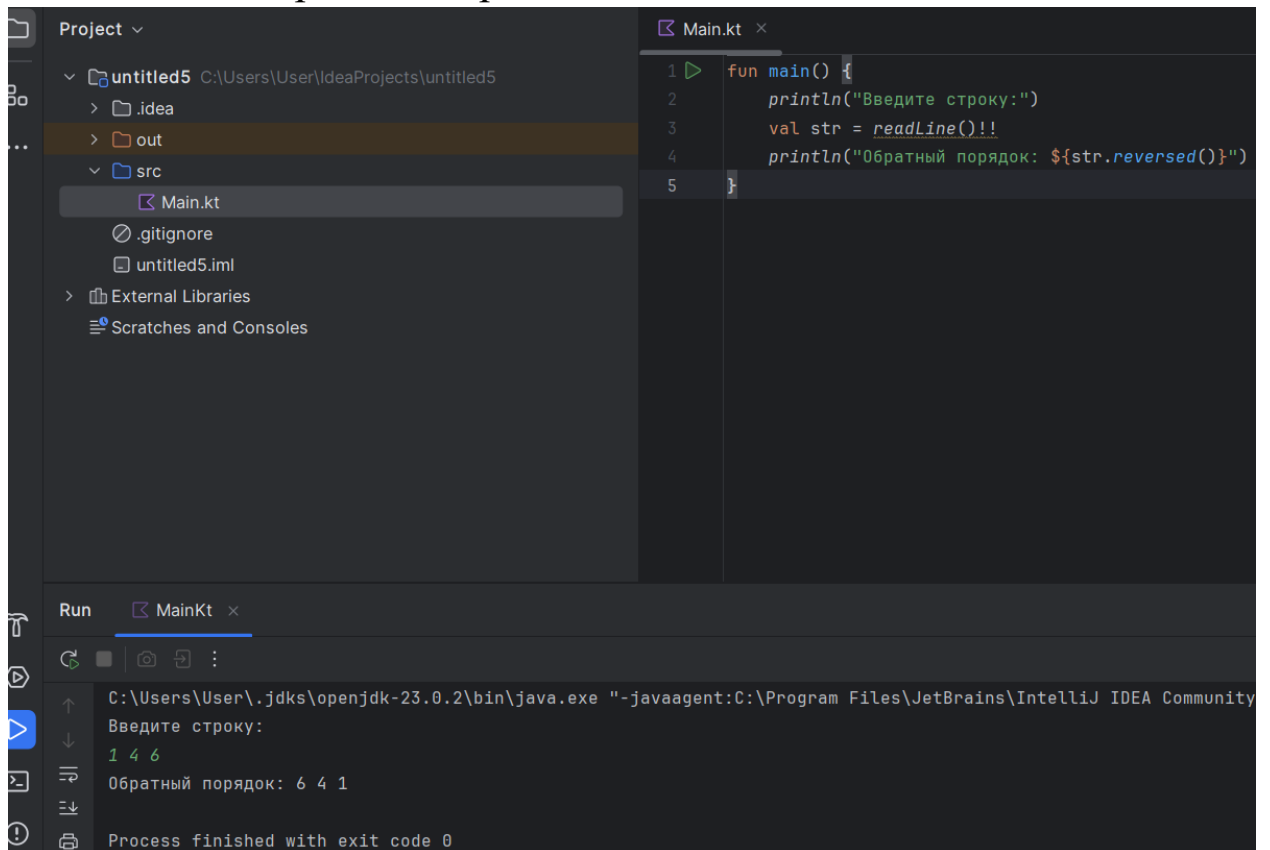
The screenshot displays an IDE interface with a project named 'untitled5'. The 'src' directory contains a file 'Main.kt'. The code in 'Main.kt' implements the Euclidean algorithm to find the GCD of two numbers entered by the user. The code is as follows:

```
1 fun main() {  
2     println("Введите два числа:")  
3     val a = readLine()!!.toInt()  
4     val b = readLine()!!.toInt()  
5     var x = a  
6     var y = b  
7     while (y != 0) {  
8         val temp = y  
9         y = x % y  
10        x = temp  
11    }  
12    println("НОД($a, $b) = $x")  
13 }
```

The 'Run' tab at the bottom shows the execution output:

```
Введите два числа:  
567  
11  
НОД(567, 11) = 1  
Process finished with exit code 0
```


9. Обратный порядок: Пользователь вводит строку, и программа выводит ее в обратном порядке.

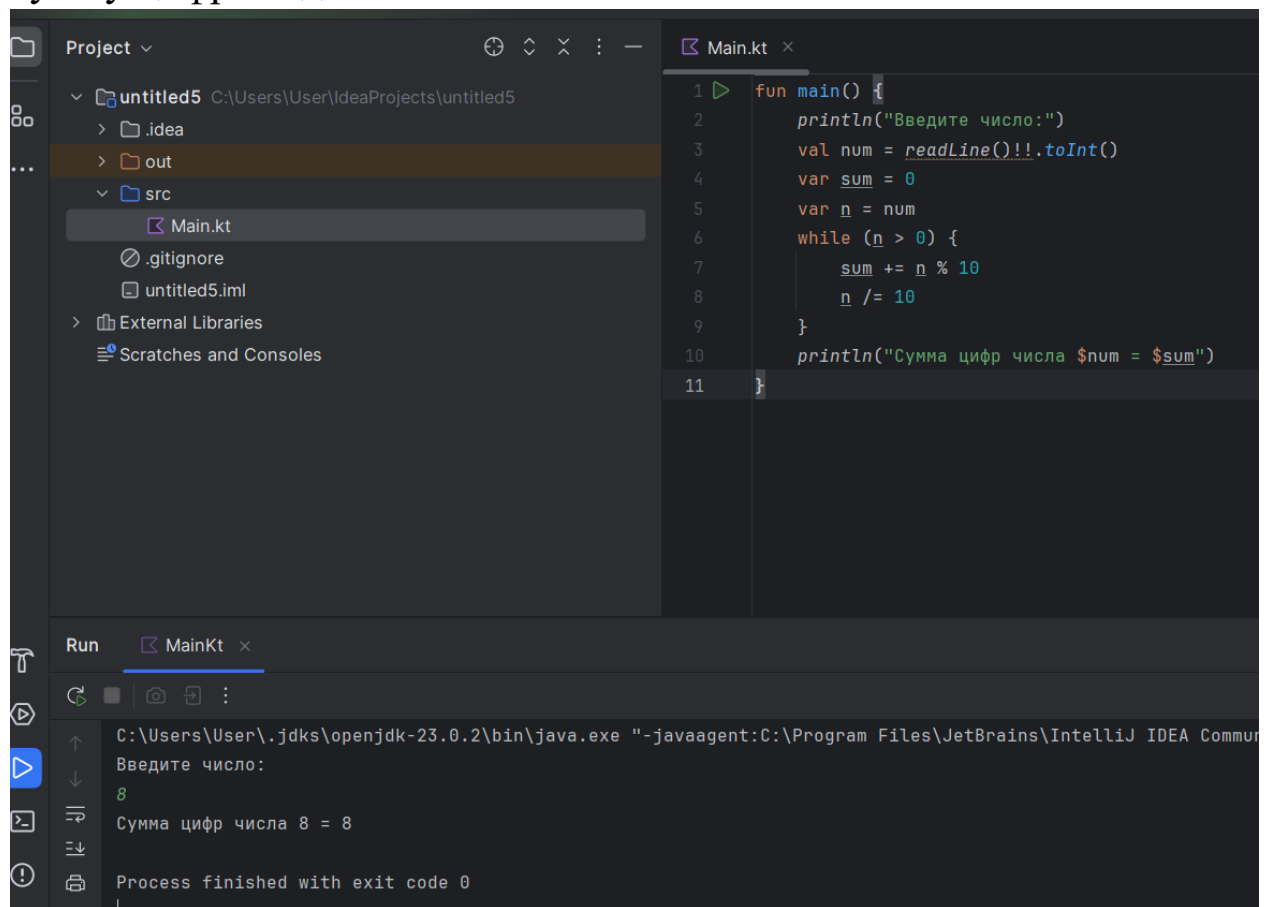


The screenshot displays the IntelliJ IDEA IDE interface. The left sidebar shows the project structure for 'untitled5', with the 'src' directory containing 'Main.kt'. The main editor window shows the following Kotlin code:

```
1 fun main() {  
2     println("Введите строку:")  
3     val str = readLine()!!  
4     println("Обратный порядок: ${str.reversed()}")  
5 }
```

The bottom panel shows the 'Run' output for 'MainKt'. The command executed is: `C:\Users\User\.jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community`. The output shows the prompt 'Введите строку:' followed by the input '1 4 6' and the resulting reversed string 'Обратный порядок: 6 4 1'. The process finished with exit code 0.

10. Сумма цифр числа: Напишите программу, которая находит сумму цифр введенного числа.



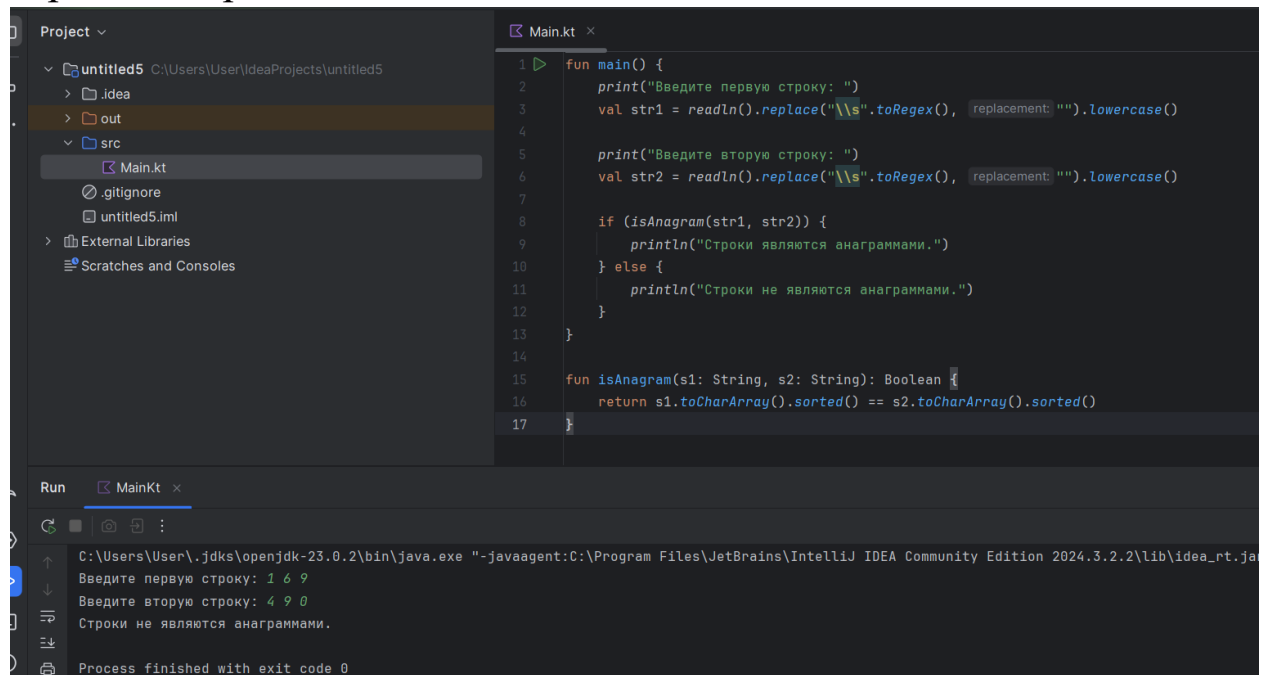
The screenshot displays the IntelliJ IDEA IDE interface. The top-left pane shows the project structure for 'untitled5', with the 'src' directory expanded to show 'Main.kt'. The top-right pane shows the code for 'Main.kt', which is a Kotlin program to calculate the sum of digits of a number. The bottom pane shows the output of the program, indicating that the user entered the number 8 and the sum of its digits is 8.

```
1 fun main() {  
2     println("Введите число:")  
3     val num = readline()!!.toInt()  
4     var sum = 0  
5     var n = num  
6     while (n > 0) {  
7         sum += n % 10  
8         n /= 10  
9     }  
10    println("Сумма цифр числа $num = $sum")  
11 }
```

Run MainKt

C:\Users\User\.jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Commu
Введите число:
8
Сумма цифр числа 8 = 8
Process finished with exit code 0

11. Анаграммы: Программа проверяет, являются ли две введенные строки анаграммами.



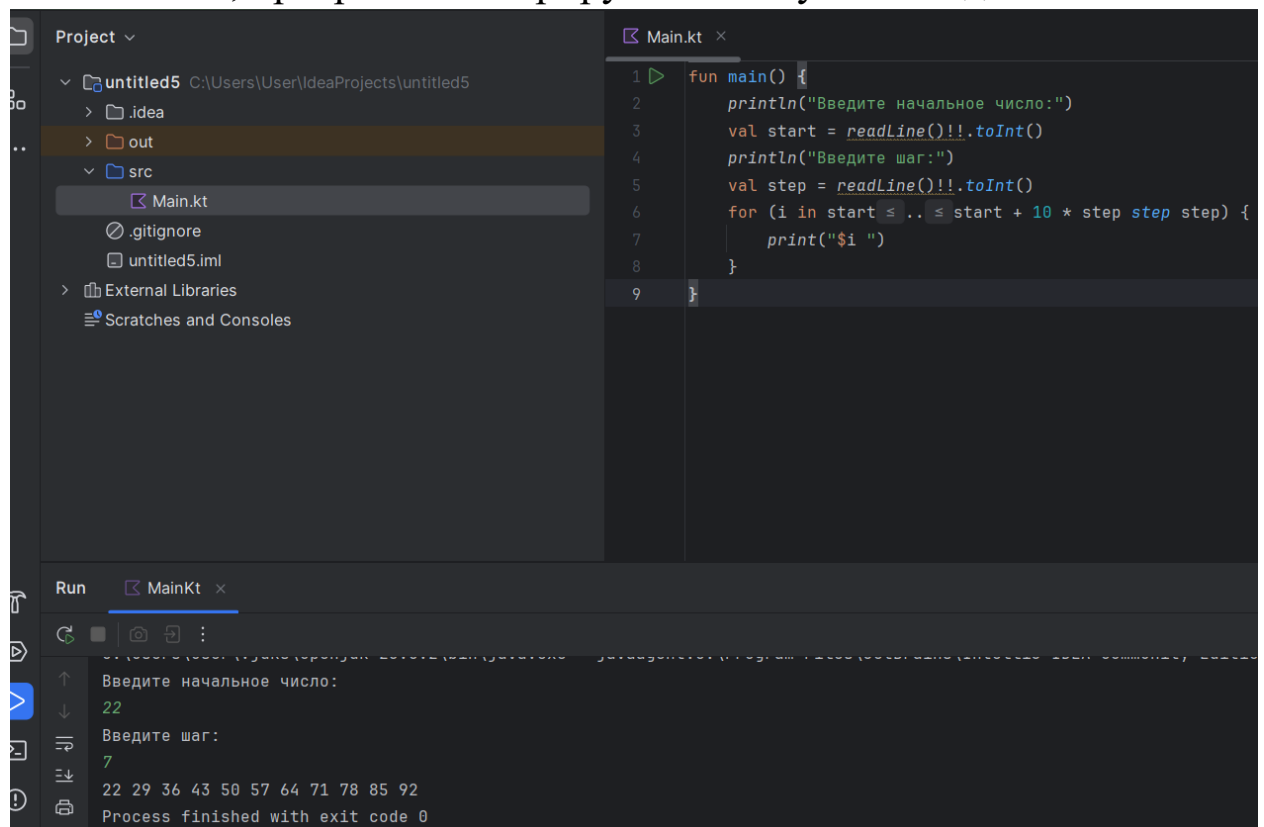
The screenshot displays the IntelliJ IDEA IDE with a Kotlin project named 'untitled5'. The 'Project' sidebar on the left shows the file structure: 'src' contains 'Main.kt', and 'out' is the output directory. The 'Main.kt' file is open in the editor, showing the following code:

```
1 fun main() {  
2     print("Введите первую строку: ")  
3     val str1 = readln().replace("\\s".toRegex(), replacement: "").lowercase()  
4  
5     print("Введите вторую строку: ")  
6     val str2 = readln().replace("\\s".toRegex(), replacement: "").lowercase()  
7  
8     if (isAnagram(str1, str2)) {  
9         println("Строки являются анаграммами.")  
10    } else {  
11        println("Строки не являются анаграммами.")  
12    }  
13 }  
14  
15 fun isAnagram(s1: String, s2: String): Boolean {  
16     return s1.toCharArray().sorted() == s2.toCharArray().sorted()  
17 }
```

The 'Run' tab at the bottom shows the execution output:

```
C:\Users\User\.jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.2.2\lib\idea_rt.jar"  
Введите первую строку: 1 6 9  
Введите вторую строку: 4 9 0  
Строки не являются анаграммами.  
Process finished with exit code 0
```

12. Числовая последовательность: Пользователь вводит начальное число и шаг, программа генерирует числовую последовательность.



The screenshot displays an IDE interface with a project named 'untitled5'. The 'src' directory contains a file 'Main.kt'. The code in 'Main.kt' is as follows:

```
1 fun main() {  
2     println("Введите начальное число:")  
3     val start = readLine()!!.toInt()  
4     println("Введите шаг:")  
5     val step = readLine()!!.toInt()  
6     for (i in start .. start + 10 * step step step) {  
7         print("$i ")  
8     }  
9 }
```

The 'Run' tab at the bottom shows the execution output:

```
Введите начальное число:  
22  
Введите шаг:  
7  
22 29 36 43 50 57 64 71 78 85 92  
Process finished with exit code 0
```

13. Таблица квадратов: Выведите таблицу квадратов чисел от 1 до 20.

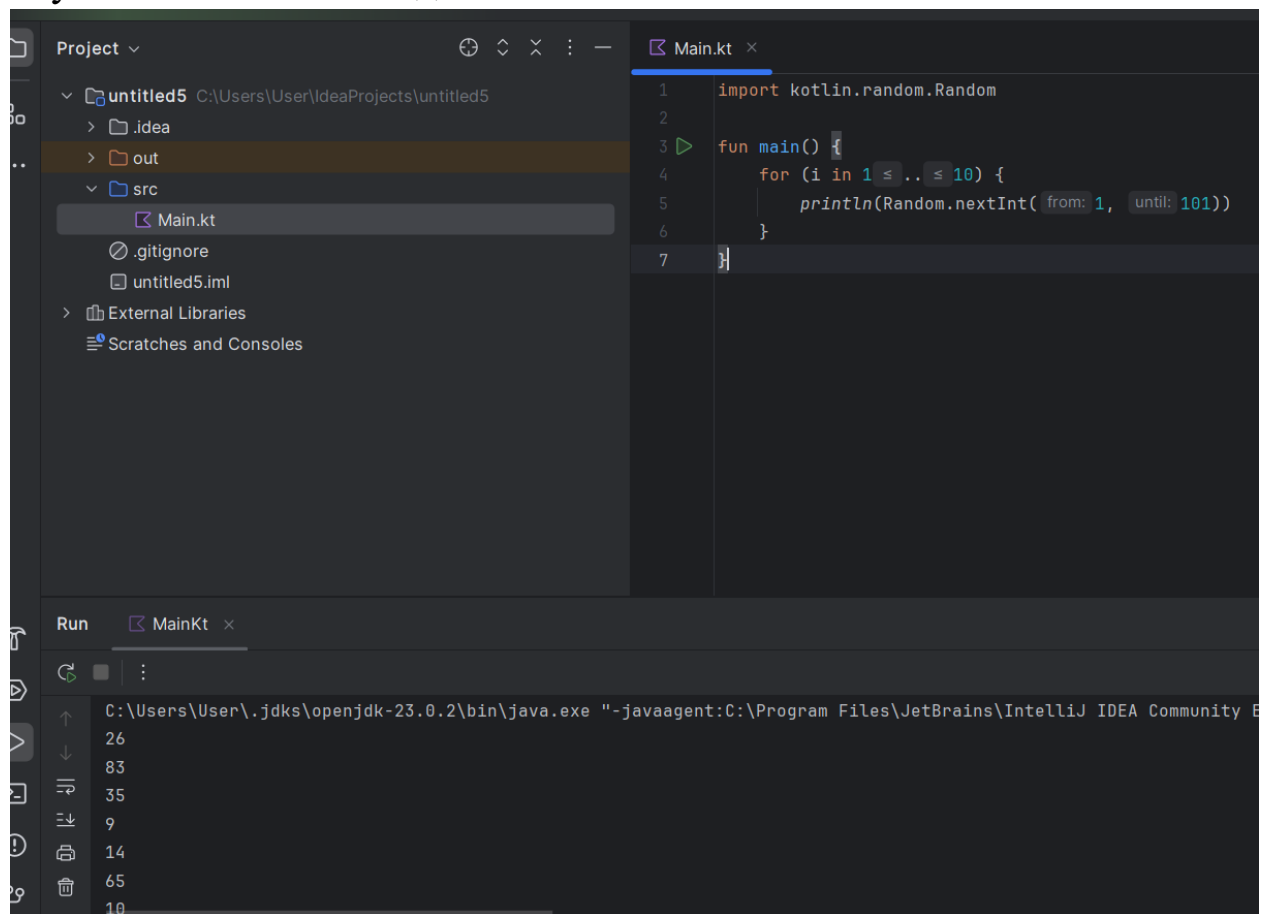
The screenshot shows an IDE with a project named 'untitled5'. The 'src' directory contains a file 'Main.kt'. The code in 'Main.kt' is as follows:

```
1 fun main() {  
2     for (i in 1..20) {  
3         println("Квадрат числа $i = ${i * i}")  
4     }  
5 }
```

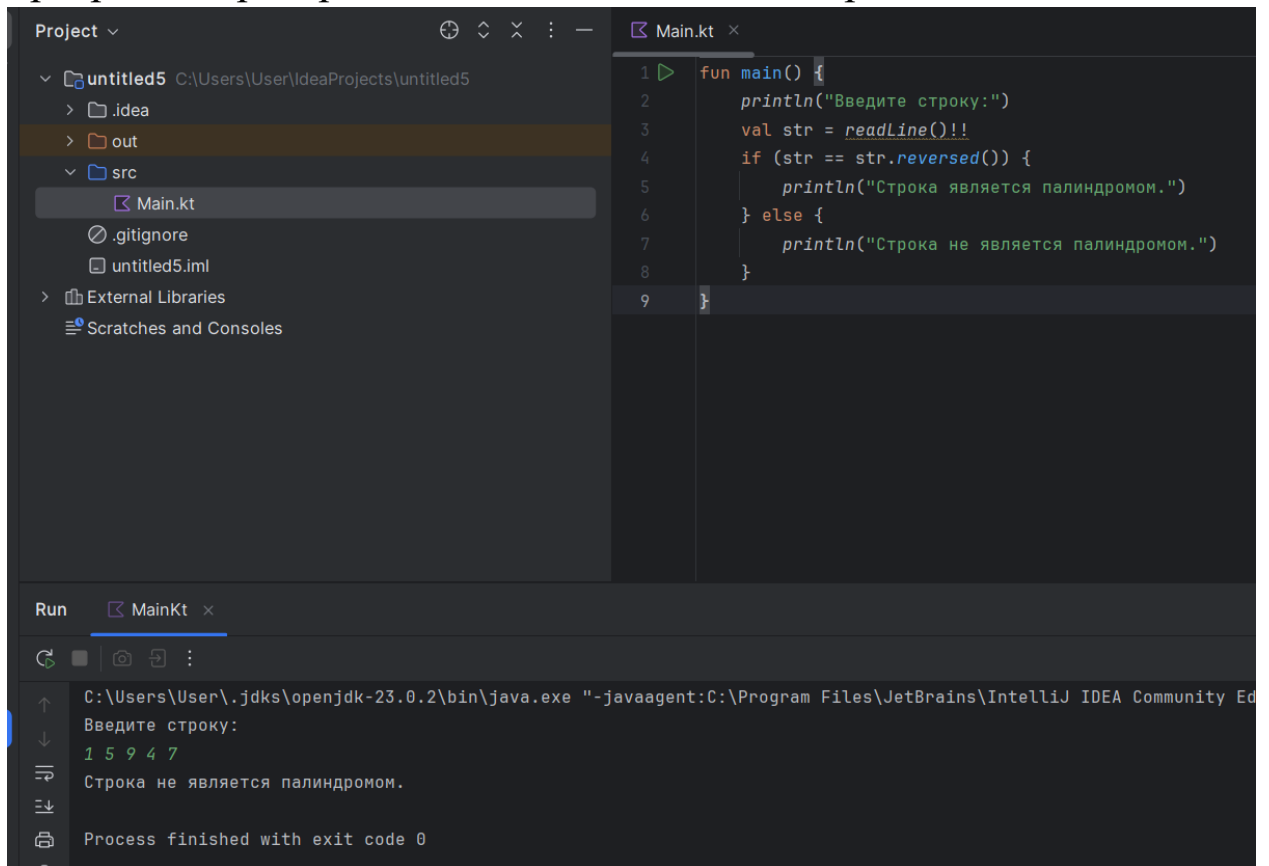
The 'Run' tab is active, showing the output of the program. The output is a list of squares of numbers from 1 to 8, displayed in a scrollable list:

- Квадрат числа 1 = 1
- Квадрат числа 2 = 4
- Квадрат числа 3 = 9
- Квадрат числа 4 = 16
- Квадрат числа 5 = 25
- Квадрат числа 6 = 36
- Квадрат числа 7 = 49
- Квадрат числа 8 = 64

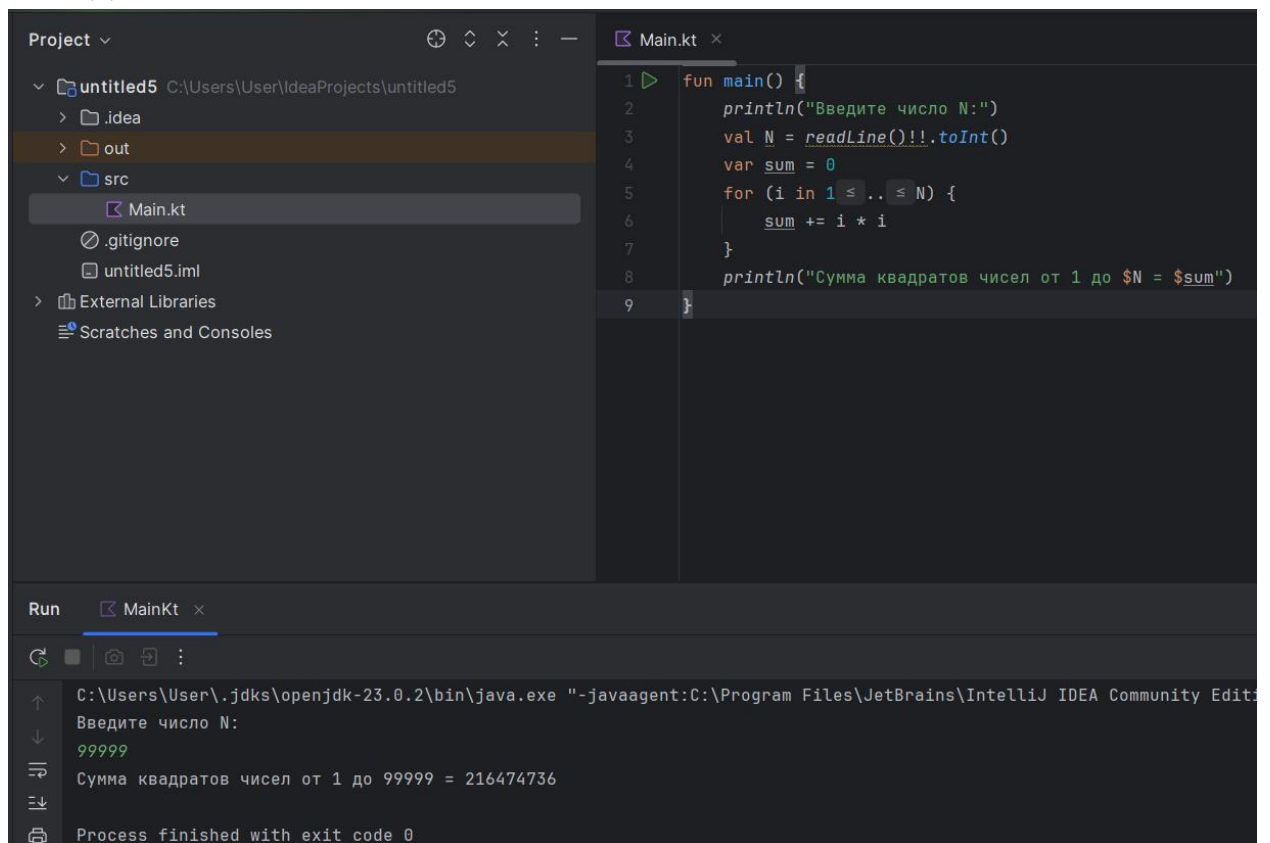
14. Генерация случайных чисел: Сгенерируйте и выведите 10 случайных чисел от 1 до 100.



15. Проверка палиндрома: Пользователь вводит строку, и программа проверяет, является ли она палиндромом.



16. Сигма (сумма квадратов): Найдите сумму квадратов всех чисел от 1 до N.



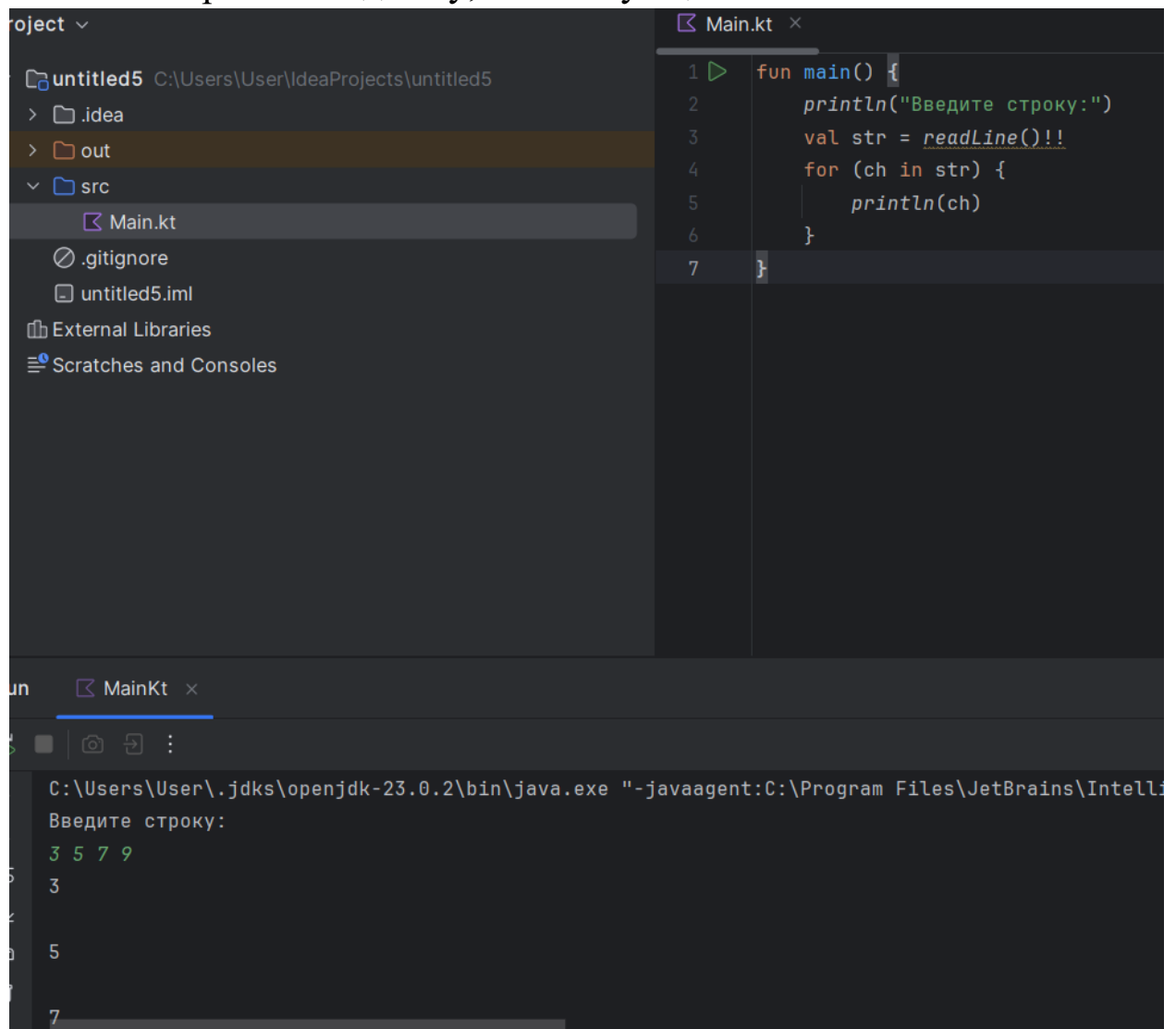
The screenshot displays the IntelliJ IDEA IDE interface. On the left, the Project view shows a project named 'untitled5' with a source folder 'src' containing 'Main.kt'. The main editor window shows the following Kotlin code in 'Main.kt':

```
1 fun main() {  
2     println("Введите число N:")  
3     val N = readLine()!!.toInt()  
4     var sum = 0  
5     for (i in 1..N) {  
6         sum += i * i  
7     }  
8     println("Сумма квадратов чисел от 1 до $N = $sum")  
9 }
```

At the bottom, the Run tool window shows the execution of 'MainKt'. The console output is as follows:

```
C:\Users\User\.jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Editi  
Введите число N:  
99999  
Сумма квадратов чисел от 1 до 99999 = 216474736  
Process finished with exit code 0
```


17. Вывод символов: Напишите программу, которая выводит символы строки по одному, используя циклы



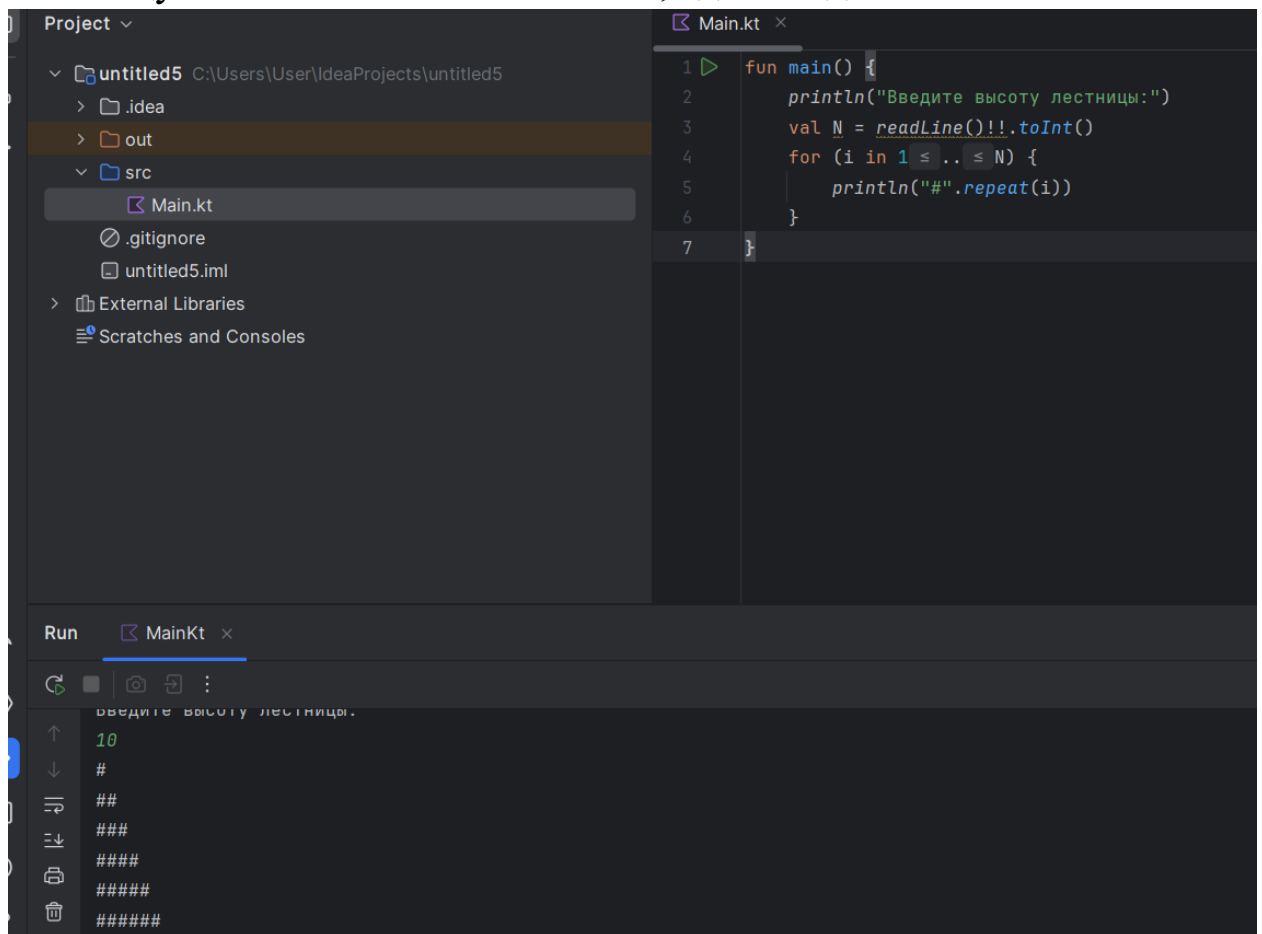
The screenshot shows an IDE with a project named 'untitled5'. The file explorer on the left shows the project structure: 'untitled5' (C:\Users\User\IdeaProjects\untitled5) containing '.idea', 'out', 'src', 'Main.kt', '.gitignore', and 'untitled5.iml'. The 'Main.kt' file is open in the editor, showing the following Kotlin code:

```
1 fun main() {  
2     println("Введите строку:")  
3     val str = readLine()!!  
4     for (ch in str) {  
5         println(ch)  
6     }  
7 }
```

The bottom panel shows the console output for the 'MainKt' application. The command prompt shows the execution of the Java agent and the program. The output is as follows:

```
C:\Users\User\jdk-23.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea-agent.jar" -jar C:\Users\User\IdeaProjects\untitled5\out\production\untitled5  
Введите строку:  
3 5 7 9  
3  
5  
7
```

18. Задача на лестницу: Напишите программу, которая выводит лестницу из символа "#" высотой N, где N задает пользователь.



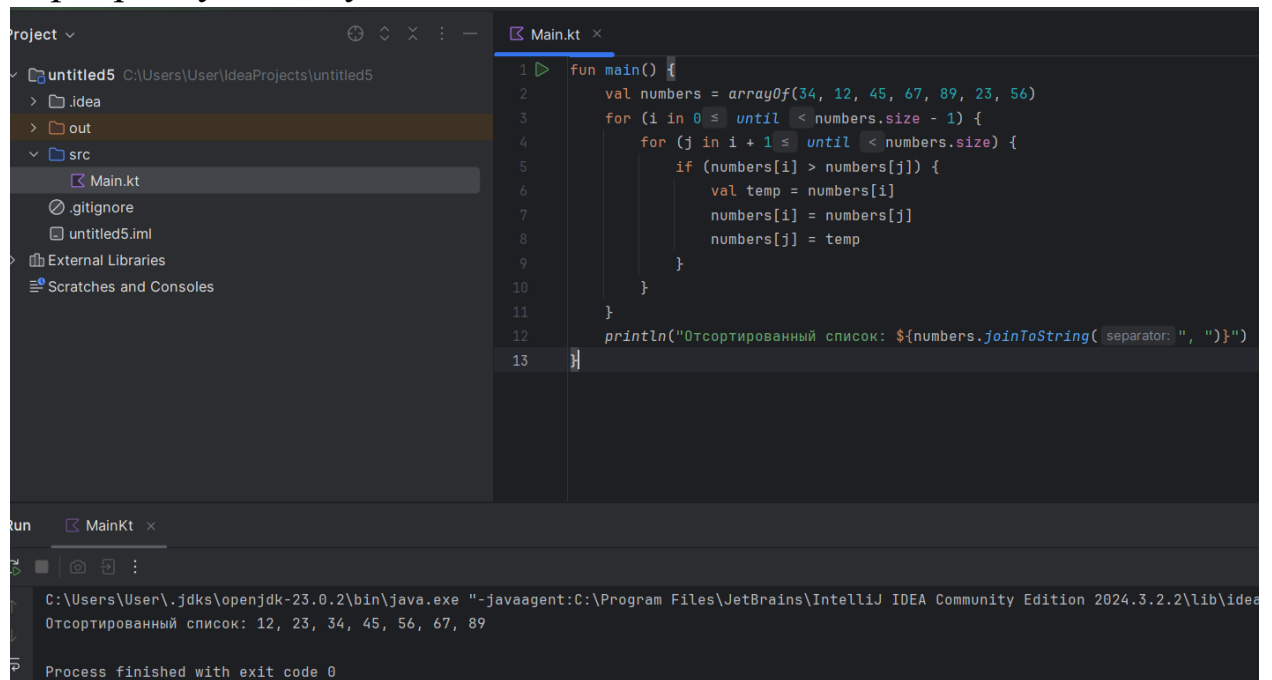
The screenshot shows an IDE with a project named 'untitled5'. The source file 'Main.kt' contains the following Kotlin code:

```
1 fun main() {  
2     println("Введите высоту лестницы:")  
3     val N = readLine()!!.toInt()  
4     for (i in 1..N) {  
5         println("#".repeat(i))  
6     }  
7 }
```

The Run window shows the execution of the program. The prompt 'Введите высоту лестницы.' is displayed, and the user has entered '10'. The output shows a staircase of '#' characters, with each line containing a number of '#' characters equal to the line number (1 to 10).

```
↑ 10  
↓ #  
↵ ##  
↵ ###  
↵ ####  
↵ #####  
↵ #####
```

19. Сортировка списка: Используя цикл, напишите простую сортировку для двухзначных чисел в массиве.20.



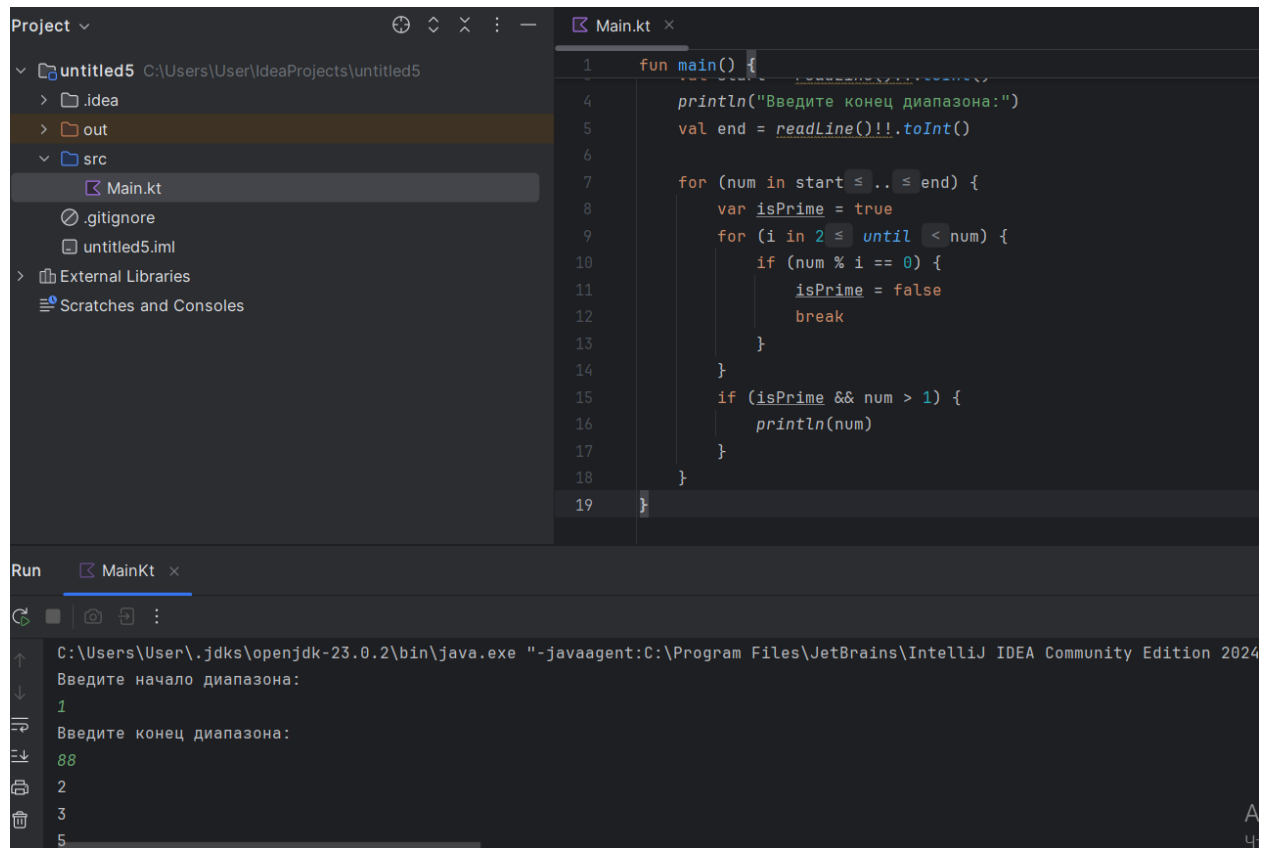
```
1 fun main() {
2     val numbers = arrayOf(34, 12, 45, 67, 89, 23, 56)
3     for (i in 0 until numbers.size - 1) {
4         for (j in i + 1 until numbers.size) {
5             if (numbers[i] > numbers[j]) {
6                 val temp = numbers[i]
7                 numbers[i] = numbers[j]
8                 numbers[j] = temp
9             }
10        }
11    }
12    println("Отсортированный список: ${numbers.joinToString(separator: ", ")}")
13 }
```

Run MainKt

C:\Users\User\.jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.2.2\lib\idea
Отсортированный список: 12, 23, 34, 45, 56, 67, 89

Process finished with exit code 0

20. Простые числа в диапазоне: Выведите все простые числа в заданном пользователем диапазоне.



The screenshot displays the IntelliJ IDEA IDE with a Kotlin project named 'untitled5'. The 'Main.kt' file contains the following code:

```
1 fun main() {  
2     // Введите начало диапазона:  
3     val start = readLine()!!.toInt()  
4     println("Введите конец диапазона:")  
5     val end = readLine()!!.toInt()  
6  
7     for (num in start..end) {  
8         var isPrime = true  
9         for (i in 2..until(num)) {  
10            if (num % i == 0) {  
11                isPrime = false  
12                break  
13            }  
14        }  
15        if (isPrime && num > 1) {  
16            println(num)  
17        }  
18    }  
19 }
```

The 'Run' console at the bottom shows the execution of the program. It prompts the user to enter the start and end of the range. The user enters '1' for the start and '88' for the end. The program then outputs the prime numbers in this range: 2, 3, 5, and 7.

```
C:\Users\User\.jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024\lib\idea_rt.jar=60253:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024\bin" -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024\lib\idea_rt.jar=60253:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024\bin  
Введите начало диапазона:  
1  
Введите конец диапазона:  
88  
2  
3  
5  
7
```

21. Вывод даты: Пользователь вводит год и месяц, программа выводит все даты в этом месяце.

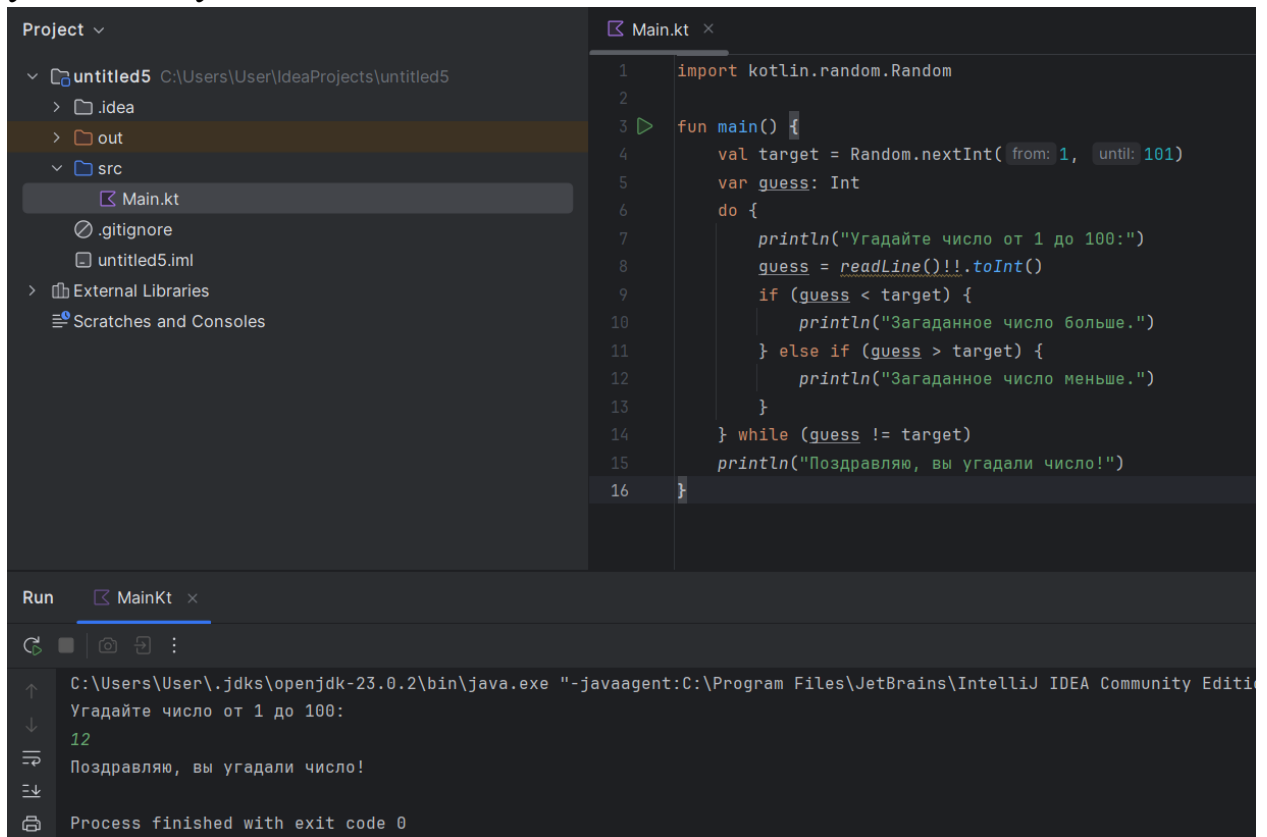
The screenshot displays the IntelliJ IDEA IDE interface. The top-left pane shows the project structure for 'untitled5', with the 'src' directory expanded to show 'Main.kt'. The top-right pane shows the Kotlin code in 'Main.kt'.

```
1 import java.time.LocalDate
2 import java.time.YearMonth
3
4 fun main() {
5     println("Введите год:")
6     val year = readLine()!!.toInt()
7     println("Введите месяц:")
8     val month = readLine()!!.toInt()
9
10    val yearMonth = YearMonth.of(year, month)
11    val daysInMonth = yearMonth.lengthOfMonth()
12
13    for (day in 1..daysInMonth) {
14        val date = LocalDate.of(year, month, day)
15        println(date)
16    }
17 }
```

The bottom pane shows the console output of the program. The command prompt path is visible at the top. The user has entered '2025' for the year and '1' for the month. The program has output the dates '2025-01-01', '2025-01-02', and '2025-01-03'.

```
C:\Users\User\.jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition\lib\idea_rt.jar=1273.0:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition\bin" -Dfile.encoding=UTF-8
Введите год:
2025
Введите месяц:
1
2025-01-01
2025-01-02
2025-01-03
```

22. Угадай число: Напишите игру, в которой пользователь должен угадать случайное число от 1 до 100.



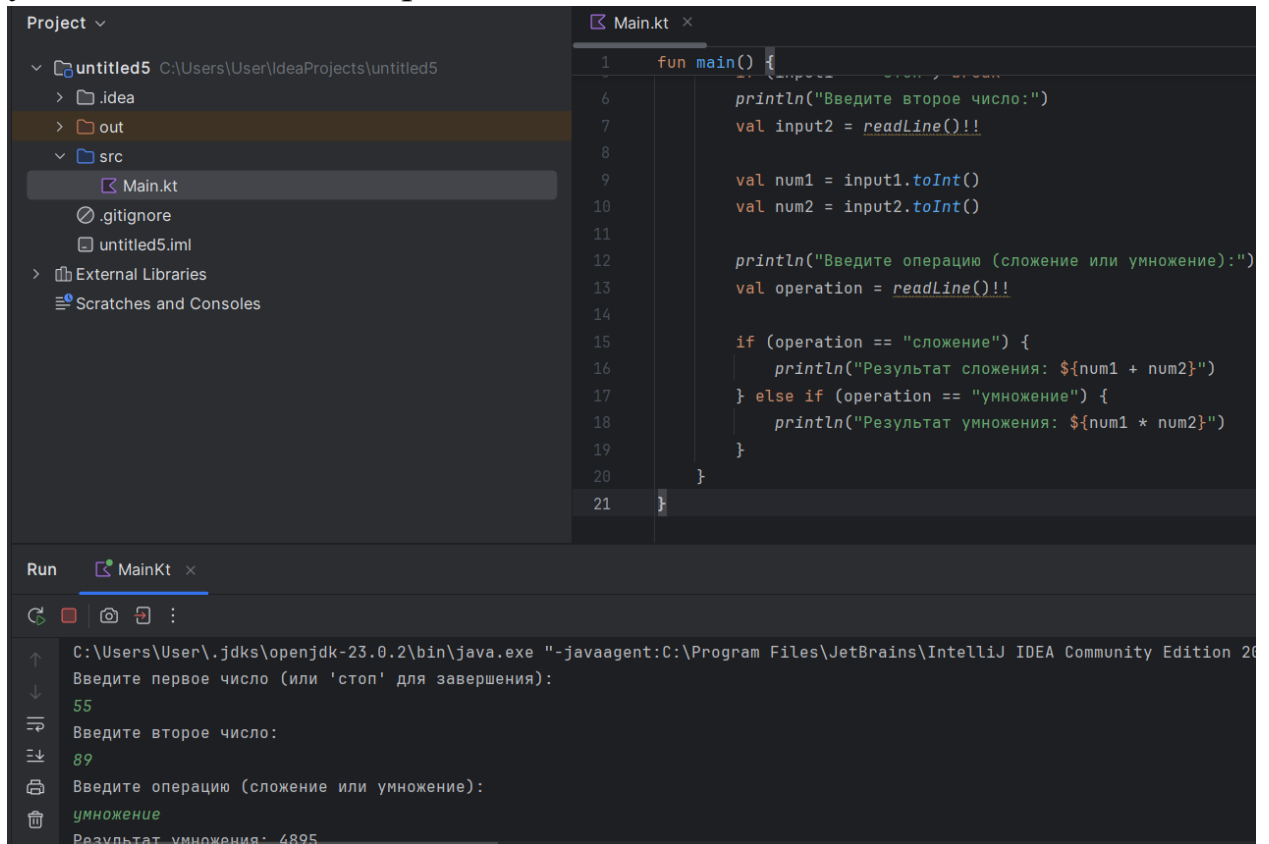
The screenshot displays the IntelliJ IDEA IDE interface. The top-left pane shows the project structure for 'untitled5', with the 'src' directory containing 'Main.kt'. The top-right pane shows the code for 'Main.kt', which imports 'kotlin.random.Random' and defines a 'main' function. The function generates a random target number between 1 and 100, prompts the user to guess, and uses a 'do-while' loop to keep asking until the guess is correct. The bottom pane shows the 'Run' output, indicating the program executed successfully with exit code 0, displaying the prompt, the user's input '12', and the congratulatory message.

```
1 import kotlin.random.Random
2
3 fun main() {
4     val target = Random.nextInt( from: 1, until: 101)
5     var guess: Int
6     do {
7         println("Угадайте число от 1 до 100:")
8         guess = readLine()!!.toInt()
9         if (guess < target) {
10             println("Загаданное число больше.")
11         } else if (guess > target) {
12             println("Загаданное число меньше.")
13         }
14     } while (guess != target)
15     println("Поздравляю, вы угадали число!")
16 }
```

Run MainKt x

C:\Users\User\.jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Editi
Угадайте число от 1 до 100:
12
Поздравляю, вы угадали число!
Process finished with exit code 0

23. Сложение и умножение: Напишите программу, которая запрашивает у пользователя две цифры и повторяет сложение или умножение, до тех пор, пока пользователь не введет "стоп".



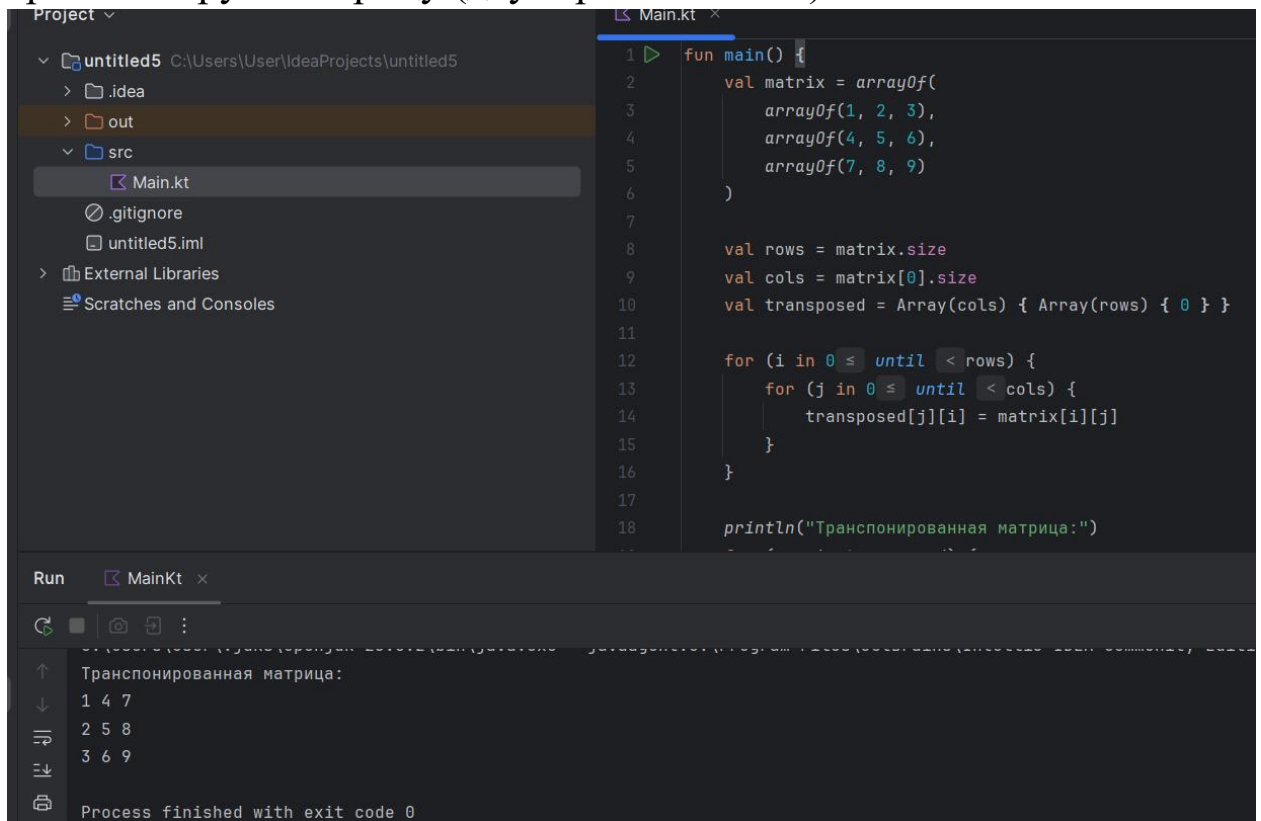
The screenshot displays the IntelliJ IDEA IDE with a Kotlin project named 'untitled5'. The 'src' directory contains a file named 'Main.kt'. The code in 'Main.kt' is as follows:

```
1 fun main() {
2     println("Введите первое число:")
3     val input1 = readLine()!!
4
5     println("Введите второе число:")
6     val input2 = readLine()!!
7
8     val num1 = input1.toInt()
9     val num2 = input2.toInt()
10
11     println("Введите операцию (сложение или умножение):")
12     val operation = readLine()!!
13
14     if (operation == "сложение") {
15         println("Результат сложения: ${num1 + num2}")
16     } else if (operation == "умножение") {
17         println("Результат умножения: ${num1 * num2}")
18     }
19 }
20
21 }
```

The 'Run' tab at the bottom shows the execution of the program. The output is as follows:

```
C:\Users\User\.jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.3\lib\idea_rt.jar=60253:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.3\bin" -Didea.config.path=C:\Users\User\AppData\Local\JetBrains\IntelliJ IDEA\config\ -Didea.system.path=C:\Users\User\AppData\Local\JetBrains\IntelliJ IDEA\system\ -Didea.version=2023.3
Введите первое число (или 'стоп' для завершения):
55
Введите второе число:
89
Введите операцию (сложение или умножение):
умножение
Результат умножения: 4895
```

24. Транспонирование матрицы: Напишите программу, которая транспонирует матрицу (двумерный массив).



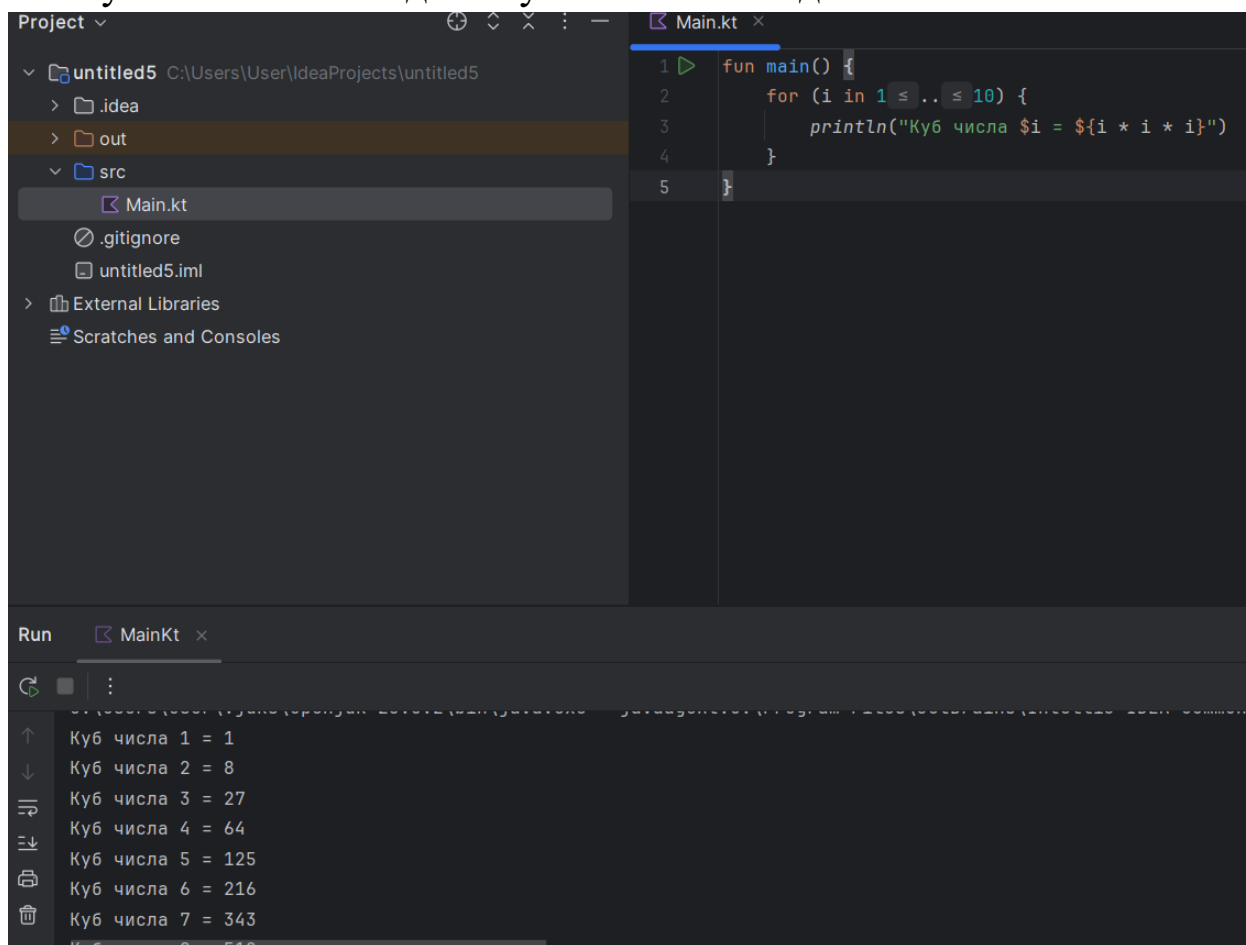
The screenshot shows an IDE with a project named 'untitled5'. The source file 'Main.kt' contains the following Kotlin code:

```
1 fun main() {  
2     val matrix = arrayOf(  
3         arrayOf(1, 2, 3),  
4         arrayOf(4, 5, 6),  
5         arrayOf(7, 8, 9)  
6     )  
7  
8     val rows = matrix.size  
9     val cols = matrix[0].size  
10    val transposed = Array(cols) { Array(rows) { 0 } }  
11  
12    for (i in 0 until rows) {  
13        for (j in 0 until cols) {  
14            transposed[j][i] = matrix[i][j]  
15        }  
16    }  
17  
18    println("Транспонированная матрица:")  
19 }
```

The Run window shows the output of the program:

```
↑ Транспонированная матрица:  
↓ 1 4 7  
  2 5 8  
  3 6 9  
  Process finished with exit code 0
```


25. Кубы чисел: Выведите кубы чисел от 1 до 10.



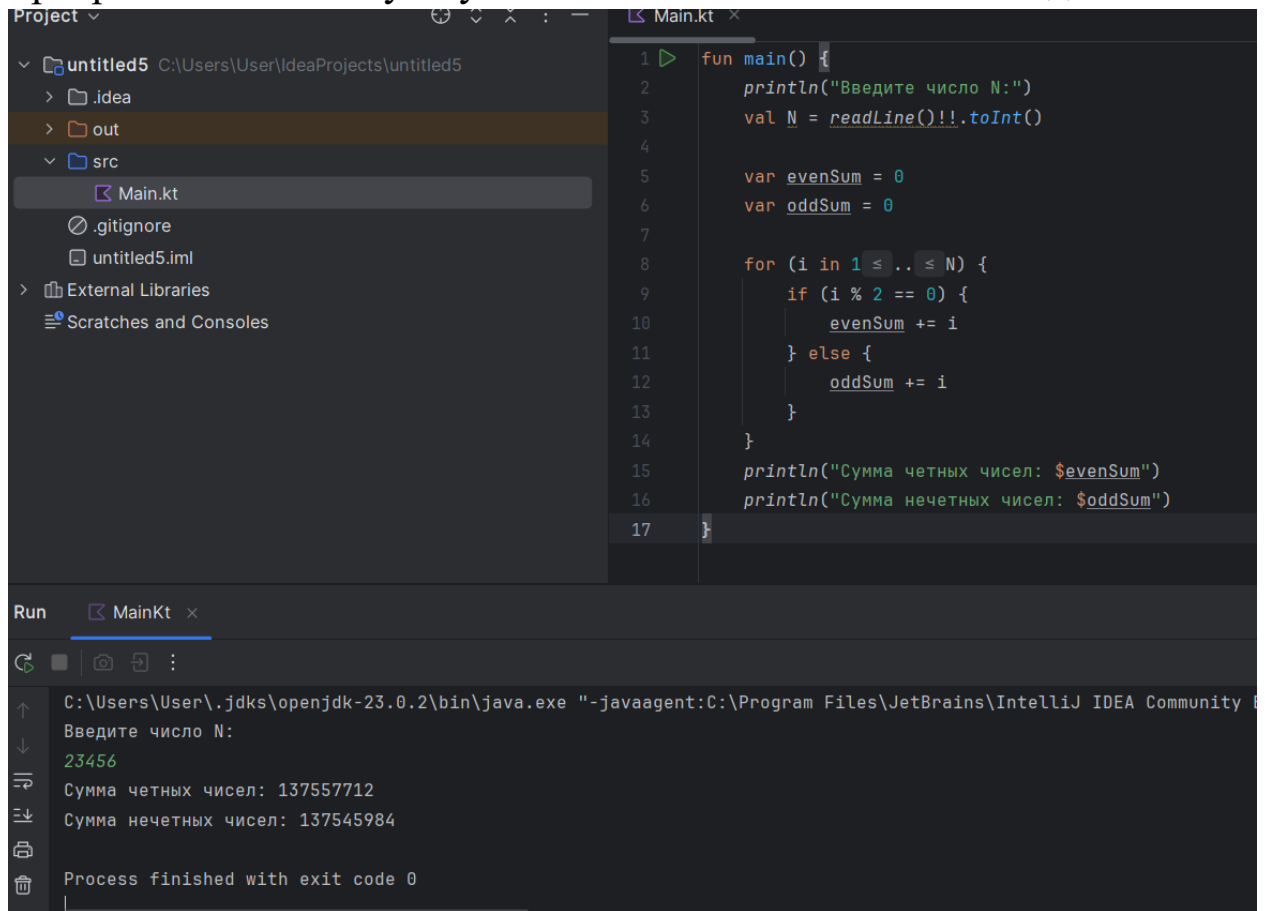
The screenshot shows an IDE with a project named 'untitled5'. The file explorer on the left shows the project structure: 'untitled5' (C:\Users\User\IdeaProjects\untitled5) containing '.idea', 'out', 'src', and 'Main.kt'. The 'Main.kt' file is open in the editor, showing the following Kotlin code:

```
1 fun main() {  
2     for (i in 1..10) {  
3         println("Куб числа $i = ${i * i * i}")  
4     }  
5 }
```

The 'Run' tab at the bottom shows the output of the program:

```
Куб числа 1 = 1  
Куб числа 2 = 8  
Куб числа 3 = 27  
Куб числа 4 = 64  
Куб числа 5 = 125  
Куб числа 6 = 216  
Куб числа 7 = 343  
Куб числа 8 = 512
```

26. Сумма четных и нечетных чисел: Пользователь вводит N, программа считает сумму четных и нечетных чисел от 1 до N.

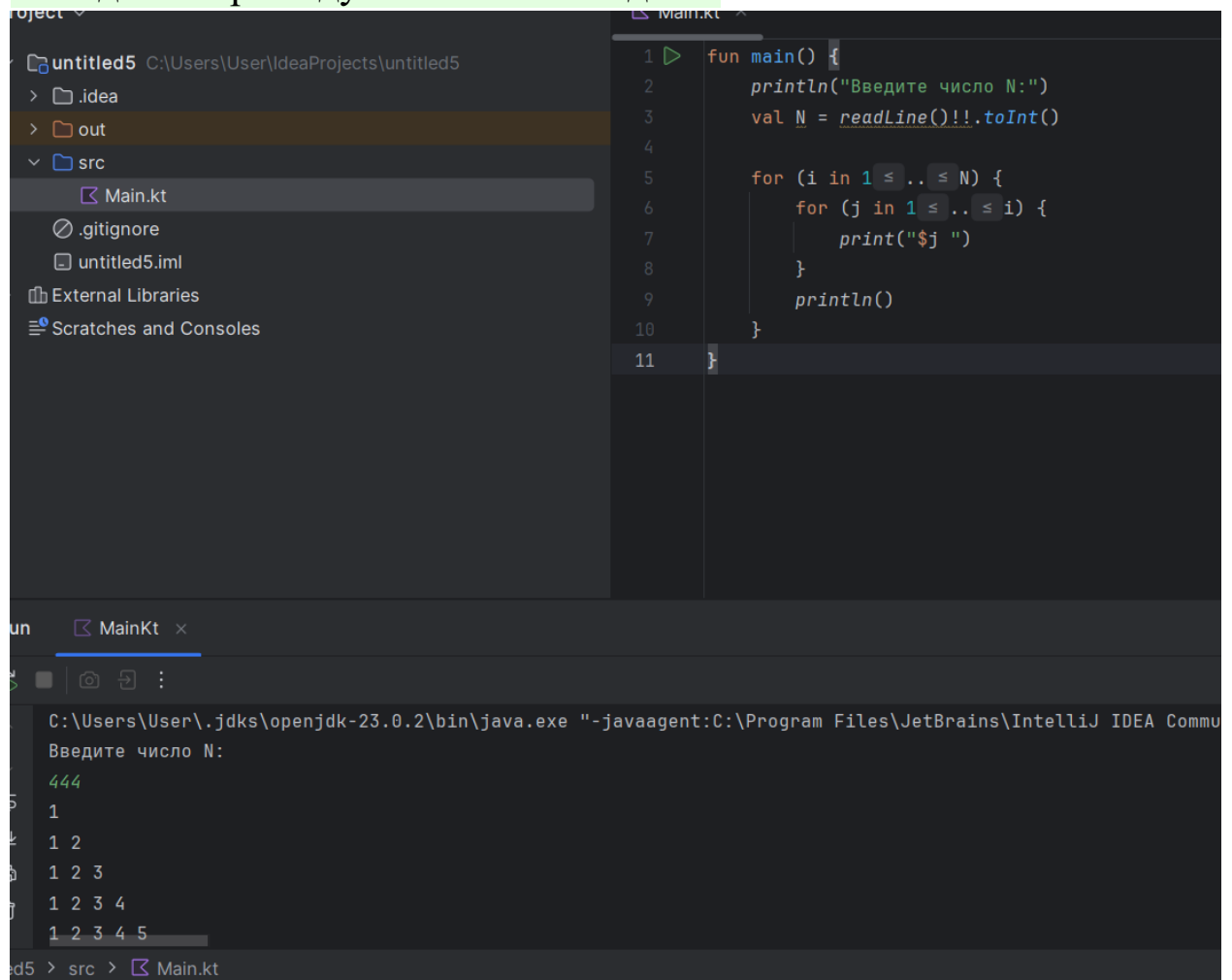


```
Project v
v untitled5 C:\Users\User\IdeaProjects\untitled5
  > .idea
  > out
  v src
    Main.kt
    .gitignore
    untitled5.iml
> External Libraries
Scratches and Consoles

1 fun main() {
2     println("Введите число N:")
3     val N = readLine()!!.toInt()
4
5     var evenSum = 0
6     var oddSum = 0
7
8     for (i in 1..N) {
9         if (i % 2 == 0) {
10             evenSum += i
11         } else {
12             oddSum += i
13         }
14     }
15     println("Сумма четных чисел: $evenSum")
16     println("Сумма нечетных чисел: $oddSum")
17 }
```

```
Run MainKt x
C:\Users\User\.jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
Введите число N:
23456
Сумма четных чисел: 137557712
Сумма нечетных чисел: 137545984
Process finished with exit code 0
```

27. Печать числа "пирамида": Напишите программу, которая выводит "пирамиду" из чисел от 1 до N.



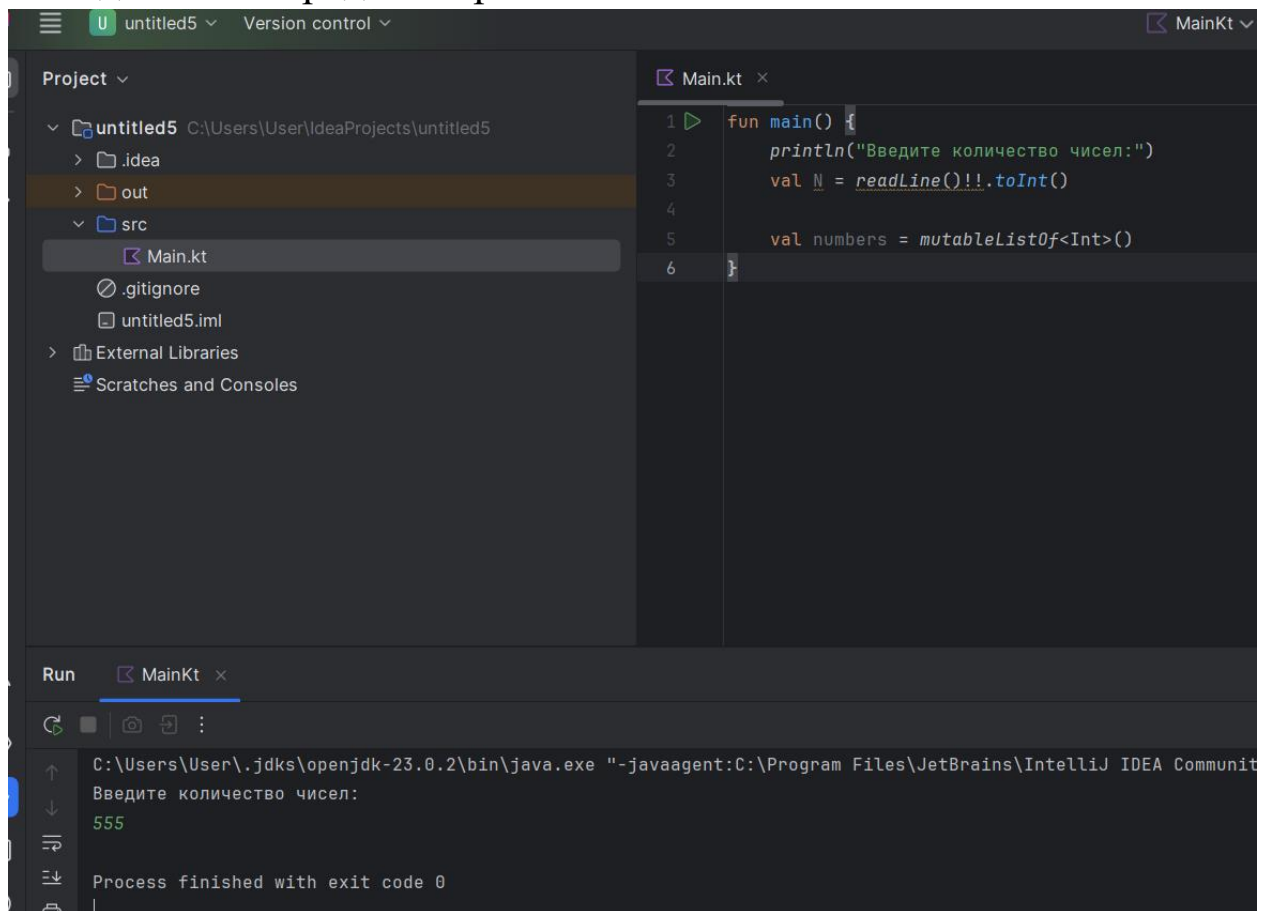
The screenshot displays the IntelliJ IDEA IDE with a Kotlin project named 'untitled5'. The project structure on the left shows the 'src' directory containing 'Main.kt'. The code in 'Main.kt' is as follows:

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

The bottom panel shows the console output. The user has entered '444' in response to the prompt 'Введите число N:'. The program has then printed a pyramid of numbers from 1 to 5, with each row containing one more number than the previous row, and a space after each number. The output is:

```
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5
```

28. Определение порядка: Программа принимает N чисел и выводит их в порядке возрастания.



29. Сумма ряда: Напишите программу, которая находит сумму ряда $1, 1/2, 1/3, \dots, 1/N$.

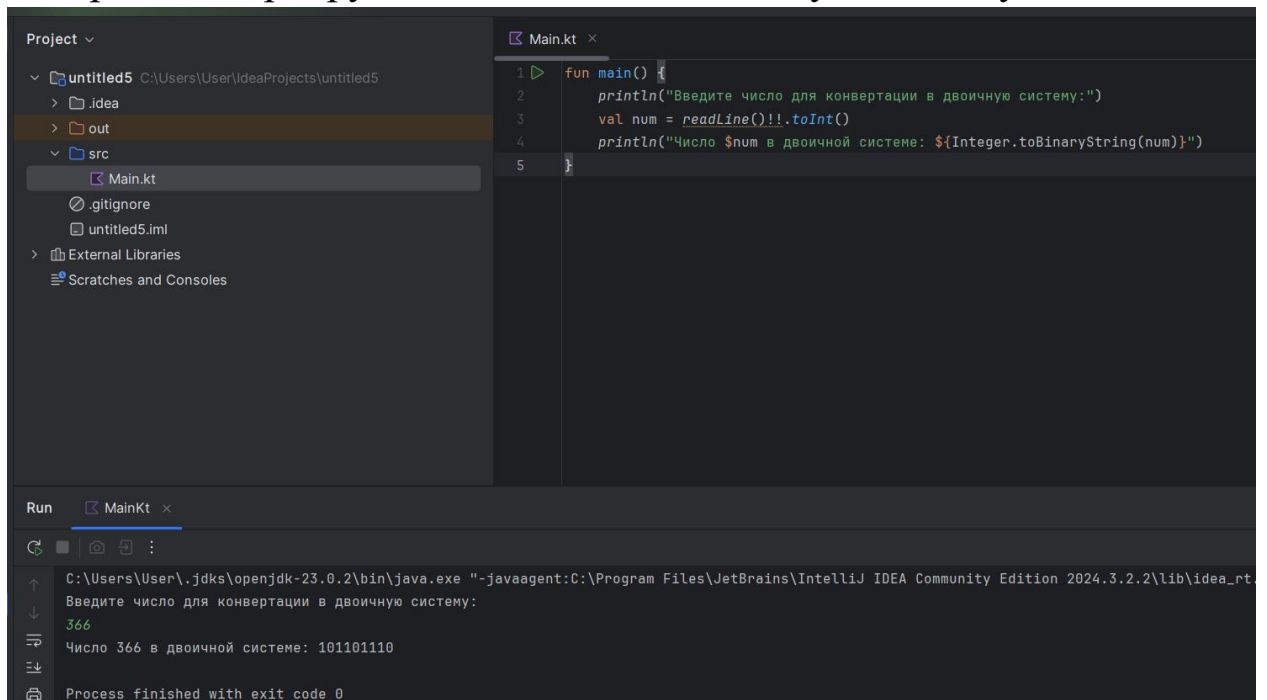
The screenshot shows the IntelliJ IDEA IDE with a project named 'untitled5'. The 'src' directory contains a file 'Main.kt'. The code in 'Main.kt' is as follows:

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

The 'Run' tab is active, showing the execution of the program. The command line shows the Java command being executed. The input '88' is entered, and the output is 'Сумма ряда: 5.0602235366812245'. The process finished with exit code 0.

```
C:\Users\User\.jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ I  
Введите число N:  
88  
Сумма ряда: 5.0602235366812245  
Process finished with exit code 0
```

30. Конвертация в двоичную систему: Напишите программу, которая конвертирует целое число в двоичную систему



The screenshot displays the IntelliJ IDEA interface. On the left, the Project Explorer shows a project named 'untitled5' with a source folder 'src' containing 'Main.kt'. The main editor window shows the following Kotlin code in 'Main.kt':

```
1 fun main() {  
2     println("Введите число для конвертации в двоичную систему:")  
3     val num = readLine()!!.toInt()  
4     println("Число $num в двоичной системе: ${Integer.toBinaryString(num)}")  
5 }
```

At the bottom, the Run console shows the execution of 'MainKt'. The output is as follows:

```
C:\Users\User\.jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.2.2\lib\idea_rt.jar" 366  
Введите число для конвертации в двоичную систему:  
366  
Число 366 в двоичной системе: 101101110  
Process finished with exit code 0
```