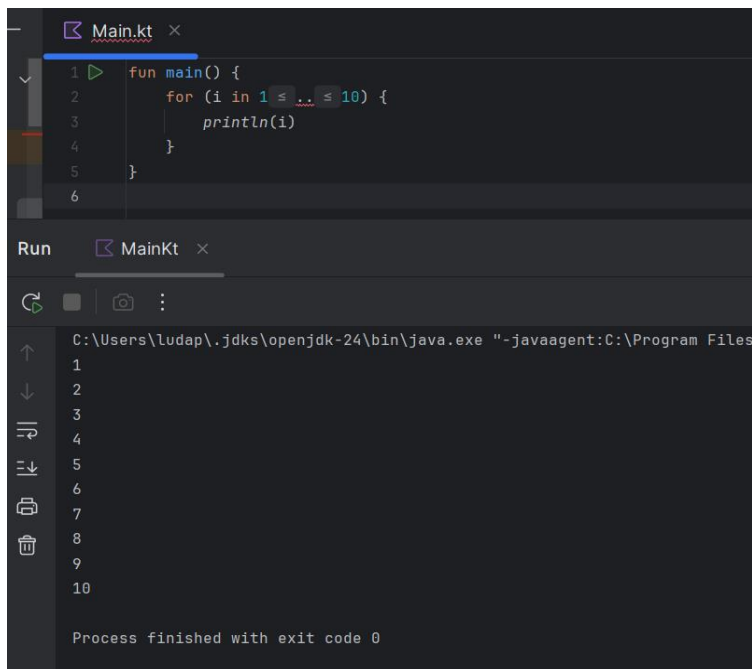


1. Вывод чисел от 1 до 10: Напишите программу, которая выводит числа от 1 до 10.

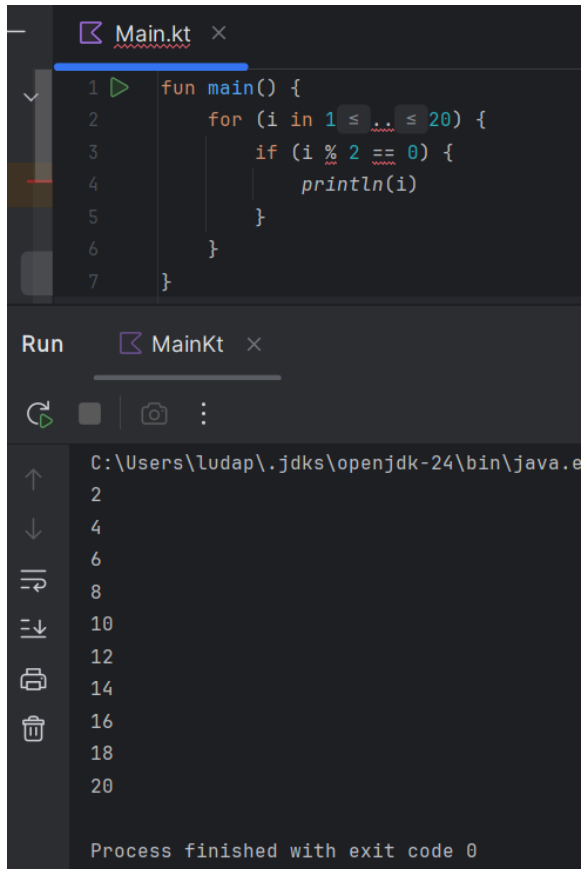


```
1 fun main() {  
2     for (i in 1..10) {  
3         println(i)  
4     }  
5 }  
6
```

Run MainKt

```
C:\Users\ludap\jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
  
Process finished with exit code 0
```

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

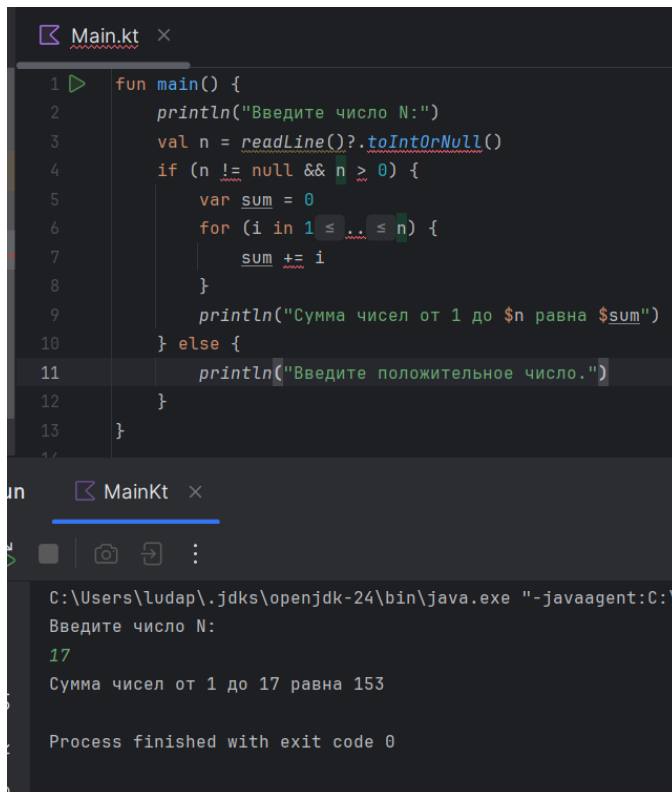


```
1 fun main() {  
2     for (i in 1..20) {  
3         if (i % 2 == 0) {  
4             println(i)  
5         }  
6     }  
7 }  
8
```

Run MainKt

```
C:\Users\ludap\jdk\openjdk-24\bin\java.e  
2  
4  
6  
8  
10  
12  
14  
16  
18  
20  
  
Process finished with exit code 0
```

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

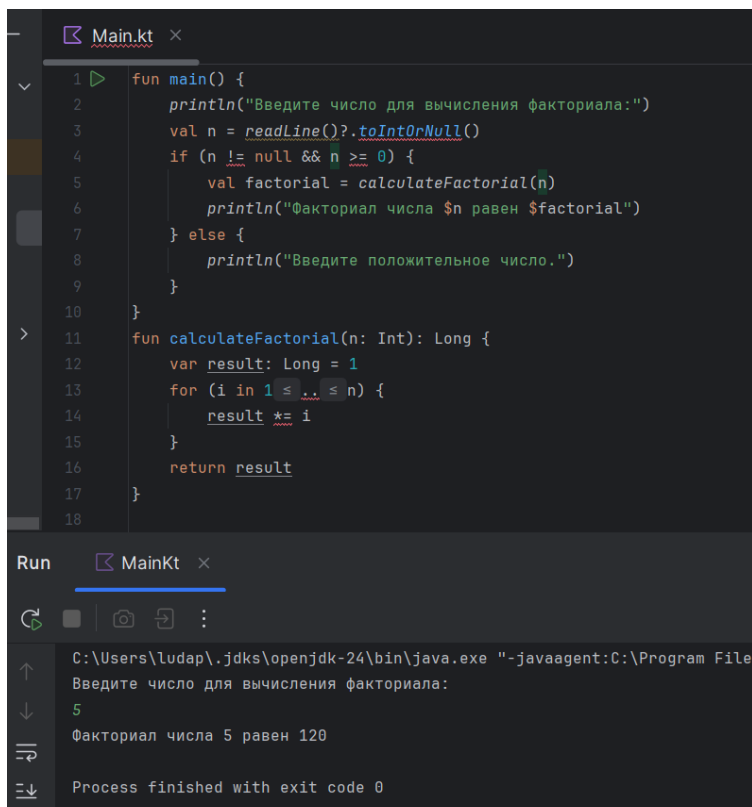


```
1 fun main() {
2     println("Введите число N:")
3     val n = readLine()?.toIntOrNull()
4     if (n != null && n > 0) {
5         var sum = 0
6         for (i in 1..n) {
7             sum += i
8         }
9         println("Сумма чисел от 1 до $n равна $sum")
10    } else {
11        println("Введите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

Run MainKt

C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\...
Введите число N:
17
Сумма чисел от 1 до 17 равна 153
Process finished with exit code 0

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

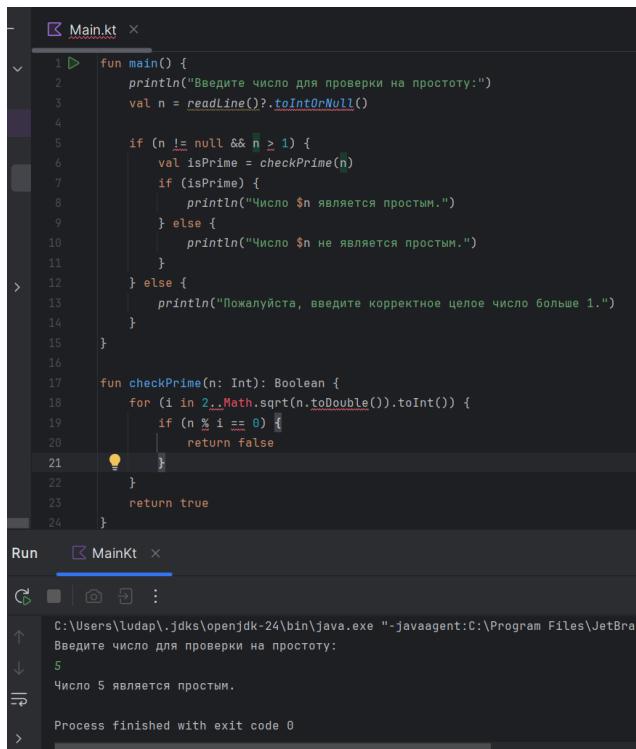


```
1 fun main() {
2     println("Введите число для вычисления факториала:")
3     val n = readLine()?.toIntOrNull()
4     if (n != null && n >= 0) {
5         val factorial = calculateFactorial(n)
6         println("Факториал числа $n равен $factorial")
7     } else {
8         println("Введите положительное число.")
9     }
10 }
11 fun calculateFactorial(n: Int): Long {
12     var result: Long = 1
13     for (i in 1..n) {
14         result *= i
15     }
16     return result
17 }
18 }
```

Run MainKt

C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files...
Введите число для вычисления факториала:
5
Факториал числа 5 равен 120
Process finished with exit code 0

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



```
1 fun main() {
2     println("Введите число для проверки на простоту:")
3     val n = readline()?.toIntOrNull()
4
5     if (n != null && n >= 1) {
6         val isPrime = checkPrime(n)
7         if (isPrime) {
8             println("Число $n является простым.")
9         } else {
10            println("Число $n не является простым.")
11        }
12    } else {
13        println("Пожалуйста, введите корректное целое число больше 1.")
14    }
15 }
16
17 fun checkPrime(n: Int): Boolean {
18     for (i in 2..Math.sqrt(n.toDouble()).toInt()) {
19         if (n % i == 0) {
20             return false
21         }
22     }
23     return true
24 }
```

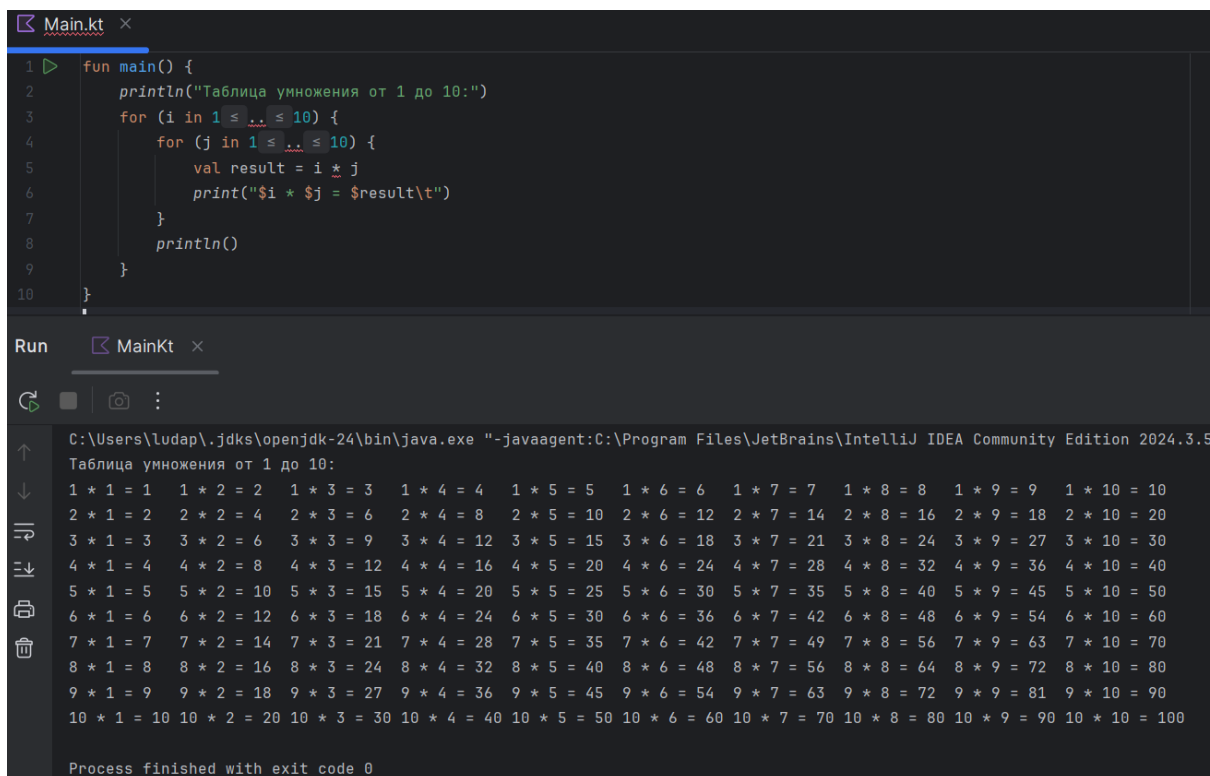
Run MainKt

C:\Users\ludap\jdk\openjdk-24\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.5\lib\idea_rt.jar=60257:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.5\bin -jar C:\Users\ludap\jdk\openjdk-24\bin\java.exe

Введите число для проверки на простоту:
5
Число 5 является простым.

Process finished with exit code 0

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



```
1 fun main() {
2     println("Таблица умножения от 1 до 10:")
3     for (i in 1..10) {
4         for (j in 1..10) {
5             val result = i * j
6             print("$i * $j = $result\t")
7         }
8         println()
9     }
10 }
```

Run MainKt

C:\Users\ludap\jdk\openjdk-24\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.5\lib\idea_rt.jar=60257:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.5\bin -jar C:\Users\ludap\jdk\openjdk-24\bin\java.exe

Таблица умножения от 1 до 10:

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

Process finished with exit code 0

7. Фибоначчи: Сгенерируйте первые N чисел Фибоначчи (где N вводит пользователь).

```
1 fun main() {
2     println("Введите некоторое количество чисел Фибоначчи:")
3     val n = readLine()?.toIntOrNull()
4     if (n != null && n >= 0) {
5         val fibonacciNumbers = generateFibonacci(n)
6         println("Первые $n чисел Фибоначчи: $fibonacciNumbers")
7     } else {
8         println("Введите положительное целое число.")
9     }
10 }
11 fun generateFibonacci(n: Int): List<Int> {
12     val fibonacciList = mutableListOf<Int>()
13     var a = 0
14     var b = 1
15     for (i in 1..n) {
16         fibonacciList.add(a)
17         val next = a + b
18         a = b
19         b = next
20     }
21     return fibonacciList
22 }
```

Run MainKt x

C:\Users\ludap\jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\Java\jdk-24\bin\javaagent.jar" -Dfile.encoding=UTF-8

Введите некоторое количество чисел Фибоначчи:
7
Первые 7 чисел Фибоначчи: [0, 1, 1, 2, 3, 5, 8]
Process finished with exit code 0

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

```
1 fun main() {
2     println("Введите первое число:")
3     val a = readLine()?.toIntOrNull()
4     println("Введите второе число:")
5     val b = readLine()?.toIntOrNull()
6     if (a != null && b != null && a >= 0 && b >= 0) {
7         val gcd = calculateGCD(a, b)
8         println("НОД чисел $a и $b равен $gcd")
9     } else {
10        println("Введите положительные целые числа.")
11    }
12 }
13 fun calculateGCD(a: Int, b: Int): Int {
14     return if (b == 0) {
15         a
16     } else {
17         calculateGCD(b, a % b)
18     }
19 }
```

MainKt x

C:\Users\ludap\jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\Java\jdk-24\bin\javaagent.jar" -Dfile.encoding=UTF-8

Введите первое число:
1505
Введите второе число:
5510
НОД чисел 1505 и 5510 равен 5
Process finished with exit code 0

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

```
1 fun main() {
2     println("Введите строку чисел:")
3     val input = readLine()
4     if (input != null) {
5         val reversed = input.reversed()
6         println("Обратный порядок: $reversed")
7     } else {
8         println("Введите корректную строку.")
9     }
10 }
```

run MainKt ×

C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-java
 Введите строку чисел:
 4 5 7 3 4 88 34 1 45
 Обратный порядок: 54 1 43 88 4 3 7 5 4
 Process finished with exit code 0

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

```
1 fun main() {
2     println("Введите число:")
3     val input = readLine()
4     if (input != null && input.all { it.isDigit() }) {
5         val sum = calculateSumOfDigits(input)
6         println("Сумма цифр числа $input равна $sum")
7     } else {
8         println("Введите целое число.")
9     }
10 }
11 fun calculateSumOfDigits(number: String): Int {
12     return number.map { it.toString().toInt() }.sum()
13 }
14
```

Run MainKt

C:\Users\ludap\jdk8\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea_rt.jar=1273.0:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea.jar" -Dfile.encoding=UTF-8

Введите число:
777

Сумма цифр числа 777 равна 21

Process finished with exit code 0

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

```
Main.kt x
1 fun main() {
2     println("Введите первую строку:")
3     val str1 = readlnOrNull() ?: ""
4     println("Введите вторую строку:")
5     val str2 = readlnOrNull() ?: ""
6     val areAnagrams = areAnagrams(str1, str2)
7     println(if (areAnagrams) "$str1 и $str2 являются анаграммами." else "$str1 и $str2 не являются анаграммами.")
8 }
9 fun areAnagrams(str1: String, str2: String): Boolean {
10     val cleanStr1 = str1.replace(" ", "").toLowerCase().toCharArray().sorted().joinToString(separator: "")
11     val cleanStr2 = str2.replace(" ", "").toLowerCase().toCharArray().sorted().joinToString(separator: "")
12     return cleanStr1 == cleanStr2
13 }
14 }

Run MainKt x
C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3
Введите первую строку:
удар
Введите вторую строку:
руда
удар и руда являются анаграммами.
Process finished with exit code 0
```

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

```
Main.kt x
1 fun main() {
2     println("Введите начальное число:")
3     val start = readLine()?.toIntOrNull() ?: return println("Неверный ввод.")
4     println("Введите шаг:")
5     val step = readLine()?.toIntOrNull() ?: return println("Неверный ввод.")
6     println("Введите количество элементов:")
7     val count = readLine()?.toIntOrNull() ?: return println("Неверный ввод.")
8     val sequence = List(count) { start + it * step }
9     println("Числовая последовательность: $sequence")
10 }
11 }

Run MainKt x
C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3
Введите начальное число:
0
Введите шаг:
5
Введите количество элементов:
15
Числовая последовательность: [0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70]
Process finished with exit code 0
```

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

```
1 fun main() {
2     println("Таблица квадратов от 1 до 20:")
3     for (i in 1..20) {
4         val square = i * i
5         println("$i\t$square")
6     }
7 }
```

Run MainKt

C:\Users\ludap\jdk\openjdk-24\bin\java.exe "-
Таблица квадратов от 1 до 20:
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

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

```
1 import kotlin.random.Random
2 fun main() {
3     println("10 случайных чисел:")
4     val randomNumbers = List(10) { Random.nextInt(1, 101) }
5     randomNumbers.forEach { println(it) }
6 }
```

Run MainKt

C:\Users\ludap\jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\Jet
10 случайных чисел:
45
43
63
61
15
11
57
18
32
37

Process finished with exit code 0

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

```
Main.kt x
1 fun main() {
2     println("Введите строку:")
3     val inputString = readlnOrNull() ?: ""
4     val isPalindrome = isPalindrome(inputString)
5     println(if (isPalindrome) "Строка '$inputString' является палиндромом." else "Строка '$inputString' не является палиндромом.")
6 }
7 fun isPalindrome(str: String): Boolean {
8     val cleanString = str.replace(" ", "").lowercase()
9     return cleanString == cleanString.reversed()
10 }
```

Run MainKt x

C:\Users\ludap\jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.5\lib\idea_...
Введите строку:
лидер бодро гордо бредил
Строка 'лидер бодро гордо бредил' является палиндромом.
Process finished with exit code 0

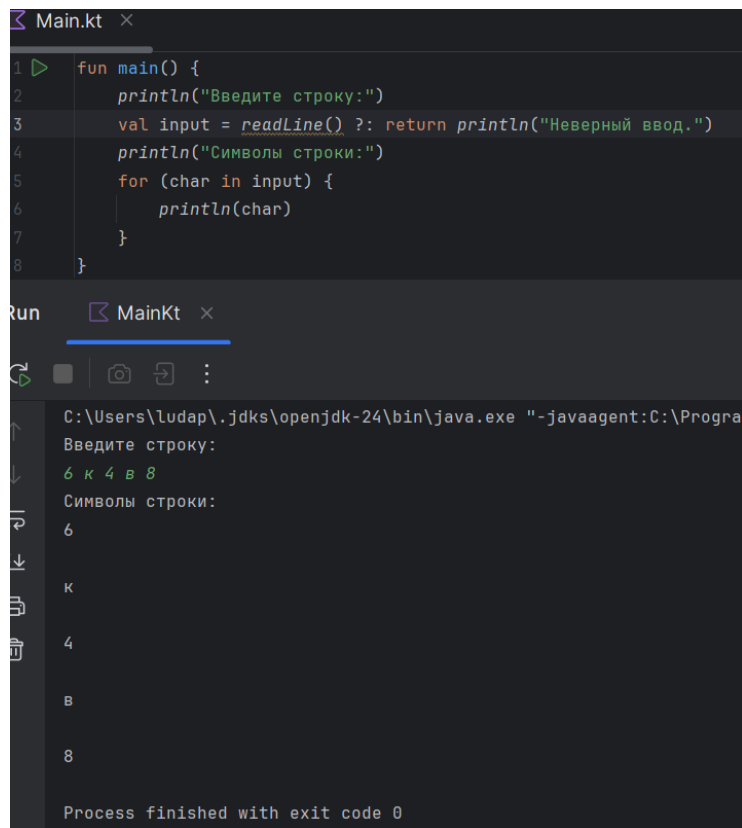
16. Сигма (сумма квадратов): Найдите сумму квадратов всех чисел от 1 до N.

```
Main.kt x
1 fun main() {
2     println("Введите число N:")
3     val n = readLine()?.toIntOrNull() ?: return println("Неверный ввод.")
4     val sumOfSquares = (1..n).sumOf { it * it }
5     println("Сигма всех чисел от 1 до $n равна $sumOfSquares.")
6 }
```

Run MainKt x

C:\Users\ludap\jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.5\lib\idea_...
Введите число N:
7
Сигма всех чисел от 1 до 7 равна 140.
Process finished with exit code 0

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



```
1 fun main() {
2     println("Введите строку:")
3     val input = readLine() ?: return println("Неверный ввод.")
4     println("Символы строки:")
5     for (char in input) {
6         println(char)
7     }
8 }
```

Run MainKt

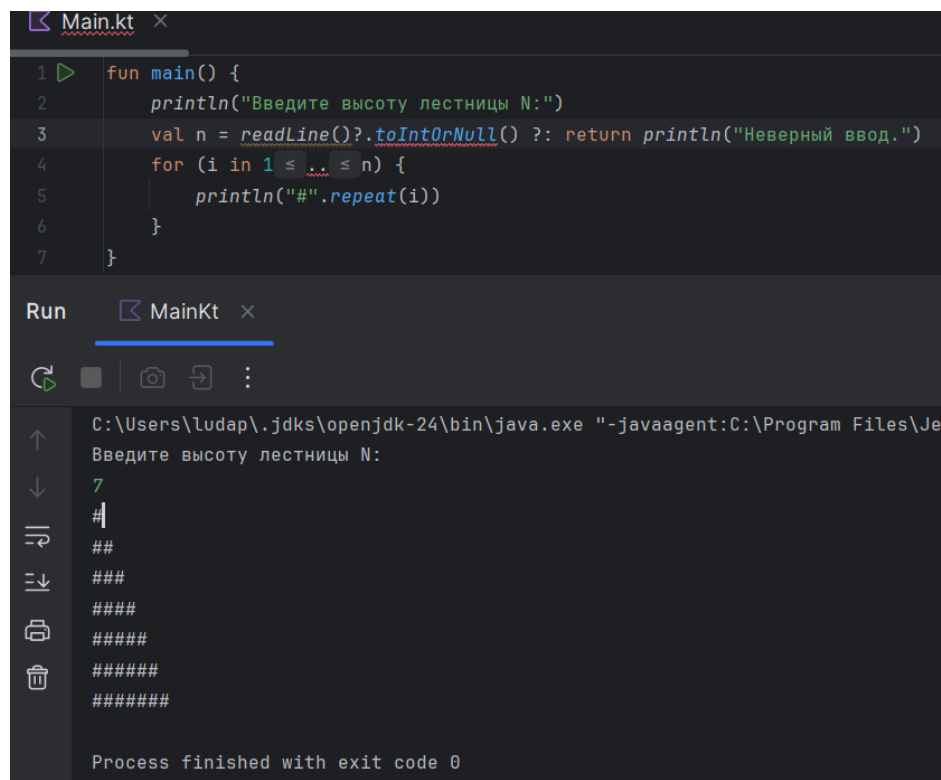
C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\Java\jdk-24\bin\javaagent.jar" -Dfile.encoding=UTF-8

Введите строку:
6 к 4 в 8

Символы строки:
6
к
4
в
8

Process finished with exit code 0

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



```
1 fun main() {
2     println("Введите высоту лестницы N:")
3     val n = readLine()?.toIntOrNull() ?: return println("Неверный ввод.")
4     for (i in 1..n) {
5         println("#".repeat(i))
6     }
7 }
```

Run MainKt

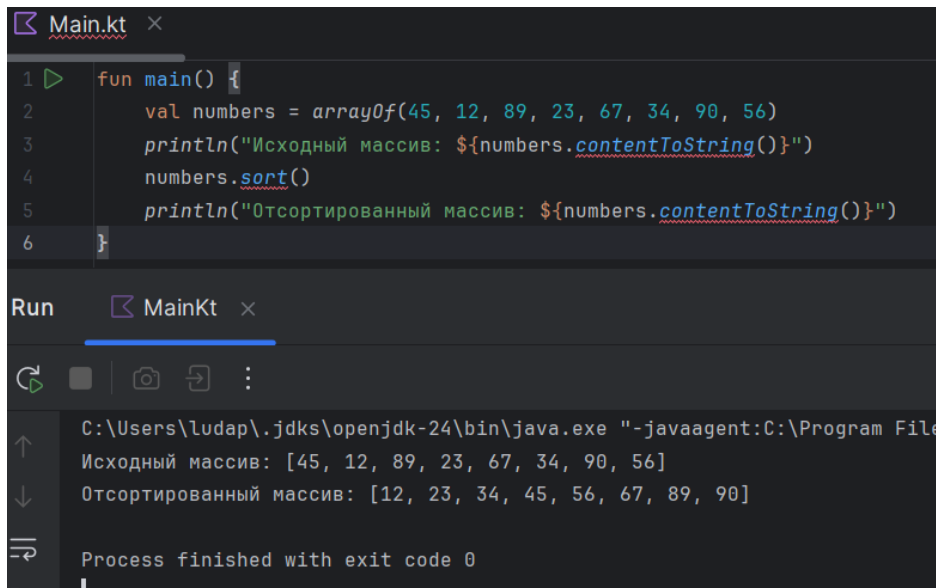
C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\Java\jdk-24\bin\javaagent.jar" -Dfile.encoding=UTF-8

Введите высоту лестницы N:
7

#####

Process finished with exit code 0

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



```
1 fun main() {
2     val numbers = arrayOf(45, 12, 89, 23, 67, 34, 90, 56)
3     println("Исходный массив: ${numbers.contentToString()}")
4     numbers.sort()
5     println("Отсортированный массив: ${numbers.contentToString()}")
6 }
```

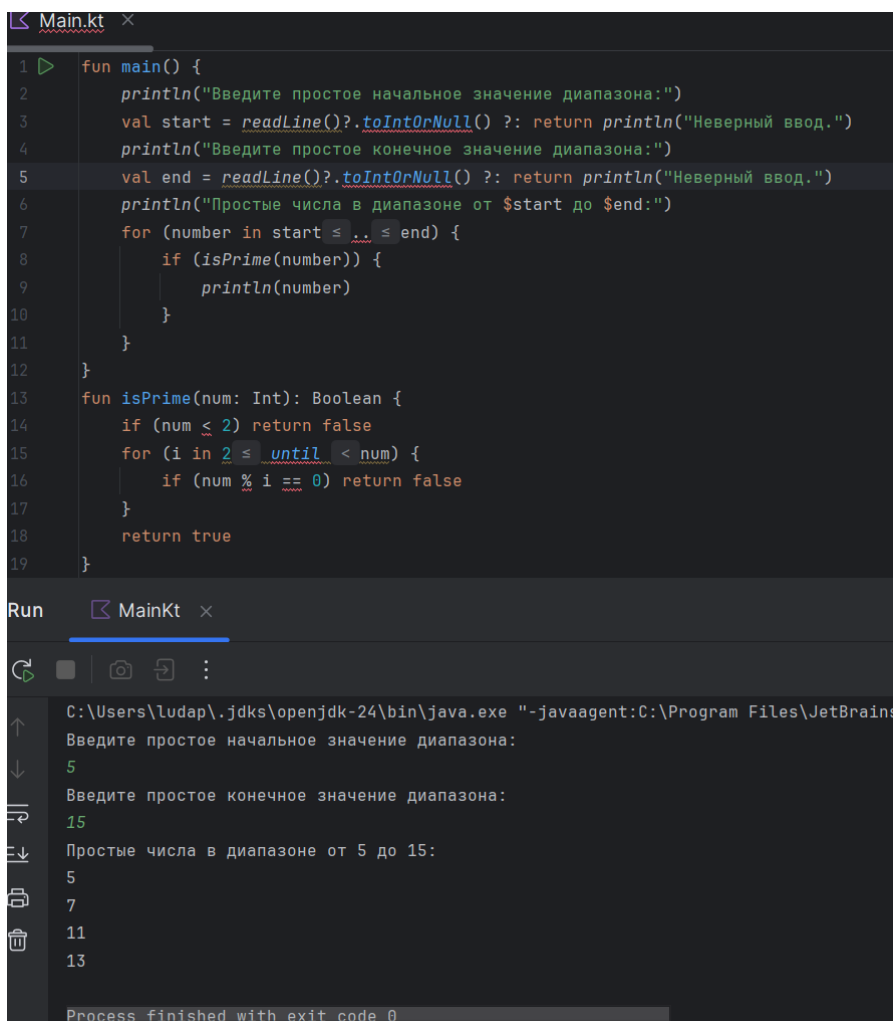
Run MainKt

C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\lib\idea_rt.jar=12738:C:\Users\ludap\.jdk\openjdk-24\bin\java.exe" -Dfile.encoding=UTF-8

Исходный массив: [45, 12, 89, 23, 67, 34, 90, 56]
Отсортированный массив: [12, 23, 34, 45, 56, 67, 89, 90]

Process finished with exit code 0

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



```
1 fun main() {
2     println("Введите простое начальное значение диапазона:")
3     val start = readLine()?.toIntOrNull() ?: return println("Неверный ввод.")
4     println("Введите простое конечное значение диапазона:")
5     val end = readLine()?.toIntOrNull() ?: return println("Неверный ввод.")
6     println("Простые числа в диапазоне от $start до $end:")
7     for (number in start..end) {
8         if (isPrime(number)) {
9             println(number)
10        }
11    }
12 }
13 fun isPrime(num: Int): Boolean {
14     if (num <= 2) return false
15     for (i in 2..until(num)) {
16         if (num % i == 0) return false
17     }
18     return true
19 }
```

Run MainKt

C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\lib\idea_rt.jar=12738:C:\Users\ludap\.jdk\openjdk-24\bin\java.exe" -Dfile.encoding=UTF-8

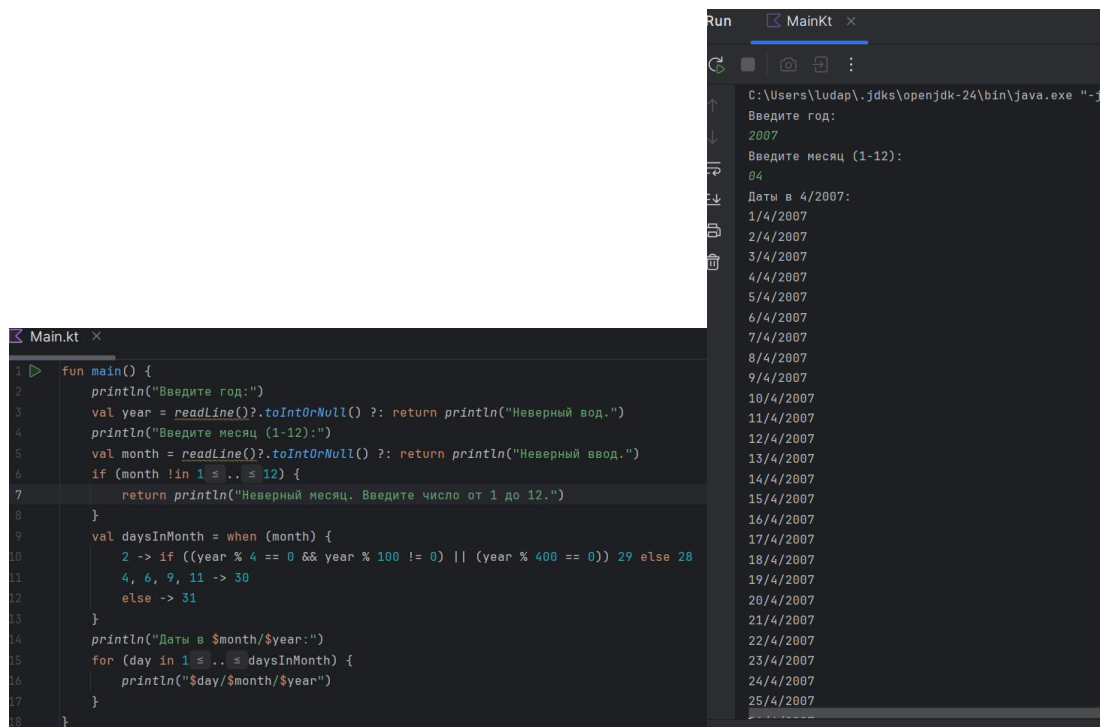
Введите простое начальное значение диапазона:
5

Введите простое конечное значение диапазона:
15

Простые числа в диапазоне от 5 до 15:
5
7
11
13

Process finished with exit code 0

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



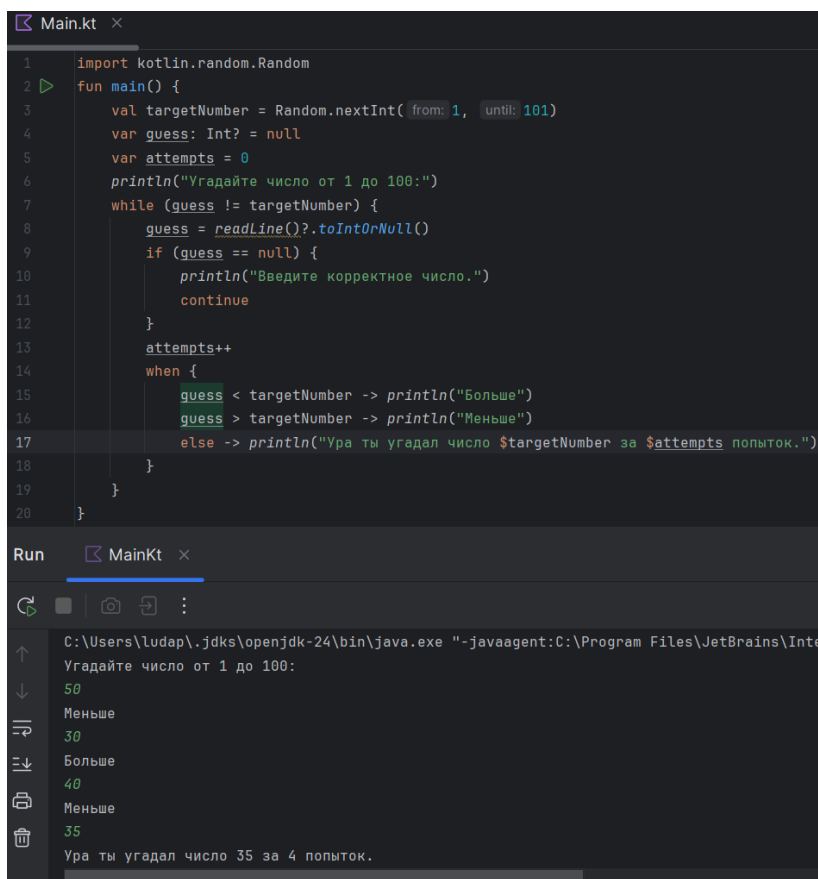
The screenshot shows an IDE with two windows. The left window, titled 'Main.kt', contains the following Kotlin code:

```
1 fun main() {
2     println("Введите год:")
3     val year = readLine()?.toIntOrNull() ?: return println("Неверный ввод.")
4     println("Введите месяц (1-12):")
5     val month = readLine()?.toIntOrNull() ?: return println("Неверный ввод.")
6     if (month !in 1..12) {
7         return println("Неверный месяц. Введите число от 1 до 12.")
8     }
9     val daysInMonth = when (month) {
10        2 -> if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) 29 else 28
11        4, 6, 9, 11 -> 30
12        else -> 31
13    }
14    println("Даты в $month/$year:")
15    for (day in 1..daysInMonth) {
16        println("$day/$month/$year")
17    }
18 }
```

The right window, titled 'Run', shows the execution output:

```
C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-j...
Введите год:
2007
Введите месяц (1-12):
04
Даты в 4/2007:
1/4/2007
2/4/2007
3/4/2007
4/4/2007
5/4/2007
6/4/2007
7/4/2007
8/4/2007
9/4/2007
10/4/2007
11/4/2007
12/4/2007
13/4/2007
14/4/2007
15/4/2007
16/4/2007
17/4/2007
18/4/2007
19/4/2007
20/4/2007
21/4/2007
22/4/2007
23/4/2007
24/4/2007
25/4/2007
```

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



The screenshot shows an IDE with two windows. The left window, titled 'Main.kt', contains the following Kotlin code:

```
1 import kotlin.random.Random
2 fun main() {
3     val targetNumber = Random.nextInt(from: 1, until: 101)
4     var guess: Int? = null
5     var attempts = 0
6     println("Угадайте число от 1 до 100:")
7     while (guess != targetNumber) {
8         guess = readLine()?.toIntOrNull()
9         if (guess == null) {
10            println("Введите корректное число.")
11            continue
12        }
13        attempts++
14        when {
15            guess < targetNumber -> println("Больше")
16            guess > targetNumber -> println("Меньше")
17            else -> println("Ура ты угадал число $targetNumber за $attempts попыток.")
18        }
19    }
20 }
```

The right window, titled 'Run', shows the execution output:

```
C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea-agent.jar"
Угадайте число от 1 до 100:
50
Меньше
30
Больше
40
Меньше
35
Ура ты угадал число 35 за 4 попыток.
```

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".

```
1 fun main() {
2     while (true) {
3         println("Введите первую цифру (или 'уляля' чтобы завершить):")
4         val input1 = readLine() ?: continue
5         if (input1.lowercase() == "уляля") break
6         println("Введите вторую цифру (или 'уляля' чтобы завершить):")
7         val input2 = readLine() ?: continue
8         if (input2.lowercase() == "стоп") break
9         val number1 = input1.toIntOrNull()
10        val number2 = input2.toIntOrNull()
11        if (number1 == null || number2 == null) {
12            println("Введите верные цифры.")
13            continue
14        }
15        println("Выберите действие: 1 - сложение, 2 - умножение")
16        val action = readLine()
17        when (action) {
18            "1" -> println("Результат: ${number1 + number2}")
19            "2" -> println("Результат: ${number1 * number2}")
20            else -> println("Неверное действие.")
21        }
22    }
23    println("Завершено.")
24 }
```

C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\Java\jdk-24\bin\javaagent.jar" -Djconsole.port=5044 -Djconsole.enabled=true -Djconsole.url=service:jmx:rmi:///jndi/rmi://localhost:5044/jmx-console --add-exports java.desktop/com.sun.java.swing=ALL-UNNAMED --add-exports java.desktop/com.sun.java.swing.internal=ALL-UNNAMED --add-exports java.desktop/com.sun.java.swing.internal=ALL-UNNAMED --add-exports java.desktop/com.sun.java.swing=ALL-UNNAMED --add-exports java.desktop/com.sun.java.swing.internal=ALL-UNNAMED --add-exports java.desktop/com.sun.java.swing=ALL-UNNAMED --add-exports java.desktop/com.sun.java.swing.internal=ALL-UNNAMED --add-exports java.desktop/com.sun.java.swing=ALL-UNNAMED --add-exports java.desktop/com.sun.java.swing.internal=ALL-UNNAMED --add-exports java.desktop/com.sun.java.swing=ALL-UNNED

Введите первую цифру (или 'уляля' чтобы завершить):
23
Введите вторую цифру (или 'уляля' чтобы завершить):
15
Выберите действие: 1 - сложение, 2 - умножение
1
Результат: 38
Введите первую цифру (или 'уляля' чтобы завершить):
уляля
Завершено.
Process finished with exit code 0

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

```
1 fun main() {
2     val matrix = arrayOf(
3         arrayOf(1, 2, 3),
4         arrayOf(4, 5, 6),
5         arrayOf(7, 8, 9)
6     )
7     val transposed = Array(matrix[0].size) { Array(matrix.size) { 0 } }
8     for (i in matrix.indices) {
9         for (j in matrix[i].indices) {
10            transposed[j][i] = matrix[i][j]
11        }
12    }
13    println("Транспонированная матрица:")
14    transposed.forEach { println(it.joinToString(separator: " ")) }
15 }
```

Run MainKt x

C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\Java\jdk-24\bin\javaagent.jar" -Djconsole.port=5044 -Djconsole.enabled=true -Djconsole.url=service:jmx:rmi:///jndi/rmi://localhost:5044/jmx-console --add-exports java.desktop/com.sun.java.swing=ALL-UNNED

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.

```
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 }
```

Run MainKt x

C:\Users\ludap\.jdk\openjdk-24\bin\java
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. Сумма четных и нечетных чисел: Пользователь вводит N, программа считает сумму четных и нечетных чисел от 1 до N.

```
1 fun main() {
2     println("Введите число N:")
3     val input = readLine()?.toIntOrNull() ?: return println("Неверный ввод.")
4     var evenSum = 0
5     var oddSum = 0
6     for (i in 1..input) {
7         if (i % 2 == 0) evenSum += i else oddSum += i
8     }
9     println("Сумма четных чисел: $evenSum")
10    println("Сумма нечетных чисел: $oddSum")
11 }
```

Run MainKt x

C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\JetBra
Введите число N:
7
Сумма четных чисел: 12
Сумма нечетных чисел: 16
Process finished with exit code 0

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

```
1 fun main() {
2     println("Введите число N:")
3     val input = readLine()?.toIntOrNull() ?: return println("Неверный ввод.")
4     for (i in 1 .. input) {
5         repeat(times: input - i) {
6             print(" ")
7         }
8         for (j in 1 .. i) {
9             print("$j ")
10        }
11        println()
12    }
13 }
```

Run MainKt x

C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\jetbrains-agent.jar" -Dfile.encoding=UTF-8

Введите число N:

10

```

      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
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 10
```

Process finished with exit code 0

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

```
1 fun main() {
2     println("Введите N чисел :")
3     val n = readLine()?.toIntOrNull() ?: return println("Неверный ввод.")
4     val numbers = mutableListOf<Int>()
5     println("Введите $n чисел:")
6     repeat(n) {
7         val number = readLine()?.toIntOrNull() ?: return println("Неверный ввод.")
8         numbers.add(number)
9     }
10    numbers.sort()
11    println("Числа в порядке возрастания:")
12    println(numbers.joinToString(separator: " "))
13 }
```

Run MainKt x

C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\jetbrains-agent.jar" -Dfile.encoding=UTF-8

Введите N чисел :

3

Введите 3 чисел:

8
3
5

Числа в порядке возрастания:

3, 5, 8

Process finished with exit code 0

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

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

Run MainKt x

C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea_rt.jar=12738:C:\Program Files\JetBrains\IntelliJ IDEA\bin\ -Dfile.encoding=UTF-8"

Введите число N:
7
Сумма ряда: 2.5928571428571425
Process finished with exit code 0

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

```
1 fun main() {
2     println("Введите целое число:")
3     val number = readLine()?.toIntOrNull() ?: return println("Неверный ввод.")
4     val binary = number.toString(radix: 2)
5     println("Двоичная система: $binary")
6 }
7
```

Run MainKt x

C:\Users\ludap\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea_rt.jar=12738:C:\Program Files\JetBrains\IntelliJ IDEA\bin\ -Dfile.encoding=UTF-8"

Введите целое число:
77
Двоичная система: 1001101
Process finished with exit code 0