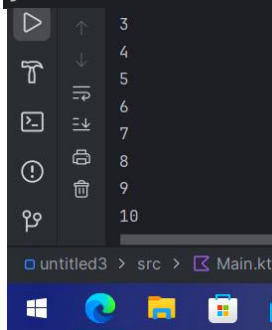


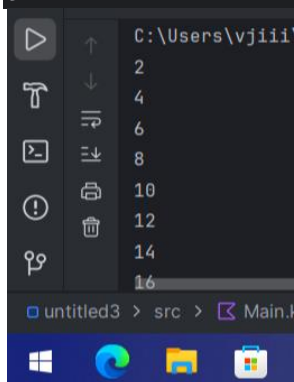
1 Вывод чисел от 1 до 10: Напишите программу, которая выводит числа от 1 до 10.

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



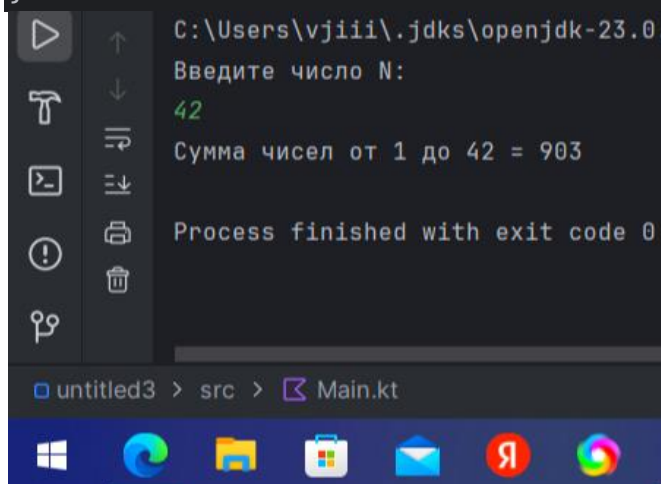
2 Вывод четных чисел от 1 до 20: Напишите программу, которая выводит все четные числа от 1 до 20.

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



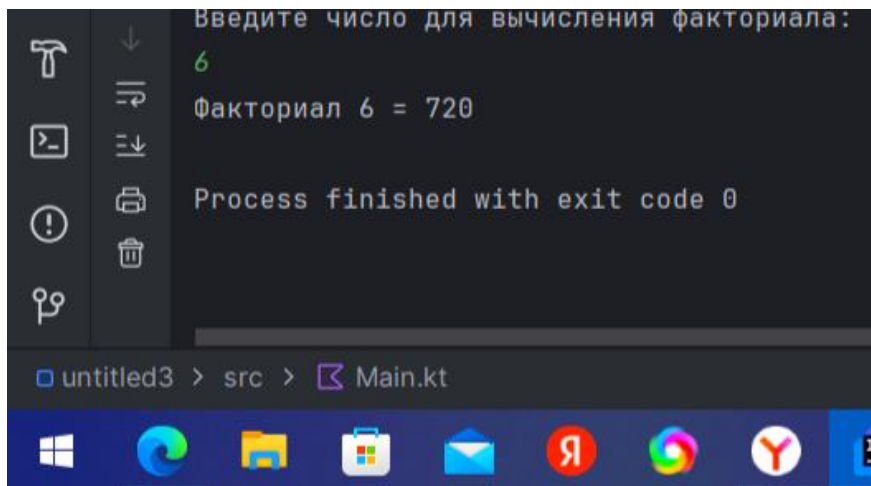
3 Сумма чисел от 1 до N: Пользователь вводит число N, программа суммирует все числа от 1 до N и выводит результат.

```
fun main() {  
    println("Введите число N:")  
    val n = readLine()!!.toInt()  
    val sum = (1..n).sum()  
    println("Сумма чисел от 1 до $n = $sum")  
}
```



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

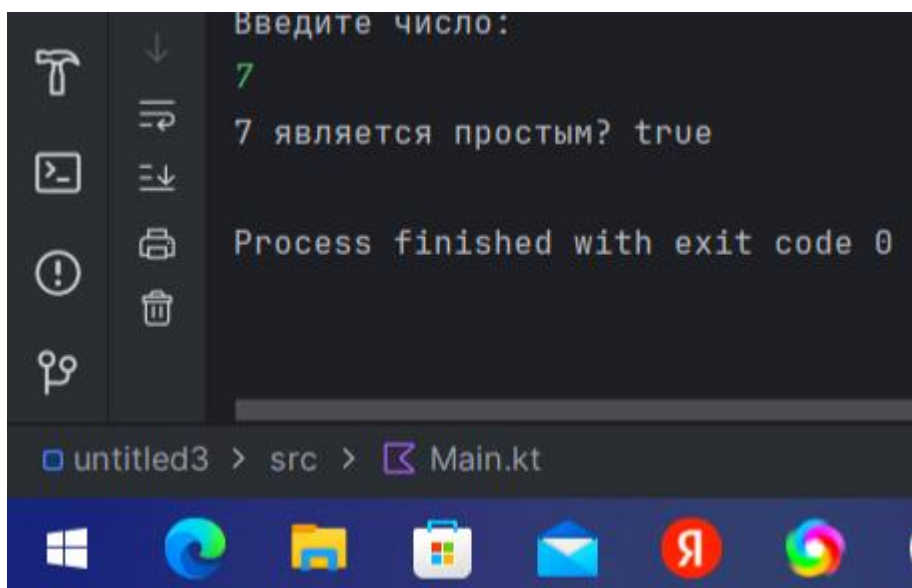
```
fun factorial(n: Int): Int {  
    return if (n == 0) 1 else n * factorial(n - 1)  
}  
  
fun main() {  
    println("Введите число для вычисления факториала:")  
    val number = readLine()!!.toInt()  
    println("Факториал $number = ${factorial(number)}")  
}
```



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

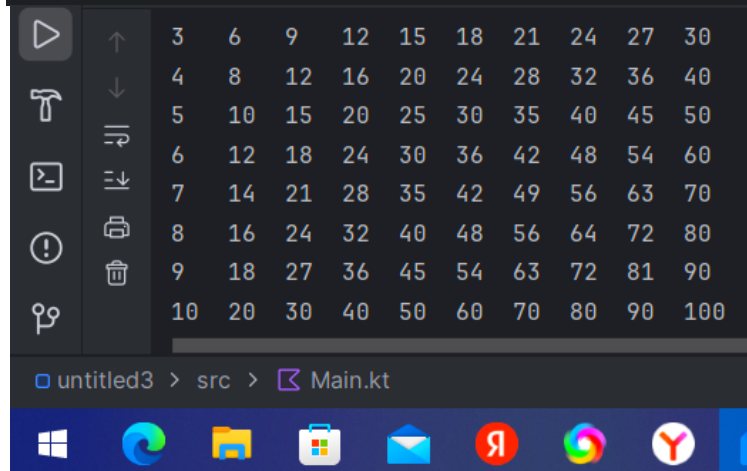
```
fun isPrime(num: Int): Boolean {
    if (num <= 1) return false
    for (i in 2 until num) {
        if (num % i == 0) return false
    }
    return true
}

fun main() {
    println("Введите число:")
    val number = readLine()!!.toInt()
    println("$number является простым? ${isPrime(number)}")
}
```



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

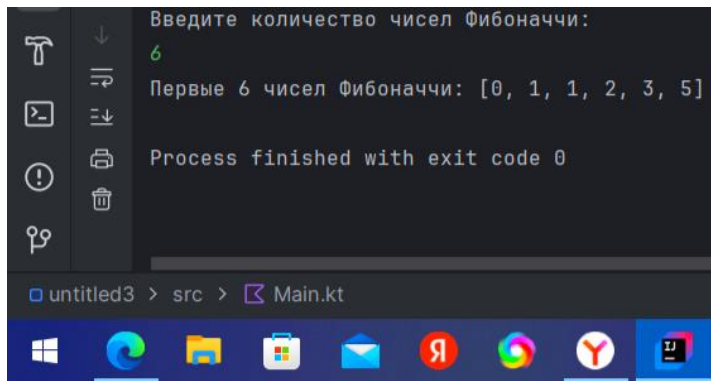
```
fun main() {
    for (i in 1..10) {
        for (j in 1..10) {
            print("${i * j}\t")
        }
        println()
    }
}
```



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

```
fun fibonacci(n: Int): List<Int> {
    val fibList = mutableListOf(0, 1)
    for (i in 2 until n) {
        fibList.add(fibList[i - 1] + fibList[i - 2])
    }
    return fibList.take(n)
}

fun main() {
    println("Введите количество чисел Фибоначчи:")
    val n = readLine()!!.toInt()
    println("Первые $n чисел Фибоначчи: ${fibonacci(n)}")
}
```

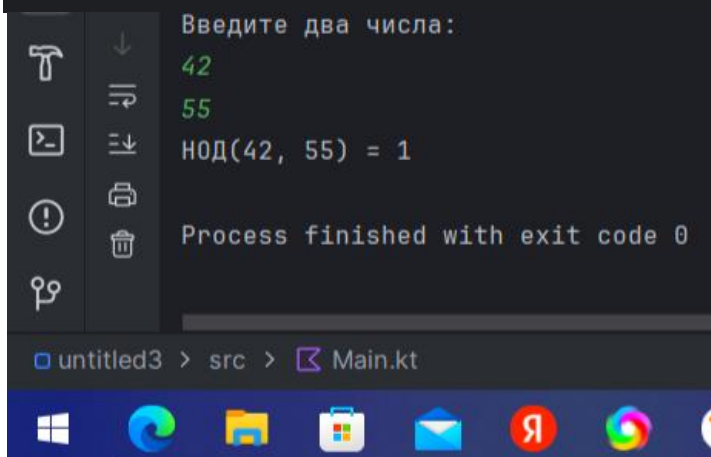


Введите количество чисел Фибоначчи:
6
Первые 6 чисел Фибоначчи: [0, 1, 1, 2, 3, 5]
Process finished with exit code 0

untitled3 > src > Main.kt

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

```
fun gcd(a: Int, b: Int): Int {  
    return if (b == 0) a else gcd(b, a % b)  
}  
  
fun main() {  
    println("Введите два числа:")  
    val a = readLine()!!.toInt()  
    val b = readLine()!!.toInt()  
    println("НОД($a, $b) = ${gcd(a, b)}")  
}
```

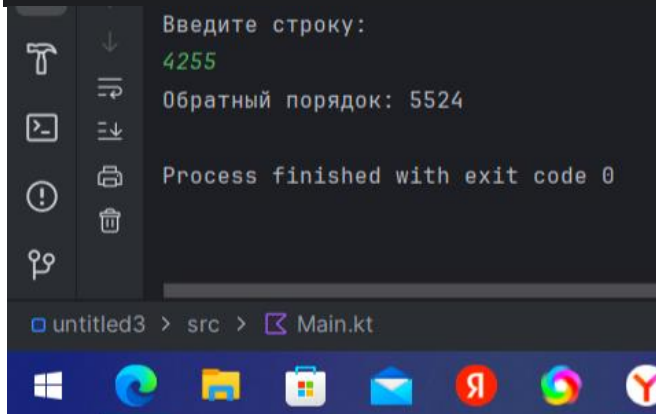


Введите два числа:
42
55
НОД(42, 55) = 1
Process finished with exit code 0

untitled3 > src > Main.kt

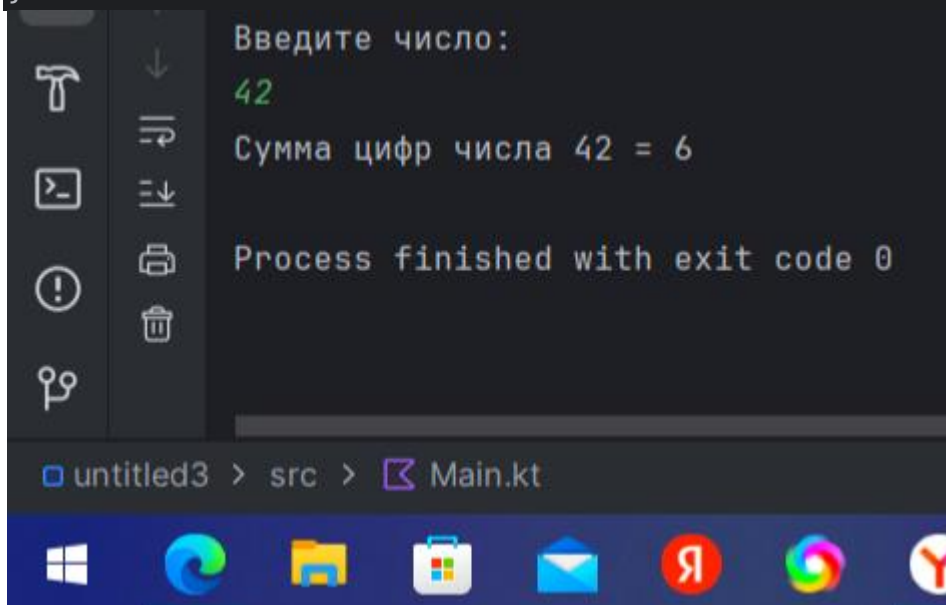
9. Обратный порядок: Пользователь вводит строку, и программа выводит ее в обратном порядке.

```
fun main() {
    println("Введите строку:")
    val input = readLine()!!
    println("Обратный порядок: ${input.reversed()}")
}
```



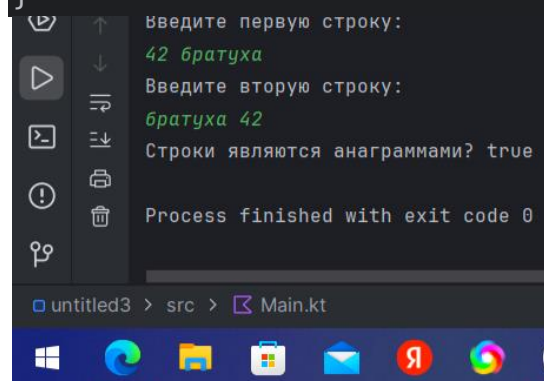
10. Сумма цифр числа: Напишите программу, которая находит сумму цифр введенного числа.

```
fun main() {
    println("Введите число:")
    val number = readLine()!!
    val sum = number.map { it.toString().toInt() }.sum()
    println("Сумма цифр числа $number = $sum")
}
```



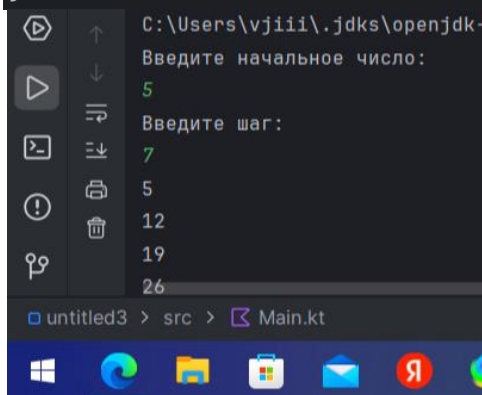
11. Анаграммы: Программа проверяет, являются ли две введенные строки анаграммами.

```
fun areAnagrams(str1: String, str2: String): Boolean {  
    return str1.toCharArray().sorted() ==  
    str2.toCharArray().sorted()  
}  
  
fun main() {  
    println("Введите первую строку:")  
    val str1 = readLine()!!  
    println("Введите вторую строку:")  
    val str2 = readLine()!!  
    println("Строки являются анаграммами? ${areAnagrams(str1,  
str2)}")  
}
```



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

```
fun main() {  
    println("Введите начальное число:")  
    val start = readLine()!!.toInt()  
    println("Введите шаг:")  
    val step = readLine()!!.toInt()  
  
    for (i in 0..10) { // Генерируем 10 элементов последовательности  
        println(start + i * step)  
    }  
}
```



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

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

```
C:\Users\vjiii\jdk\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\In  
1^2 = 1  
2^2 = 4  
3^2 = 9  
4^2 = 16  
5^2 = 25  
6^2 = 36  
7^2 = 49  
8^2 = 64  
9^2 = 81  
10^2 = 100  
11^2 = 121  
12^2 = 144  
13^2 = 169  
14^2 = 196  
15^2 = 225  
16^2 = 256  
17^2 = 289  
18^2 = 324  
19^2 = 361  
20^2 = 400
```


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

```
import kotlin.random.Random

fun main() {
    repeat(10) {
        println(Random.nextInt(1, 101)) // Случайное число от 1 до
100
    }
}
```

```
C:\Users\vjiii\jdk\openjdk-23.0.2\bin
80
58
98
5
19
5
2
46
1
28
```

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

```
fun isPalindrome(str: String): Boolean {
    return str == str.reversed()
}

fun main() {
    println("Введите строку:")
    val input = readLine()!!
    println("Строка является палиндромом? ${isPalindrome(input)}")
}
```

```
Введите строку.
шалаш
Строка является палиндромом? true

Process finished with exit code 0
```

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

```
fun main() {  
    println("Введите число N:")  
    val n = readLine()!!.toInt()  
    val sumOfSquares = (1..n).sumOf { it * it }  
    println("Сумма квадратов от 1 до $n = $sumOfSquares")  
}
```

Введите число N:

42

Сумма квадратов от 1 до 42 = 25585

Process finished with exit code 0

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

```
fun main() {  
    println("Введите строку:")  
    val input = readLine()!!  
  
    for (char in input) {  
        println(char)  
    }  
}
```

Введите строку:

1234567890

1

2

3

4

5

6

7

8

9

0

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

```
fun main() {
    println("Введите высоту лестницы N:")
    val n = readLine()!!.toInt()

    for (i in 1..n) {
        println("#".repeat(i))
    }
}
```

[illegible]

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

```
fun main() {  
    val numbers = mutableListOf(34, 12, 45, 23, 56, 78, 11, 99)  
  
    for (i in numbers.indices) {  
        for (j in i + 1 until numbers.size) {  
            if (numbers[i] > numbers[j]) {  
                val temp = numbers[i]  
                numbers[i] = numbers[j]  
                numbers[j] = temp  
            }  
        }  
    }  
}
```

```
println("Отсортированный список: $numbers")
}

Отсортированный список: [11, 12, 23, 34, 45, 56, 78, 99]

Process finished with exit code 0
```

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

```
fun isPrime(num: Int): Boolean {
    if (num <= 1) return false
    for (i in 2 until num) {
        if (num % i == 0) return false
    }
    return true
}

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

    println("Простые числа в диапазоне от $start до $end:")
    for (i in start..end) {
        if (isPrime(i)) {
            println(i)
        }
    }
}
```

```
Введите начальное число диапазона:
1
Введите конечное число диапазона:
55
Простые числа в диапазоне от 1 до 55:
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
```

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

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

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

    val yearMonth = YearMonth.of(year, month)
    val daysInMonth = yearMonth.lengthOfMonth()

    println("Даты в месяце $month/$year:")
    for (day in 1..daysInMonth) {
        println(LocalDate.of(year, month, day))
    }
}
```

```
Введите год:
2049
Введите месяц:
7
Даты в месяце 7/2049:
2049-07-01
2049-07-02
2049-07-03
2049-07-04
2049-07-05
2049-07-06
2049-07-07
2049-07-08
2049-07-09
2049-07-10
2049-07-11
2049-07-12
2049-07-13
2049-07-14
```

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

```
import kotlin.random.Random

fun main() {
    val numberToGuess = Random.nextInt(1, 101)
    var guess: Int? = null

    println("Угадайте число от 1 до 100:")

    while (guess != numberToGuess) {
        guess = readLine()!!.toInt()
        when {
            guess < numberToGuess -> println("Слишком низко! Попробуйте снова.")
            guess > numberToGuess -> println("Слишком высоко! Попробуйте снова.")
            else -> println("Поздравляю! Вы угадали число!")
        }
    }
}
```

```
}
```

```
Угадайте число от 1 до 100:
42
Слишком высоко! Попробуйте снова.
30
Слишком низко! Попробуйте снова.
35
Слишком низко! Попробуйте снова.
37
Слишком низко! Попробуйте снова.
39
Поздравляю! Вы угадали число!
```

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".

```
fun main() {
    println("Введите первую цифру:")
    val firstNumber = readLine()!!.toDouble()
    println("Введите вторую цифру:")
    val secondNumber = readLine()!!.toDouble()

    while (true) {
        println("Введите 'сложение' для сложения или 'умножение' для
умножения или 'стоп' для выхода:")
        val operation = readLine()!!

        if (operation == "стоп") break

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

```

        "умножение" -> println("Результат умножения:
${firstNumber * secondNumber}")
        else -> println("Некорректная операция.")
    }
}
}

```

```

Введите первую цифру:
6
Введите вторую цифру:
8
Введите 'сложение' для сложения или 'умножение' для умножения или 'стоп' для выхода:
сложение
Результат сложения: 14.0
Введите 'сложение' для сложения или 'умножение' для умножения или 'стоп' для выхода:
умножение
Результат умножения: 48.0
Введите 'сложение' для сложения или 'умножение' для умножения или 'стоп' для выхода:
стоп

```

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

```

fun main() {
    println("Введите количество строк матрицы:")
    val rows = readLine()!!.toInt()
    println("Введите количество столбцов матрицы:")
    val cols = readLine()!!.toInt()
    val matrix = Array(rows) { IntArray(cols) }
    println("Введите элементы матрицы:")
    for (i in 0 until rows) {
        for (j in 0 until cols) {
            matrix[i][j] = readLine()!!.toInt()
        }
    }
    val transposedMatrix = Array(cols) { IntArray(rows) }

    for (i in 0 until rows) {
        for (j in 0 until cols) {

```



```

        transposedMatrix[j][i] = matrix[i][j]
    }
}
println("Транспонированная матрица:")
for (row in transposedMatrix) {
    println(row.joinToString(" "))
}
}

```

```

Введите количество строк матрицы:
3
Введите количество столбцов матрицы:
3
Введите элементы матрицы:
1
2
3
4
5
6
7
8
9
Транспонированная матрица:
1 4 7
2 5 8
3 6 9

```

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

```

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

```

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
```

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

```
fun main() {
    println("Введите число N:")
    val n = readLine()!!.toInt()

    var evenSum = 0
    var oddSum = 0

    for (i in 1..n) {
        if (i % 2 == 0) {
            evenSum += i
        } else {
            oddSum += i
        }
    }

    println("Сумма четных чисел от 1 до $n = $evenSum")
    println("Сумма нечетных чисел от 1 до $n = $oddSum")
}
```

Введите число N:

42

Сумма четных чисел от 1 до 42 = 462

Сумма нечетных чисел от 1 до 42 = 441

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

```
fun main() {
    println("Введите число N:")
}
```

```

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

for (i in 1..n) {
    println(" ".repeat(n - i) + (1..i).joinToString(" "))
}

```

Введите число N:

```

9
      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

```

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

```

fun main() {
    println("Введите количество чисел:")
    val n = readLine()!!.toInt()
    val numbers = mutableListOf<Int>()
    println("Введите $n чисел:")
    repeat(n) {
        numbers.add(readLine()!!.toInt())
    }
    numbers.sort()
    println("Числа в порядке возрастания: $numbers")
}

```

Введите количество чисел:

```

5
Введите 5 чисел:
4
7
23
76
42
Числа в порядке возрастания: [4, 7, 23, 42, 76]

```

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

```
fun main() {  
    println("Введите число N:")  
    val n = readLine()!!.toInt()  
  
    val sum = (1..n).sumOf { 1.0 / it }  
  
    println("Сумма ряда от 1 до $n: $sum")  
}
```

Введите число N:

6

Сумма ряда от 1 до 6: 2.4499999999999997

Process finished with exit code 0

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

```
fun main() {  
    println("Введите целое число:")  
    val number = readLine()!!.toInt()  
  
    val binaryString = number.toString(2)
```

```
println("Двоичное представление числа $number: $binaryString")  
}
```

Введите целое число:

42

Двоичное представление числа 42: 101010

