

```
fun main() {
    // 1. Вывод чисел от 1 до 10
    for (i in 1..10) {
        println(i)
    }

    // 2. Вывод четных чисел от 1 до 20
    for (i in 1..20) {
        if (i % 2 == 0) {
            println(i)
        }
    }

    // 3. Сумма чисел от 1 до N
    println("Введите число N:")
    val n = readLine()?.toIntOrNull() ?: return
    val sum = (1..n).sum()
    println("Сумма чисел от 1 до $n: $sum")

    // 4. Факториал числа
    println("Введите число для вычисления факториала:")
    val number = readLine()?.toIntOrNull() ?: return
    var factorial = 1
    for (i in 1..number) {
        factorial *= i
    }
    println("Факториал числа $number: $factorial")

    // 5. Проверка числа на простоту
    println("Введите число для проверки на простоту:")
    val primeCandidate = readLine()?.toIntOrNull() ?: return
    val isPrime = (2 until primeCandidate).none { primeCandidate % it == 0 }
    println(if (isPrime) "$primeCandidate - простое число" else
"$primeCandidate - не простое число")

    // 6. Вывод таблицы умножения
    for (i in 1..10) {
        for (j in 1..10) {
            print("${i} * ${j} \t")
        }
        println()
    }

    // 7. Фибоначчи
    println("Введите количество чисел Фибоначчи для генерации:")
    val fibCount = readLine()?.toIntOrNull() ?: return
    var a = 0
    var b = 1
    for (i in 1..fibCount) {
        println(a)
        val next = a + b
        a = b
        b = next
    }

    // 8. НОД двух чисел
    println("Введите два числа для нахождения НОД:")
    val num1 = readLine()?.toIntOrNull() ?: return
    val num2 = readLine()?.toIntOrNull() ?: return
    fun gcd(a: Int, b: Int): Int = if (b == 0) a else gcd(b, a % b)
    println("НОД($num1, $num2) = ${gcd(num1, num2)}")
}
```

```

// 9. Обратный порядок строки
println("Введите строку для вывода в обратном порядке:")
val inputString = readLine() ?: return
println(inputString.reversed())

// 10. Сумма цифр числа
println("Введите число для нахождения суммы его цифр:")
val inputNumber = readLine()?.toIntOrNull() ?: return
val digitSum = inputNumber.toString().map { it.toString().toInt() }.sum()
println("Сумма цифр числа $inputNumber: $digitSum")

// 11. Анаграммы
println("Введите первую строку:")
val str1 = readLine() ?: return
println("Введите вторую строку:")
val str2 = readLine() ?: return
val areAnagrams = str1.toCharArray().sorted() ==
str2.toCharArray().sorted()
println(if (areAnagrams) "Строки являются анаграммами" else "Строки не
являются анаграммами")

// 12. Числовая последовательность
println("Введите начальное число:")
val startNum = readLine()?.toIntOrNull() ?: return
println("Введите шаг:")
val step = readLine()?.toIntOrNull() ?: return
for (i in startNum..startNum + step * 10 step step) {
    print("$i ")
}
println()

// 13. Таблица квадратов чисел от 1 до 20
for (i in 1..20) {
    println("$i^2 = ${i * i}")
}

// 14. Генерация случайных чисел
println("Случайные числа:")
repeat(10) {
    println((Math.random() * 101).toInt())
}

// 15. Проверка палиндрома
println("Введите строку для проверки на палиндром:")
val palindromeString = readLine() ?: return
val isPalindrome = palindromeString == palindromeString.reversed()
println(if (isPalindrome) "$palindromeString - палиндром" else
"$palindromeString - не палиндром")

// 16. Сигма (сумма квадратов)
println("Введите число N для нахождения суммы квадратов:")
val nForSquares = readLine()?.toIntOrNull() ?: return
val sumOfSquares = (1..nForSquares).sumOf { it * it }
println("Сумма квадратов от 1 до $nForSquares: $sumOfSquares")

// 17. Вывод символов строки по одному
println("Введите строку для вывода символов по одному:")
val lineToPrint = readLine() ?: return
for (char in lineToPrint) {
    println(char)
}

// 18. Лестница из символа "#"
println("Введите высоту лестницы:")

```

```

val height = readLine()?.toIntOrNull() ?: return
for (i in 1..height) {
    println("#".repeat(i))
}

// 19. Сортировка списка двухзначных чисел
val twoDigitNumbers = mutableListOf(34, 12, 45, 23, 56, 78, 11, 90)
for (i in twoDigitNumbers.indices) {
    for (j in i + 1 until twoDigitNumbers.size) {
        if (twoDigitNumbers[i] > twoDigitNumbers[j]) {
            val temp = twoDigitNumbers[i]
            twoDigitNumbers[i] = twoDigitNumbers[j]
            twoDigitNumbers[j] = temp
        }
    }
}
println("Отсортированный список двухзначных чисел: $twoDigitNumbers")

// 20. Простые числа в диапазоне
println("Введите диапазон для поиска простых чисел (начало и конец):")
val startRange = readLine()?.toIntOrNull() ?: return
val endRange = readLine()?.toIntOrNull() ?: return

for (num in startRange..endRange) {
    if (num > 1 && (2 until num).none { num % it == 0 }) {
        print("$num ")
    }
}

// 21. Вывод дат в месяце
println("Введите год и месяц для вывода всех дат в этом месяце:")
val yearInput = readLine()?.toIntOrNull() ?: return
val monthInput = readLine()?.toIntOrNull() ?: return
val daysInMonth = when(monthInput) {
    1,3,5,7,8,10,12 -> 31
    4,6,9,11 -> 30
    2 -> if ((yearInput % 4 == 0 && yearInput % 100 != 0) || (yearInput %
400 == 0)) 29 else 28
    else -> throw IllegalArgumentException("Некорректный месяц")
}
for(day in 1..daysInMonth) {
    println("$day/$monthInput/$yearInput")
}

//22. Игра "Угадай число"
val randomNum = (Math.random()*101).toInt()
var guess: Int?
do {
    println("Угадайте число от 1 до 100:")
    guess = readLine()?.toIntOrNull()
    when {
        guess == null -> continue
        guess < randomNum -> println("Слишком маленькое число!")
        guess > randomNum -> println("Слишком большое число!")
        else -> println("Поздравляю! Вы угадали число!")
    }
} while (guess != randomNum)

//23. Сложение и умножение с остановкой
var operation: String?
do {
    println("Введите две цифры через пробел или 'стоп' для выхода:")
    operation = readLine()
}

```

```

        if(operation != "стоп") {
            val parts = operation?.split(" ")?.map { it.toDouble() }
            if(parts?.size == 2) {
                val sumResult = parts[0] + parts[1]
                val mulResult = parts[0] * parts[1]
                println("Сложение: $sumResult, Умножение: $mulResult")
            } else {
                println("Некорректный ввод!")
            }
        }
    } while (operation != "стоп")

//24. Транспонирование матрицы
val matrix = arrayOf(
    arrayOf(1,2,3),
    arrayOf(4,5,6),
    arrayOf(7,8,9)
)
val transposedMatrix = Array(matrix[0].size) { IntArray(matrix.size) }
for(i in matrix.indices) {
    for(j in matrix[i].indices) {
        transposedMatrix[j][i] = matrix[i][j]
    }
}
println("Транспонированная матрица:")
for(row in transposedMatrix) {
    println(row.joinToString(", "))
}

//25. Кубы чисел от 1 до 10
for(i in 1..10) {
    println("$i^3 = ${i*i*i}")
}

//26. Сумма четных и нечетных чисел от 1 до N
println("Введите N для подсчета четных и нечетных чисел:")
val nForSumEvenOdd = readLine()?.toIntOrNull() ?: return
var evenSum = 0
var oddSum = 0
for(i in 1..nForSumEvenOdd) {
    if(i % 2 == 0) evenSum += i
    else oddSum += i
}
println("Сумма четных: $evenSum, Сумма нечетных: $oddSum")

//27. Печать "пирамида" из чисел от 1 до N
println("Введите N для печати пирамиды:")
val nForPyramid = readLine()?.toIntOrNull() ?: return
for(i in 1..nForPyramid) {
    repeat(nForPyramid - i) { print(" ") }
    repeat(i) { print("$i ") }
    println()
}

//28. Определение порядка N чисел
println("Введите количество чисел для сортировки:")
val countForSorting = readLine()?.toIntOrNull() ?: return
val numbersToSort = mutableListOf<Int>()
repeat(countForSorting) {
    println("Введите число ${it + 1}:")
    numbersToSort.add(readLine()?.toIntOrNull() ?: return)
}
numbersToSort.sort()
println("Отсортированные числа: $numbersToSort")

```

```
//29. Сумма ряда
println("Введите N для нахождения суммы ряда:")
val nForSeriesSum = readLine()?.toIntOrNull() ?: return
var seriesSum = (1..nForSeriesSum).sumOf { 1.0 / it }
println("Сумма ряда от 1 до $nForSeriesSum: $seriesSum")

//30. Конвертация в двоичную систему
println("Введите целое число для конвертации в двоичную систему:")
val numberToConvert = readLine()?.toIntOrNull() ?: return
val binaryRepresentation = numberToConvert.toString(2)
println("Двоичное представление числа $numberToConvert:
$binaryRepresentation")
}
```