

### 1. Вывод чисел от 1 до 10

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

### 2. Вывод четных чисел от 1 до 20

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

### 3. Сумма чисел от 1 до N

```
fun main() {  
    println("Введите число N:")  
    val n = readLine()?.toIntOrNull()  
    if (n != null) {  
        val sum = (1..n).sum()  
        println("Сумма чисел от 1 до $n: $sum")  
    } else {  
        println("Некорректный ввод.")  
    }  
}
```

#### 4. Факториал числа

```
fun factorial(n: Int): Long {  
    return if (n <= 1) 1 else n * factorial(n - 1)  
}  
  
fun main() {  
    println("Введите число для вычисления факториала:")  
    val n = readLine()?.toIntOrNull()  
    if (n != null && n >= 0) {  
        println("Факториал $n: ${factorial(n)}")  
    } else {  
        println("Некорректный ввод.")  
    }  
}
```

#### 5. Проверка числа на простоту

```
fun isPrime(n: Int): Boolean {  
    if (n <= 1) return false  
    for (i in 2..Math.sqrt(n.toDouble()).toInt()) {  
        if (n % i == 0) return false  
    }  
    return true  
}  
  
fun main() {  
    println("Введите число для проверки на простоту:")  
    val n = readLine()?.toIntOrNull()  
    if (n != null) {
```

```

    if (isPrime(n)) {
        println("$n является простым числом.")
    } else {
        println("$n не является простым числом.")
    }
} else {
    println("Некорректный ввод.")
}
}

```

## 6. Вывод таблицы умножения

```

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

```

## 7. Фибоначчи

```

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

```

```
}
```

```
fun main() {  
    println("Введите количество чисел Фибоначчи:")  
    val n = readLine()?.toIntOrNull()  
    if (n != null && n > 0) {  
        println("Первые $n чисел Фибоначчи: ${fibonacci(n)}")  
    } else {  
        println("Некорректный ввод.")  
    }  
}
```

## 8. Наибольший общий делитель (НОД)

```
fun gcd(a: Int, b: Int): Int {  
    return if (b == 0) a else gcd(b, a % b)  
}
```

```
fun main() {  
    println("Введите два числа для нахождения НОД:")  
    val a = readLine()?.toIntOrNull()  
    val b = readLine()?.toIntOrNull()  
  
    if (a != null && b != null) {  
        println("НОД($a, $b) = ${gcd(a, b)}")  
    } else {  
        println("Некорректный ввод.")  
    }  
}
```

## 9. Обратный порядок

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

## 10. Сумма цифр числа

```
fun sumOfDigits(n: Int): Int {  
    return n.toString().map { it.toString().toInt() }.sum()  
}
```

```
fun main() {  
    println("Введите число для нахождения суммы его цифр:")  
    val n = readLine()?.toIntOrNull()  
  
    if (n != null) {  
        println("Сумма цифр числа $n: ${sumOfDigits(n)}")  
    } else {  
        println("Некорректный ввод.")  
    }  
}
```

## 11. Анаграммы

```
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() ?: ""

    if (areAnagrams(str1, str2)) {
        println("Строки являются анаграммами.")
    } else {
        println("Строки не являются анаграммами.")
    }
}
```

## 12. Числовая последовательность

```
fun main() {
    println("Введите начальное число:")
    val start = readLine()?.toIntOrNull()
    println("Введите шаг:")
    val step = readLine()?.toIntOrNull()

    if (start != null && step != null) {
        println("Числовая последовательность:")
        for (i in 0..9) {
            println(start + i * step)
        }
    }
}
```

```
    } else {  
        println("Некорректный ввод.")  
    }  
}
```

### 13. Таблица квадратов

```
fun main() {  
    println("Таблица квадратов чисел от 1 до 20:")  
    for (i in 1..20) {  
        println("$i^2 = ${i * i}")  
    }  
}
```

### 14. Генерация случайных чисел

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

```
fun main() {  
    println("10 случайных чисел от 1 до 100:")  
    repeat(10) {  
        println(Random.nextInt(1, 101))  
    }  
}
```

### 15. Проверка палиндрома

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

```
}
```

```
fun main() {  
    println("Введите строку:")  
    val input = readLine() ?: ""  
  
    if (isPalindrome(input)) {  
        println("Строка является палиндромом.")  
    } else {  
        println("Строка не является палиндромом.")  
    }  
}
```

16. Сигма (сумма квадратов)

```
fun sumOfSquares(n: Int): Int {  
    return (1..n).sumOf { it * it }  
}
```

```
fun main() {  
    println("Введите число N:")  
    val n = readLine()?.toIntOrNull()  
  
    if (n != null && n > 0) {  
        println("Сумма квадратов от 1 до $n: ${sumOfSquares(n)}")  
    } else {  
        println("Некорректный ввод.")  
    }  
}
```



## 17. ВЫВОД СИМВОЛОВ

```
fun main() {  
    println("Введите строку:")  
    val input = readLine() ?: ""  
  
    println("Символы строки по одному:")  
    for (char in input) {  
        println(char)  
    }  
}
```

## 18. Задача на лестницу

```
fun main() {  
    println("Введите высоту лестницы:")  
    val height = readLine()?.toIntOrNull()  
  
    if (height != null && height > 0) {  
        for (i in 1..height) {  
            println("#".repeat(i))  
        }  
    } else {  
        println("Некорректный ввод.")  
    }  
}
```

## 19. Сортировка списка

```
fun simpleSort(arr: IntArray): IntArray {  
    for (i in arr.indices) {  
        for (j in i + 1 until arr.size) {  
            if (arr[i] > arr[j]) {  
                val temp = arr[i]  
                arr[i] = arr[j]  
                arr[j] = temp  
            }  
        }  
    }  
    return arr  
}
```

```
fun main() {  
    val numbers = intArrayOf(34, 12, 45, 23, 56, 78, 11)  
  
    println("Исходный массив: ${numbers.joinToString(", ")}")  
  
    val sortedNumbers = simpleSort(numbers)  
  
    println("Отсортированный массив: ${sortedNumbers.joinToString(", ")}")  
}
```

## 20. Простые числа в диапазоне

```
fun isPrime(n: Int): Boolean {  
    if (n <= 1) return false  
    for (i in 2..Math.sqrt(n.toDouble()).toInt()) {
```

```

        if (n % i == 0) return false
    }
    return true
}

fun main() {
    println("Введите начальное число диапазона:")
    val start = readLine()?.toIntOrNull()

    println("Введите конечное число диапазона:")
    val end = readLine()?.toIntOrNull()

    if (start != null && end != null && start <= end) {
        println("Простые числа в диапазоне от $start до $end:")
        for (i in start..end) {
            if (isPrime(i)) {
                print("$i ")
            }
        }
        println()
    } else {
        println("Некорректный ввод.")
    }
}

```

## 21. Вывод даты

```

import java.time.LocalDate
import java.time.Month

```

```

fun main() {
    println("Введите год:")
    val year = readLine()?.toIntOrNull()
    println("Введите месяц (1-12):")
    val month = readLine()?.toIntOrNull()

    if (year != null && month != null && month in 1..12) {
        val daysInMonth = LocalDate.of(year, Month.of(month), 1).lengthOfMonth()
        println("Даты в месяце $month/$year:")
        for (day in 1..daysInMonth) {
            println("$day/$month/$year")
        }
    } else {
        println("Некорректный ввод.")
    }
}

```

## 22. Угадай число

```

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()?.toIntOrNull()
    }
}

```

```

when {
    guess == null -> println("Введите корректное число.")
    guess < numberToGuess -> println("Слишком низко!")
    guess > numberToGuess -> println("Слишком высоко!")
    else -> println("Поздравляем! Вы угадали число $numberToGuess.")
}
}
}

```

## 23. Сложение и умножение

```

fun main() {
    println("Введите первую цифру (или 'стоп' для выхода):")
    val firstInput = readLine()

    if (firstInput == "стоп") return

    val firstNumber = firstInput?.toDoubleOrNull()

    if (firstNumber == null) {
        println("Некорректный ввод.")
        return
    }

    while (true) {
        println("Введите вторую цифру (или 'стоп' для выхода):")
        val secondInput = readLine()
    }
}

```

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

```
val secondNumber = secondInput?.toDoubleOrNull()
```

```
if (secondNumber == null) {  
    println("Некорректный ввод.")  
    continue  
}
```

```
println("Выберите операцию: 1 - сложение, 2 - умножение")
```

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

```
when (operation) {  
    1 -> println("Результат: ${firstNumber + secondNumber}")  
    2 -> println("Результат: ${firstNumber * secondNumber}")  
    else -> println("Некорректная операция.")  
}  
}
```

## 24. Транспонирование матрицы

```
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 main() {
    val matrix = arrayOf(
        intArrayOf(1, 2, 3),
        intArrayOf(4, 5, 6),
        intArrayOf(7, 8, 9)
    )

    val transposedMatrix = transposeMatrix(matrix)

    println("Исходная матрица:")
    matrix.forEach { println(it.joinToString(" ")) }

    println("Транспонированная матрица:")
    transposedMatrix.forEach { println(it.joinToString(" ")) }
}

```

## 25. Кубы чисел

```

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

```

## 26. Сумма четных и нечетных чисел

```

fun main() {
    println("Введите число N:")
    val n = readLine()?.toIntOrNull()

    if (n != null && n > 0) {
        var evenSum = 0
        var oddSum = 0

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

        println("Сумма четных чисел: $evenSum")
        println("Сумма нечетных чисел: $oddSum")
    } else {
        println("Некорректный ввод.")
    }
}

```

## 27. Печать числа "пирамида"

```

fun main() {
    println("Введите число N:")
    val n = readLine()?.toIntOrNull()

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

```



```

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

```

## 28. Определение порядка

```

fun main() {
    println("Введите количество чисел:")
    val count = readLine()?.toIntOrNull()

    if (count != null && count > 0) {
        val numbers = mutableListOf<Int>()

        for (i in 1..count) {
            println("Введите число $i:")
            val number = readLine()?.toIntOrNull()
            if (number != null) {
                numbers.add(number)
            } else {
                println("Некорректный ввод.")
                return
            }
        }

        numbers.sort()
    }
}

```

```

        println("Числа в порядке возрастания: ${numbers.joinToString(", ")}")
    } else {
        println("Некорректный ввод.")
    }
}

```

## 29. Сумма ряда

```

fun main() {
    println("Введите число 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 от 1 до $n: $sum")
    } else {
        println("Некорректный ввод.")
    }
}

```

## 30. Конвертация в двоичную систему

```

fun toBinary(n: Int): String {
    return Integer.toBinaryString(n)
}

```

```
fun main() {  
    println("Введите целое число:")  
    val number = readLine()?.toIntOrNull()  
  
    if (number != null) {  
        val binaryRepresentation = toBinary(number)  
        println("Двоичное представление числа $number: $binaryRepresentation")  
    } else {  
        println("Некорректный ввод.")  
    }  
}
```