

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

```
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 для суммы от 1 до N:")
```

```
    val nSum = readLine()!!.toInt()
```

```
    println("Сумма от 1 до $nSum:  $\{nSum * (nSum + 1) / 2\}$ ")
```

```
    // 4. Факториал числа
```

```
    println("Введите число для вычисления факториала:")
```

```
    val nFact = readLine()!!.toInt()
```

```
    var factorial = 1
```

```
    for (i in 1..nFact) {
```

```
        factorial *= i
```

```
    }
```

```
    println("Факториал $nFact: $factorial")
```

```
    // 5. Проверка числа на простоту
```

```
    println("Введите число для проверки на простоту:")
```

```
    val nPrime = readLine()!!.toInt()
```

```
var isPrime = true
for (i in 2 until nPrime) {
    if (nPrime % i == 0) {
        isPrime = false
        break
    }
}
println("$nPrime является ли простым: $isPrime")
```

// 6. Вывод таблицы умножения

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

// 7. Фибоначчи

```
println("Введите число N для генерации чисел Фибоначчи:")
val nFib = readLine()!!.toInt()
var a = 0
var b = 1
println("Числа Фибоначчи:")
for (i in 1..nFib) {
    print("$a ")
    val next = a + b
    a = b
    b = next
}
println()
```

// 8. Наибольший общий делитель (НОД)

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

```
println("Введите два числа для нахождения НОД:")  
val num1 = readLine()!!.toInt()  
val num2 = readLine()!!.toInt()  
println("НОД ($num1, $num2): ${gcd(num1, num2)}")
```

// 9. Обратный порядок

```
println("Введите строку для вывода в обратном порядке:")  
val inputString = readLine()!!  
println("Обратный порядок: ${inputString.reversed()}")
```

// 10. Сумма цифр числа

```
println("Введите число для нахождения суммы его цифр:")  
val inputNumber = readLine()!!.toInt()  
val sumDigits = inputNumber.toString().sumOf { it.toString().toInt() }  
println("Сумма цифр: $sumDigits")
```

// 11. Анаграммы

```
println("Введите две строки для проверки на анаграммы:")  
val str1 = readLine()!!  
val str2 = readLine()!!  
val isAnagram = str1.toCharArray().sorted() == str2.toCharArray().sorted()  
println("Являются ли строки анаграммами: $isAnagram")
```

// 12. Числовая последовательность

```
println("Введите начальное число и шаг:")  
val start = readLine()!!.toInt()  
val step = readLine()!!.toInt()  
for (i in start..start + 10 * step step step) {
```

```
        print("$i ")
    }
    println()
```

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

```
// 14. Генерация случайных чисел
println("10 случайных чисел от 1 до 100:")
repeat(10) {
    println(Random.nextInt(1, 101))
}
```

```
// 15. Проверка палиндрома
println("Введите строку для проверки на палиндром:")
val palindromeString = readLine()!!
val isPalindrome = palindromeString == palindromeString.reversed()
println("Является ли строка палиндромом: $isPalindrome")
```

```
// 16. Сигма (сумма квадратов)
println("Введите число N для нахождения суммы квадратов от 1 до N:")
val nSquares = readLine()!!.toInt()
val sumSquares = (1..nSquares).sumOf { it * it }
println("Сумма квадратов от 1 до $nSquares: $sumSquares")
```

```
// 17. Вывод символов
println("Введите строку для вывода символов по одному:")
val charString = readLine()!!
for (char in charString) {
```

```
println(char)
}
```

```
// 18. Задача на лестницу
```

```
println("Введите высоту лестницы:")
val height = readLine()!!.toInt()
for (i in 1..height) {
    println("#".repeat(i))
}
```

```
// 19. Сортировка списка
```

```
val numbers = mutableListOf(23, 45, 12, 67, 34, 89, 10, 99)
for (i in 0 until numbers.size) {
    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")
```

```
// 20. Простые числа в диапазоне
```

```
println("Введите диапазон (начало и конец):")
val startRange = readLine()!!.toInt()
val endRange = readLine()!!.toInt()
println("Простые числа в диапазоне $startRange до $endRange:")
for (i in startRange..endRange) {
    if (i > 1 && (2 until i).none { i % it == 0 }) {
        println(i)
    }
}
```

```
}
```

```
// 21. Вывод даты
```

```
println("Введите год и месяц:")
```

```
val year = readLine()!!.toInt()
```

```
val month = readLine()!!.toInt()
```

```
val daysInMonth = java.time.YearMonth.of(month, year).lengthOfMonth()
```

```
println("Даты в месяце $month/$year:")
```

```
for (day in 1..daysInMonth) {
```

```
    println("$year-$month-$day")
```

```
}
```

```
// 22. Угадай число
```

```
val randomNumber = Random.nextInt(1, 101)
```

```
var guessed = false
```

```
while (!guessed) {
```

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

```
    val guess = readLine()!!.toInt()
```

```
    when {
```

```
        guess < randomNumber -> println("Слишком маленькое число.")
```

```
        guess > randomNumber -> println("Слишком большое число.")
```

```
        else -> {
```

```
            println("Поздравляем! Вы угадали число.")
```

```
            guessed = true
```

```
        }
```

```
    }
```

```
}
```

```
// 23. Сложение и умножение
```

```
println("Введите две цифры для сложения/умножения:")
```

```
val firstNum = readLine()!!.toInt()
```

```
val secondNum = readLine()!!.toInt()
```

```

var operation = ""
while (operation != "стоп") {
    println("Введите операцию (+, *) или 'стоп' для выхода:")
    operation = readLine()!!
    when (operation) {
        "+" -> println("Результат: ${firstNum + secondNum}")
        "*" -> println("Результат: ${firstNum * secondNum}")
    }
}

```

// 24. Транспонирование матрицы

```

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

println("Транспонированная матрица:")
for (i in matrix.indices) {
    for (j in matrix[i].indices) {
        if (j == 0) print("[")
        print("${matrix[j][i]} ")
        if (j == matrix[i].lastIndex) println("]")
    }
}

```

// 25. Кубы чисел

```

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

```

// 26. Сумма четных и нечетных чисел

```
println("Введите число N для нахождения суммы четных и нечетных чисел:")
val nEvenOdd = readLine()!!.toInt()
val sumEven = (1..nEvenOdd).filter { it % 2 == 0 }.sum()
val sumOdd = (1..nEvenOdd).filter { it % 2 != 0 }.sum()
println("Сумма четных: $sumEven, Сумма нечетных: $sumOdd")
```

// 27. Печать числа "пирамида"

```
println("Введите число N для печати пирамиды:")
val nPyramid = readLine()!!.toInt()
for (i in 1..nPyramid) {
    println(" ".repeat(nPyramid - i) + (1..i).joinToString(" "))
}
```

// 28. Определение порядка

```
println("Введите количество чисел для сортировки:")
val count = readLine()!!.toInt()
val numbersToSort = mutableListOf<Int>()
repeat(count) {
    println("Введите число:")
    numbersToSort.add(readLine()!!.toInt())
}
println("Отсортированные числа: ${numbersToSort.sorted()}")
```

// 29. Сумма ряда

```
println("Введите число N для нахождения суммы ряда  $1 + 1/2 + \dots + 1/N$ :")
val nSeries = readLine()!!.toInt()
val seriesSum = (1..nSeries).sumOf { 1.0 / it }
println("Сумма ряда: $seriesSum")
```

// 30. Конвертация в двоичную систему

```
println("Введите целое число для конвертации в двоичную систему:")
val numberToConvert = readLine()!!.toInt()
```



```
        println("Число $numberToConvert в двоичной системе:  
${Integer.toBinaryString(numberToConvert)}")  
    }
```