

Цикл

Задание 1

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

Ответ:

1
2
3
4
5
6
7
8
9
10

Process finished with exit code 0

Задание 2

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

Ответ:

2

4

6

8

10

12

14

16

18

20

Process finished with exit code 0

Задание 3

```
fun main() {  
    println("Введите число N:")  
    val n = readLine()?.toIntOrNull() // Читаем ввод пользователя и  
    конвертируем в Int  
  
    if (n != null && n > 0) { // Проверяем, что число корректное и  
        положительное  
        var sum = 0 // Переменная для хранения суммы  
  
        for (i in 1..n) { // Цикл от 1 до N  
            sum += i // Добавляем текущее число к сумме  
        }  
  
        println("Сумма чисел от 1 до $n = $sum") // Выводим результат  
    } else {  
        println("Пожалуйста, введите положительное целое число.") //  
        Сообщение об ошибке  
    }  
}
```

Ответ:

Введите число N:

22999

Сумма чисел от 1 до 22999 = 264488500

Process finished with exit code 0

Задание 4

```
fun main() {  
    println("Введите число для вычисления факториала:")  
    val input = readLine()  
  
    if (input != null && input.toIntOrNull() != null) {  
        val number = input.toInt()  
  
        if (number < 0) {  
            println("Факториал отрицательного числа не определен.")  
        } else {  
            val factorial = calculateFactorial(number)  
            println("Факториал числа $number равен $factorial")  
        }  
    } else {  
        println("Пожалуйста, введите целое число.")  
    }  
}  
  
fun calculateFactorial(n: Int): Long {  
    var result: Long = 1  
    for (i in 1..n) {  
        result *= i  
    }  
    return result  
}
```

Ответ:

Введите число для вычисления факториала:

6

Факториал числа 6 равен 720

Process finished with exit code 0

Задание 5

```
fun main() {  
    println("Введите число:")  
    val input = readLine() ?: return  
    val number = input.toIntOrNull()  
  
    if (number == null || number < 1) {  
        println("Пожалуйста, введите положительное целое число.")  
        return  
    }  
  
    if (isPrime(number)) {  
        println("$number является простым числом.")  
    } else {  
        println("$number не является простым числом.")  
    }  
}  
  
fun isPrime(num: Int): Boolean {  
    if (num < 2) return false // Простые числа начинаются с 2.  
    for (i in 2..Math.sqrt(num.toDouble()).toInt()) {  
        if (num % i == 0) {  
            return false  
        }  
    }  
    return true  
}
```

Ответ:

Введите число:

10

10 не является простым числом.

Process finished with exit code 0

Задание 6

```
fun main() {  
    // Вывод заголовка таблицы  
    println("Таблица умножения от 1 до 10:")  
  
    // Внешний цикл для множителя  
    for (i in 1..10) {  
        // Внутренний цикл для умножаемого  
        for (j in 1..10) {  
            // Вывод результата умножения  
            print("${i * j}\t") // Используем табуляцию для лучшего  
форматирования  
        }  
        // Переход на новую строку после каждой строки результатов  
        println()  
    }  
}
```

Ответ:

Таблица умножения от 1 до 10:

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

Process finished with exit code 0

Задание 7

```
fun main() {  
    println("Введите количество чисел Фибоначчи для генерации:")  
    val n = readLine()?.toIntOrNull()  
  
    if (n == null || n <= 0) {  
        println("Пожалуйста, введите положительное целое число.")  
        return  
    }  
  
    val fibonacciNumbers = mutableListOf<Int>()  
  
    var first = 0  
    var second = 1  
  
    for (i in 0 until n) {  
        fibonacciNumbers.add(first)  
        val next = first + second  
        first = second  
        second = next  
    }  
  
    println("Первые $n чисел Фибоначчи: $fibonacciNumbers")  
}
```

Ответ:

Введите количество чисел Фибоначчи для генерации:

10

Первые 10 чисел Фибоначчи: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

Process finished with exit code 0

Задание 8

```
fun gcd(a: Int, b: Int): Int {  
    var num1 = a  
    var num2 = b  
  
    while (num2 != 0) {  
        val temp = num2  
        num2 = num1 % num2  
        num1 = temp  
    }  
  
    return num1  
}  
  
fun main() {  
    println("Введите первое число:")  
    val firstNumber = readLine()?.toIntOrNull()  
  
    println("Введите второе число:")  
    val secondNumber = readLine()?.toIntOrNull()  
  
    if (firstNumber != null && secondNumber != null) {  
        val result = gcd(firstNumber, secondNumber)  
        println("Наибольший общий делитель (НОД) чисел $firstNumber и  
$secondNumber = $result")  
    } else {  
        println("Вы ввели некорректные данные.")  
    }  
}
```

Ответ:

Введите первое число:

6

Введите второе число:

7

Наибольший общий делитель (НОД) чисел 6 и 7 = 1

Process finished with exit code 0

Задание 9

```
fun main() {  
    println("Введите строку:")  
    val input = readLine() ?: ""  
  
    // Создаем переменную для хранения обратной строки  
    var reversedString = ""  
  
    // Цикл для обхода строки с конца до начала  
    for (i in input.length - 1 downTo 0) {  
        reversedString += input[i]  
    }  
  
    // Выводим обратную строку  
    println("Обратная строка: $reversedString")  
}
```

Ответ:

Введите строку:

20

Обратная строка: 02

Process finished with exit code 0

Задание 10

```
fun main() {  
    println("Введите число:")  
    val input = readLine() ?: return // Читаем ввод пользователя  
  
    var sum = 0 // Переменная для хранения суммы цифр  
  
    for (char in input) { // Итерируемся по каждому символу строки  
        if (char.isDigit()) { // Проверяем, является ли символ цифрой  
            sum += char.toString().toInt() // Преобразуем символ в число и  
            добавляем к сумме  
        }  
    }  
  
    println("Сумма цифр числа: $sum") // Выводим результат  
}
```

Ответ:

Введите число:

10

Сумма цифр числа: 1

Process finished with exit code 0

Задание 11

```
fun areAnagrams(str1: String, str2: String): Boolean {
    // Удаляем пробелы и переводим строки в нижний регистр
    val cleanStr1 = str1.replace(" ", "").lowercase()
    val cleanStr2 = str2.replace(" ", "").lowercase()

    // Если длины строк не равны, они не могут быть анаграммами
    if (cleanStr1.length != cleanStr2.length) {
        return false
    }

    // Создаем массивы для подсчета символов
    val charCount = IntArray(26) // Предполагаем, что только английские буквы

    // Подсчитываем количество символов в первой строке
    for (char in cleanStr1) {
        charCount[char - 'a']++ // Увеличиваем счетчик для соответствующей
буквы
    }

    // Вычитаем количество символов во второй строке
    for (char in cleanStr2) {
        charCount[char - 'a']-- // Уменьшаем счетчик для соответствующей
буквы
    }

    // Если все счетчики равны нулю, строки анаграммы
    for (count in charCount) {
        if (count != 0) {
            return false
        }
    }

    return true
}

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.")
    }
}
```

Ответ:

Введите первую строку:

9

Введите вторую строку:

10

Строки не являются анаграммами.

Process finished with exit code 0

Задание 12

```
fun main() {  
    // Чтение начального числа  
    print("Введите начальное число: ")  
    val start = readLine()?.toIntOrNull()  
    if (start == null) {  
        println("Ошибка: введено невалидное число.")  
        return  
    }  
  
    // Чтение шага  
    print("Введите шаг: ")  
    val step = readLine()?.toIntOrNull()  
    if (step == null) {  
        println("Ошибка: введен невалидный шаг.")  
        return  
    }  
  
    // Чтение количества элементов в последовательности  
    print("Введите количество элементов в последовательности: ")  
    val count = readLine()?.toIntOrNull()  
    if (count == null) {  
        println("Ошибка: введено невалидное количество элементов.")  
        return  
    }  
  
    // Генерация и вывод последовательности  
    println("Сгенерированная последовательность:")  
    var current = start  
    for (i in 0 until count) {  
        println(current)  
        current += step  
    }  
}
```

Ответ:

Введите начальное число: 7

Введите шаг: 1

Введите количество элементов в последовательности: 3

Сгенерированная последовательность:

7

8

9

Process finished with exit code 0

Задание 13

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

Ответ:

$1^2 = 1$

$2^2 = 4$

$3^2 = 9$

$4^2 = 16$

$5^2 = 25$

$6^2 = 36$

$7^2 = 49$

$8^2 = 64$

$9^2 = 81$

$10^2 = 100$

$11^2 = 121$

$12^2 = 144$

$13^2 = 169$

$14^2 = 196$

$15^2 = 225$

$16^2 = 256$

$17^2 = 289$

$18^2 = 324$

$19^2 = 361$

$20^2 = 400$

Process finished with exit code 0

Задание 14

```
fun main() {  
    val random = java.util.Random()  
  
    for (i in 1..10) {  
        val randomNumber = random.nextInt(100) + 1 // Генерация числа от 1 до  
100  
        println(randomNumber)  
    }  
}
```

Ответ:

77

23

97

69

85

88

14

26

11

30

Process finished with exit code 0

Задание 15

```
fun main() {  
    while (true) {  
        println("Введите строку (или 'exit' для выхода):")  
        val input = readLine() ?: break  
  
        if (input.lowercase() == "exit") {  
            println("Выход из программы.")  
            break  
        }  
  
        // Удаляем пробелы и приводим к нижнему регистру  
        val sanitizedInput = input.replace("\\s+".toRegex(), "").lowercase()  
  
        // Проверка на палиндром  
        if (isPalindrome(sanitizedInput)) {  
            println("Строка является палиндромом.")  
        } else {  
            println("Строка не является палиндромом.")  
        }  
    }  
}  
  
fun isPalindrome(str: String): Boolean {  
    val length = str.length  
    for (i in 0 until length / 2) {  
        if (str[i] != str[length - 1 - i]) {  
            return false  
        }  
    }  
    return true  
}
```

Ответ:

Введите строку (или 'exit' для выхода):

1

Строка является палиндромом.

Введите строку (или 'exit' для выхода):

2

Строка является палиндромом.

Введите строку (или 'exit' для выхода):

4

Строка является палиндромом.

Введите строку (или 'exit' для выхода):

7

Строка является палиндромом.

Введите строку (или 'exit' для выхода):

8

Строка является палиндромом.

Введите строку (или 'exit' для выхода):

exit

Выход из программы.

Process finished with exit code 0

Задание 16

```
fun sumOfSquares(n: Int): Int {  
    var sum = 0  
    for (i in 1..n) {  
        sum += i * i  
    }  
    return sum  
}  
  
fun main() {  
    val N = 10 // Замените 10 на любое положительное число  
    val result = sumOfSquares(N)  
    println("Сумма квадратов чисел от 1 до $N: $result")  
}
```

Ответ:

Сумма квадратов чисел от 1 до 10: 385

Process finished with exit code 0

Задание 17

```
fun main() {  
    val str = "Привет, мир!" // Ваша строка  
  
    for (char in str) {  
        println(char) // Выводим каждый символ на новой строке  
    }  
}
```

Ответ:

П

р

и

в

е

т

,

м

и

р

!

Process finished with exit code 0

Задание 18

```
fun main() {  
    while (true) {  
        println("Введите высоту лестницы (отрицательное число для выхода):")  
        val input = readLine()?.toIntOrNull()  
  
        // Проверяем, что введенное значение не является null  
        if (input == null) {  
            println("Ошибка: введите корректное число.")  
            continue  
        }  
  
        // Проверяем условие выхода  
        if (input < 0) {  
            println("Выход из программы.")  
            break  
        }  
  
        // Вывод лестницы  
        for (i in 1..input) {  
            println("#".repeat(i)) // Формируем строку из символов "#"   
        }  
    }  
}
```

Ответ:

Введите высоту лестницы (отрицательное число для выхода):

8

#

##

###

####

#####

#####

#####

#####

Введите высоту лестницы (отрицательное число для выхода):

-8

Выход из программы.

Process finished with exit code 0

Задание 19

```
fun main() {  
    // Исходный массив двухзначных чисел  
    val numbers = arrayOf(45, 23, 89, 12, 34, 56, 78, 11, 90, 67)  
  
    // Вызываем функцию сортировки  
    bubbleSort(numbers)  
  
    // Выводим отсортированный массив  
    println("Отсортированный массив: ${numbers.joinToString(", ")}")  
}  
  
// Функция сортировки пузырьком  
fun bubbleSort(arr: Array<Int>) {  
    val n = arr.size  
    // Внешний цикл для прохода по всем элементам массива  
    for (i in 0 until n - 1) {  
        // Внутренний цикл для сравнения и обмена элементов  
        for (j in 0 until n - 1 - i) {  
            // Если текущий элемент больше следующего, меняем их местами  
            if (arr[j] > arr[j + 1]) {  
                // Меняем местами arr[j] и arr[j + 1]  
                val temp = arr[j]  
                arr[j] = arr[j + 1]  
                arr[j + 1] = temp  
            }  
        }  
    }  
}
```

Ответ:

Отсортированный массив: 11, 12, 23, 34, 45, 56, 67, 78, 89, 90

Process finished with exit code 0

Задание 20

```
fun main() {  
    // Ввод диапазона от и до  
    println("Введите начальное значение диапазона:")  
    val start = readLine()!!.toInt()  
  
    println("Введите конечное значение диапазона:")  
    val end = readLine()!!.toInt()  
  
    println("Простые числа от $start до $end:")  
  
    // Проходим по всем числам в заданном диапазоне  
    for (num in start..end) {  
        if (isPrime (num)) {  
            println(num)  
        }  
    }  
}
```

Ответ:

Введите начальное значение диапазона:

8

Введите конечное значение диапазона:

9

Простые числа от 8 до 9:

Process finished with exit code 0

Задание 21

```
import java.time.LocalDate
import java.time.Month
import java.util.Scanner

fun main() {
    val scanner = Scanner(System.`in`)

    // Запрашиваем у пользователя год
    println("Введите год (например, 2023):")
    val year = scanner.nextInt()

    // Запрашиваем у пользователя месяц
    println("Введите месяц (1-12):")
    val monthValue = scanner.nextInt()

    // Получаем месяц по значению
    val month = Month.of(monthValue)

    // Получаем первый и последний день месяца
    val firstDayOfMonth = LocalDate.of(year, month, 1)
    val lastDayOfMonth =
firstDayOfMonth.withDayOfMonth(firstDayOfMonth.lengthOfMonth())

    // Выводим все даты в месяце
    println("Даты в месяце ${month.name} $year:")
    var date = firstDayOfMonth
    while (date <= lastDayOfMonth) {
        println(date)
        date = date.plusDays(1)
    }
}
```

Ответ:

Введите год (например, 2023):

2025

Введите месяц (1-12):

7

Даты в месяце JULY 2025:

2025-07-01

2025-07-02

2025-07-03

2025-07-04

2025-07-05

2025-07-06

2025-07-07

2025-07-08

2025-07-09

2025-07-10

2025-07-11

2025-07-12

2025-07-13

2025-07-14

2025-07-15

2025-07-16

2025-07-17

2025-07-18

2025-07-19

2025-07-20

2025-07-21

2025-07-22

2025-07-23

2025-07-24

2025-07-25

2025-07-26

2025-07-27

2025-07-28

2025-07-29

2025-07-30

2025-07-31

Process finished with exit code 0

Задание 22

```
import kotlin.random.Random

fun main() {
    val randomNumber = Random.nextInt(1, 101) // Генерируем случайное число
    от 1 до 100
    var guessedNumber: Int? = null // Переменная для хранения числа,
    введенного пользователем
    var attempts = 0 // Счетчик попыток

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

    while (guessedNumber != randomNumber) {
        println("Введите ваше число:")
        val input = readLine()

        // Проверяем, является ли ввод числом
        guessedNumber = input?.toIntOrNull()

        if (guessedNumber == null) {
            println("Пожалуйста, введите корректное число.")
            continue // Пропускаем оставшиеся действия, если ввод некорректен
        }

        attempts++ // Увеличиваем количество попыток

        when {
            guessedNumber < randomNumber -> println("Загаданное число
            больше.")
            guessedNumber > randomNumber -> println("Загаданное число
            меньше.")
            else -> println("Поздравляю! Вы угадали число $randomNumber за
            $attempts попыток.")
        }
    }
}
```

Ответ:

Угадайте число от 1 до 100!

Введите ваше число:

5

Загаданное число больше.

Введите ваше число:

6

Загаданное число больше.

Введите ваше число:

95

Загаданное число меньше.

Введите ваше число:

80

Загаданное число меньше.

Введите ваше число:

50

Загаданное число больше.

Введите ваше число:

70

Загаданное число меньше.

Введите ваше число:

60

Загаданное число больше.

Введите ваше число:

65

Загаданное число больше.

Введите ваше число:

66

Поздравляю! Вы угадали число 66 за 9 попыток.

Process finished with exit code 0

Задание 23

```
fun main() {  
    while (true) {  
        println("Введите первое число (или 'стоп' для выхода):")  
        val input1 = readLine()  
        if (input1 == "стоп") break  
  
        println("Введите второе число (или 'стоп' для выхода):")  
        val input2 = readLine()  
        if (input2 == "стоп") break  
  
        try {  
            val num1 = input1?.toDoubleOrNull()  
            val num2 = input2?.toDoubleOrNull()  
  
            if (num1 == null || num2 == null) {  
                println("Неверный ввод. Пожалуйста, введите числа.")  
                continue  
            }  
  
            println("1. Сложение")  
            println("2. Умножение")  
            println("Выберите операцию (1 или 2): ")  
            val operation = readLine()  
  
            when (operation) {  
                "1" -> println("Результат сложения: ${num1 + num2}")  
                "2" -> println("Результат умножения: ${num1 * num2}")  
                else -> println("Неверный выбор операции.")  
            }  
        } catch (e: Exception) {  
            println("Произошла ошибка: ${e.message}")  
        }  
    }  
    println("Программа завершена.")  
}
```

Ответ:

Введите первое число (или 'стоп' для выхода):

1

Введите второе число (или 'стоп' для выхода):

2

1. Сложение

2. Умножение

Выберите операцию (1 или 2):

2

Результат умножения: 2.0

Введите первое число (или 'стоп' для выхода):

стоп

Программа завершена.

Process finished with exit code 0

Задание 24

```
fun main() {  
    // Исходная матрица  
    val matrix = arrayOf(  
        arrayOf(1, 2, 3),  
        arrayOf(4, 5, 6),  
        arrayOf(7, 8, 9)  
    )  
  
    // Выводим исходную матрицу  
    println("Исходная матрица:")  
    printMatrix(matrix)  
  
    // Транспонирование матрицы  
    val transposedMatrix = transposeMatrix(matrix)  
  
    // Выводим транспонированную матрицу  
    println("Транспонированная матрица:")  
    printMatrix(transposedMatrix)  
}  
  
fun transposeMatrix(matrix: Array<Array<Int>>): Array<Array<Int>> {  
    val rows = matrix.size  
    val cols = matrix[0].size  
    val transposed = Array(cols) { Array(rows) { 0 } } // Создаем пустую  
    матрицу  
  
    for (i in 0 until rows) {  
        for (j in 0 until cols) {  
            transposed[j][i] = matrix[i][j] // Обмен строк и столбцов  
        }  
    }  
    return transposed  
}  
  
fun printMatrix(matrix: Array<Array<Int>>) {  
    for (row in matrix) {  
        println(row.joinToString(" "))  
    }  
}
```

Ответ:

Исходная матрица:

1 2 3

4 5 6

7 8 9

Транспонированная матрица:

1 4 7

2 5 8

3 6 9

Process finished with exit code 0

Задание 25

```
fun main() {  
    for (i in 1..10) {  
        val cube = i * i * i  
        println("Куб числа $i равен $cube")  
    }  
}
```

Ответ:

Куб числа 1 равен 1

Куб числа 2 равен 8

Куб числа 3 равен 27

Куб числа 4 равен 64

Куб числа 5 равен 125

Куб числа 6 равен 216

Куб числа 7 равен 343

Куб числа 8 равен 512

Куб числа 9 равен 729

Куб числа 10 равен 1000

Process finished with exit code 0

Задание 26

```
fun main() {  
    // Запрос числа у пользователя  
    println("Введите число:")  
    val input = readLine()  
  
    // Проверка, что введенное значение не является null и является числом  
    if (input != null) {  
        val number = input.toIntOrNull()  
        if (number != null && number > 0) {  
            var evenSum = 0  
            var oddSum = 0  
  
            // Цикл для подсчета сумм четных и нечетных чисел  
            for (i in 1..number) {  
                if (i % 2 == 0) {  
                    evenSum += i // Добавляем к четной сумме  
                } else {  
                    oddSum += i // Добавляем к нечетной сумме  
                }  
            }  
  
            // Вывод результатов  
            println("Сумма четных чисел от 1 до $number: $evenSum")  
            println("Сумма нечетных чисел от 1 до $number: $oddSum")  
        } else {  
            println("Пожалуйста, введите положительное целое число.")  
        }  
    } else {  
        println("Ошибка ввода!")  
    }  
}
```

Ответ:

Введите число:

55

Сумма четных чисел от 1 до 55: 756

Сумма нечетных чисел от 1 до 55: 784

Process finished with exit code 0

Задание 27

```
fun main() {  
    // Запрашиваем у пользователя ввод числа  
    println("Введите число:")  
    val n = readLine()?.toIntOrNull()  
  
    // Проверяем, что введенное число корректно  
    if (n == null || n <= 0) {  
        println("Пожалуйста, введите положительное целое число.")  
        return  
    }  
  
    // Печатаем пирамиду  
    for (i in 1..n) {  
        // Печатаем пробелы для выравнивания  
        repeat(n - i) {  
            print(" ")  
        }  
        // Печатаем числа  
        for (j in 1..i) {  
            print("$j ")  
        }  
        // Переходим на новую строку  
        println()  
    }  
}
```

Ответ:

Введите число:

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

Задание 28

```
fun main() {  
    println("Введите количество чисел:")  
    val n = readLine()?.toIntOrNull() ?: return println("Некорректный ввод")  
  
    val numbers = mutableListOf<Int>()  
  
    println("Введите $n чисел:")  
    repeat(n) {  
        val number = readLine()?.toIntOrNull()  
        if (number != null) {  
            numbers.add(number)  
        } else {  
            println("Некорректный ввод числа, попробуйте снова.")  
        }  
    }  
  
    // Сортируем числа  
    numbers.sort()  
  
    // Выводим отсортированные числа  
    println("Числа в порядке возрастания:")  
    for (number in numbers) {  
        println(number)  
    }  
}
```

Ответ:

Введите количество чисел:

5

Введите 5 чисел:

2

8

4

6

9

Числа в порядке возрастания:

2

4

6

8

9

Process finished with exit code 0

Задание 29

```
fun main() {  
    // Запрашиваем у пользователя значение N  
    println("Введите значение N:")  
    val N = readLine()!!.toInt()  
  
    // Инициализируем переменную для хранения суммы  
    var sum = 0.0  
  
    // Вычисляем сумму ряда  
    for (i in 1..N) {  
        sum += 1.0 / i  
    }  
  
    // Выводим результат  
    println("Сумма ряда до N = $N: $sum")  
}
```

Ответ:

Введите значение N:

7

Сумма ряда до N = 7: 2.5928571428571425

Process finished with exit code 0

Задание 30

```
fun main() {  
    println("Введите целое число:")  
    val number = readLine()?.toIntOrNull()  
  
    if (number != null) {  
        val binaryRepresentation = convertToBinary(number)  
        println("Двоичное представление числа $number:  
$binaryRepresentation")  
    } else {  
        println("Некорректный ввод. Пожалуйста, введите целое число.")  
    }  
}  
  
fun convertToBinary(number: Int): String {  
    if (number == 0) return "0"  
  
    var n = number  
    val binaryString = StringBuilder()  
  
    while (n > 0) {  
        val remainder = n % 2  
        binaryString.insert(0, remainder) // Вставляем остаток в начало  
строки  
        n /= 2 // Делим число на 2  
    }  
  
    return binaryString.toString()  
}
```

Ответ:

Введите целое число:

4

Двоичное представление числа 4: 100

Process finished with exit code 0