

```

1)
fun sumOfList(numbers: List<Int>): Int {
    return numbers.sum()
}
fun main() {
    println("Введите элементы списка через запятую:")
    val input = readLine() ?: ""
    val numbers = input.split(",").map { it.trim().toInt() }
    println("Сумма элементов: ${sumOfList(numbers)}")
}

```

Введите элементы списка через запятую:

5,2

Сумма элементов: 7

Process finished with exit code 0

2)

```

fun differenceMaxMin(numbers: List<Int>): Int {
    return numbers.maxOrNull()!! - numbers.minOrNull()!!
}
fun main() {
    println("Введите целые числа через запятую:")
    val input = readLine() ?: ""
    val numbers = input.split(",").map { it.trim().toInt() }
    println("Разность между максимальным и минимальным: ${differenceMaxMin(numbers)}")
}

```

Введите целые числа через запятую:

4,6,52

Разность между максимальным и минимальным: 48

Process finished with exit code 0

3)

```

fun mergeLists(list1: List<Int>, list2: List<Int>): List<Int> {
    return list1 + list2
}
fun main() {
    println("Введите первый список чисел через запятую:")
    val input1 = readLine() ?: ""
    val list1 = input1.split(",").map { it.trim().toInt() }

    println("Введите второй список чисел через запятую:")
    val input2 = readLine() ?: ""
    val list2 = input2.split(",").map { it.trim().toInt() }
    println("Объединенный список: ${mergeLists(list1, list2)}")
}

```

Введите первый список чисел через запятую:

1,7,9

Введите второй список чисел через запятую:

2,6,34

Объединенный список: [1, 7, 9, 2, 6, 34]

Process finished with exit code 0

4)

```
fun isProfitable(prob: Double, prize: Double, pay: Double): Boolean {  
    return prob * prize > pay  
}  
fun main() {  
    println("Введите вероятность (prob):")  
    val prob = readLine()!!.toDouble()  
    println("Введите приз (prize):")  
    val prize = readLine()!!.toDouble()  
    println("Введите плату (pay):")  
    val pay = readLine()!!.toDouble()  
    println("Прибыльно ли это? ${isProfitable(prob, prize, pay)}")  
}
```

Введите вероятность (prob):

55

Введите приз (prize):

4

Введите плату (pay):

2

Прибыльно ли это? true

Process finished with exit code 0

6)

```
fun isSumLessThan100(a: Int, b: Int): Boolean {  
    return a + b < 100  
}  
fun main() {  
    println("Введите первое число:")  
    val a = readLine()!!.toInt()  
    println("Введите второе число:")  
    val b = readLine()!!.toInt()  
    println("Сумма меньше 100? ${isSumLessThan100(a, b)}")  
}
```

```
Введите первое число:  
4  
Введите второе число:  
5  
Сумма меньше 100? true  
  
Process finished with exit code 0
```

7)

```
fun isDivisibleBy100(number: Int): Boolean {  
    return number % 100 == 0  
}  
fun main() {  
    println("Введите целое число:")  
    val number = readLine()!!.toInt()  
    println("Делится ли число на 100? ${isDivisibleBy100(number)}")  
}
```

```
Введите целое число:  
1  
Делится ли число на 100? false  
  
Process finished with exit code 0
```

8)

```
fun calculateFrames(minutes: Int, fps: Int): Int {  
    return minutes * 60 * fps  
}  
fun main() {  
    println("Введите количество минут:")  
    val minutes = readLine()!!.toInt()  
    println("Введите частоту кадров (FPS):")  
    val fps = readLine()!!.toInt()  
    println("Количество кадров: ${calculateFrames(minutes, fps)}")  
}
```

```
Введите количество минут:  
5  
Введите частоту кадров (FPS):  
144  
Количество кадров: 43200  
  
Process finished with exit code 0
```

9)

```
fun isKPowerK(n: Int, k: Int): Boolean {
    return Math.pow(k.toDouble(), k.toDouble()).toInt() == n
}
fun main() {
    println("Введите значение n:")
    val n = readLine()!!.toInt()
    println("Введите значение k:")
    val k = readLine()!!.toInt()
    println("Является ли k^k равным n? ${isKPowerK(n, k)}")
}
```

Введите значение n:

4

Введите значение k:

2

Является ли k^k равным n? true

Process finished with exit code 0

10)

```
fun repeatString(txt: String, n: Int): String {
    return if (n <= 0) "" else txt + repeatString(txt, n - 1)
}
fun main() {
    println("Введите строку для повторения:")
    val txt = readLine() ?: ""
    println("Введите количество повторений:")
    val n = readLine()!!.toInt()
    println("Результат: ${repeatString(txt, n)}")
}
```

1234567890987654321234567890987654321234567890-0987654321

Введите количество повторений:

52

Результат: 1234567890987654321234567890987654321234567890-09876543211234567890987654321234

Process finished with exit code 0

11)

```
fun evaluateExpression(expression: String): Double {
    val cleanExpression = expression.replace(" ", "")
    return when {
        cleanExpression.contains("+") -> {
            val parts = cleanExpression.split("+")
            parts[0].toDouble() + parts[1].toDouble()
        }
        cleanExpression.contains("-") -> {
            val parts = cleanExpression.split("-")
            parts[0].toDouble() - parts[1].toDouble()
        }
    }
}
```

```

        parts[0].toDouble() - parts[1].toDouble()
    }
    cleanExpression.contains("*") -> {
        val parts = cleanExpression.split("*")
        parts[0].toDouble() * parts[1].toDouble()
    }
    cleanExpression.contains("/") -> {
        val parts = cleanExpression.split("/")
        parts[0].toDouble() / parts[1].toDouble()
    }
    else -> throw IllegalArgumentException("Unsupported operation")
}
}
fun main() {
    println("Введите уравнение (например, 1+1):")
    val equation = readLine() ?: ""
    try {
        val result = evaluateExpression(equation)
        println("Результат уравнения: $result")
    } catch (e: Exception) {
        println("Ошибка: ${e.message}")
    }
}

```

Введите уравнение (например, 1+1):

1+1

Результат уравнения: 2.0

Process finished with exit code 0

12)

```

fun googleWord(number: Int): String {
    return "G" + "o".repeat(number) + "gle"
}
fun main() {
    println("Введите количество букв 'o' в слове Google:")
    val numberO = readLine()?.toIntOrNull() ?: 0
    println("Слово: ${googleWord(numberO)}")
}

```

Введите количество букв 'o' в слове Google:

7

Слово: Goooooogle

Process finished with exit code 0

13)

```

fun greet() {
    println("Привет, мир!")
}

```

```
}  
fun main() {  
    greet()  
}
```

Привет, мир!

Process finished with exit code 0

14)

```
fun sumTwoNumbers(a: Int, b: Int): Int {  
    return a + b  
}  
fun main() {  
    println("Введите первое число:")  
    val firstNumber = readLine()?.toIntOrNull() ?: 0  
    println("Введите второе число:")  
    val secondNumber = readLine()?.toIntOrNull() ?: 0  
    println("Сумма: ${sumTwoNumbers(firstNumber, secondNumber)}")  
}
```

Введите первое число:

5

Введите второе число:

2

Сумма: 7

Process finished with exit code 0

15)

```
fun maxOfTwoNumbers(a: Int, b: Int): Int {  
    return if (a > b) a else b  
}  
fun main() {  
    println("Введите первое число для сравнения:")  
    val num1 = readLine()?.toIntOrNull() ?: 0  
    println("Введите второе число для сравнения:")  
    val num2 = readLine()?.toIntOrNull() ?: 0  
    println("Большее число: ${maxOfTwoNumbers(num1, num2)}")  
}
```

Введите первое число для сравнения:

8

Введите второе число для сравнения:

9

Большее число: 9

Process finished with exit code 0

16)

```
fun isEven(number: Int): Boolean {
    return number % 2 == 0
}
fun main() {
    println("Введите число для проверки четности:")
    val checkEvenNumber = readLine()?.toIntOrNull() ?: 0
    println("Число четное? ${isEven(checkEvenNumber)}")
}
```

Введите число для проверки четности:

6

Число четное? true

Process finished with exit code 0

17)

```
fun factorial(n: Int): Int {
    return if (n <= 1) 1 else n * factorial(n - 1)
}
fun main() {
    println("Введите число для вычисления факториала:")
    val factorialNumber = readLine()?.toIntOrNull() ?: 0
    println("Факториал числа $factorialNumber:
    ${factorial(factorialNumber)}")
}
```

Введите число для вычисления факториала:

7

Факториал числа 7: 5040

Process finished with exit code 0

18)

```
import kotlin.math.sqrt
fun isPrime(n: Int): Boolean {
    if (n < 2) return false
    for (i in 2..sqrt(n.toDouble()).toInt()) {
        if (n % i == 0) return false
    }
    return true
}
fun main() {
    println("Введите число для проверки на простоту:")
    val primeCheckNumber = readLine()?.toIntOrNull() ?: 0
    println("Число $primeCheckNumber простое? ${isPrime(primeCheckNumber)}")
}
```

Введите число для проверки на простоту:

5

Число 5 простое? true

Process finished with exit code 0

19)

```
fun sumOfArray(arr: IntArray): Int {  
    return arr.sum()  
}  
fun main() {  
    println("Введите числа через пробел для суммы в массиве:")  
    val arrayInput = readLine()?.split(" ")?.map { it.toInt() }?.toIntArray()  
    ?: intArrayOf()  
    println("Сумма чисел в массиве: ${sumOfArray(arrayInput)}")  
}
```

Введите числа через пробел для суммы в массиве:

5 6

Сумма чисел в массиве: 11

Process finished with exit code 0

20)

```
fun maxInArray(arr: IntArray): Int {  
    return arr.maxOrNull() ?: throw IllegalArgumentException("Array is  
empty")  
}  
fun main() {  
    println("Введите числа через пробел для нахождения максимального в  
массиве:")  
    val maxArrayInput = readLine()?.split(" ")?.map { it.toInt() }  
    }?.toIntArray() ?: intArrayOf()  
    println("Наибольшее число в массиве: ${maxInArray(maxArrayInput)}")  
}
```

Введите числа через пробел для нахождения максимального в массиве:

7 8 43

Наибольшее число в массиве: 43

Process finished with exit code 0

21)

```
fun sortArray(arr: IntArray): IntArray {
    return arr.sortedArray()
}
fun main() {
    println("Введите числа через пробел для сортировки массива:")
    val sortArrayInput = readLine()?.split(" ")?.map { it.toInt()
}?.toIntArray() ?: intArrayOf()
    println("Отсортированный массив:
${sortArray(sortArrayInput).joinToString(", ")}")
}
```

Введите числа через пробел для сортировки массива:

7 4 12 78 5

Отсортированный массив: 4, 5, 7, 12, 78

Process finished with exit code 0

22)

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

fun main() {
    println("Введите строку для проверки на палиндром:")
    val input = readLine()!!
    println("Это палиндром? ${isPalindrome(input)}")
}
```

C:\Users\yoigg\jdk\openjdk-24\bin\java.exe

Введите строку для проверки на палиндром:

течет

Это палиндром? true

Process finished with exit code 0

23)

```
fun countChars(str: String): Int = str.length

fun main() {
    println("Введите строку:")
    val input = readLine()!!
    println("Количество символов: ${countChars(input)}")
}
```

C:\Users\yoigg\jdk\openjdk-24\bin\java.exe

Введите строку:

Количество символов

Количество символов: 19

Process finished with exit code 0

24)

```
fun toUpperCase(str: String): String = str.uppercase()

fun main() {
    println("Введитестроку:")
    val input = readLine()!!
    println("В верхнем регистре: ${toUpperCase(input)}")
}
```

```
Введите строку:
апрель
В верхнем регистре: АПРЕЛЬ

Process finished with exit code 0
```

25)

```
fun concatStrings(str1: String, str2: String): String = str1 + str2

fun main() {
    println("Введитепервуюстроку:")
    val str1 = readLine()!!
    println("Введитевторуюстроку:")
    val str2 = readLine()!!
    println("Результатобъединения: ${concatStrings(str1, str2)}")
}
```

```
Введите первую строку:
Результат
Введите вторую строку:
объединения
Результат объединения: Результат объединения

Process finished with exit code 0
```

26)

```
fun lastElement(arr: Array<Int>): Int = arr.last()

fun main() {
    println("Введите числа через пробел:")
    val input = readLine()!!.split(" ").map{ it.toInt() }.toTypedArray()
    println("Последний элемент: ${lastElement(input)}")
}
```

```
Введите числа через пробел:
1 2 3 4 5
Последний элемент: 5

Process finished with exit code 0
```

27)

```
fun containsElement(arr: Array<Int>, element: Int): Boolean =
    arr.contains(element)

fun main() {
    println("Введите числа через пробел:")
    val arr = readLine()!!.split(" ").map { it.toInt() }.toTypedArray()
    println("Введите элемент для поиска:")
    val element = readLine()!!.toInt()
    println("Элемент присутствует? ${containsElement(arr, element)}")
}
```

Введите числа через пробел:

1 2 3 4 5

Введите элемент для поиска:

3

Элемент присутствует? true

Process finished with exit code 0

28)

```
fun createArray(n: Int): IntArray = IntArray(n) { it + 1 }

fun main() {
    println("Введите число N:")
    val n = readLine()!!.toInt()
    println("Массив: ${createArray(n).joinToString()}")
}
```

Введите число N:

10

Массив: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Process finished with exit code 0

29)

```
fun findMinMax(arr: Array<Int>): Pair<Int, Int> = arr.minOrNull()!! to
    arr.maxOrNull()!!
```

```
fun main() {
    println("Введите числа через пробел:")
    val arr = readLine()!!.split(" ").map { it.toInt() }.toTypedArray()
    val (min, max) = findMinMax(arr)
    println("Минимум: $min, Максимум: $max")
}
```

Введите числа через пробел:

1 2 3 4 5

Минимум: 1, Максимум: 5

Process finished with exit code 0

30)

```
fun sumToN(n: Int): Int = (1..n).sum()

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

Введите число N:

23654

Сумма: 279767685

Process finished with exit code 0

31)

```
fun celsiusToFahrenheit(c: Double): Double = c * 9/5 + 32

fun main() {
    println("Введите температуру в Цельсиях:")
    val celsius = readLine()!!.toDouble()
    println("Фаренгейты: ${celsiusToFahrenheit(celsius)}")
}
```

Введите температуру в Цельсиях:

12

Фаренгейты: 53.6

Process finished with exit code 0

32)

```
fun reverseString(str: String): String = str.reversed()

fun main() {
    println("Введите строку:")
    val input = readLine()!!
    println("Обратная строка: ${reverseString(input)}")
}
```

Введите строку:

слово

Обратная строка: оворс

Process finished with exit code 0

33)

```
fun getElementAtIndex(arr: Array<Int>, index: Int): Int = arr[index]

fun main() {
    println("Введите числа через пробел:")
    val arr = readLine()!!.split(" ").map { it.toInt() }.toTypedArray()
    println("Введите индекс:")
    val index = readLine()!!.toInt()
}
```

```
println("Элемент: ${getElementAtIndex(arr, index)}")
}
```

Введите числа через пробел:

132 223 486 465 783 464 786 676 823 678 623

Введите индекс:

4

Элемент: 783

Process finished with exit code 0

34)

```
fun removeSpaces(str: String): String = str.replace(" ", "")
```

```
fun main() {
println("Введитестрокуспробелами:")
valinput = readLine()!!
println("Без пробелов: ${removeSpaces(input)}")
}
```

Введите строку с пробелами:

с л о в о

Без пробелов: слово

Process finished with exit code 0

35)

```
fun sumNaturalNumbers(n: Int): Int = n * (n + 1) / 2
```

```
fun main() {
println("Введитечисло N:")
valn = readLine()!!.toInt()
println("Сумма: ${sumNaturalNumbers(n)}")
}
```

Введите число N:

50

Сумма: 1275

Process finished with exit code 0

36)

```
fun containsSubstring(str: String, sub: String): Boolean = str.contains(sub)
```

```
fun main() {
println("Введитестроку:")
valstr = readLine()!!
println("Введитеподстроку:")
valsub = readLine()!!
println("Содержитподстроку? ${containsSubstring(str, sub)}")
}
```

```
0x0000000000000000: 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
Введите строку:
Амурский государственный университет
Введите подстроку:
университет
Содержит подстроку? true

Process finished with exit code 0
```

37)

```
fun printMultiplicationTable(n: Int) {
    for (iin 1..10) {
        println("$n x $i = ${n * i}")
    }
}

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

Введите число:

5

$$\begin{array}{l} 5 \times 1 = 5 \\ 5 \times 2 = 10 \\ 5 \times 3 = 15 \\ 5 \times 4 = 20 \\ 5 \times 5 = 25 \\ 5 \times 6 = 30 \\ 5 \times 7 = 35 \\ 5 \times 8 = 40 \\ 5 \times 9 = 45 \\ 5 \times 10 = 50 \end{array}$$

```
Process finished with exit code 0
```

38)

```
fun stringLength(str: String): Int = str.length

fun main() {
    println("Введите строку:")
    val input = readLine()!!
    println("Длина строки: ${stringLength(input)}")
}
```

Введите строку:

строка

Длина строки: 6

```
Process finished with exit code 0
```

39)

```
fun reverseArray(arr: Array<Int>): Array<Int> = arr.reversedArray()

fun main() {
    println("Введите числа через пробел:")
    val arr = readLine()!!.split(" ").map{ it.toInt() }.toTypedArray()
    println("Перевернутый массив: ${reverseArray(arr).joinToString()}")
}
```

```
C:\Users\yoigg\.jdk\openjdk-24\bin\
Введите числа через пробел:
1 5 2 4 3
Перевернутый массив: 3, 4, 2, 5, 1

Process finished with exit code 0
```

40)

```
fun copyArray(arr: Array<Int>): Array<Int> = arr.copyOf()

fun main() {
    println("Введите числа через пробел:")
    val arr = readLine()!!.split(" ").map{ it.toInt() }.toTypedArray()
    println("Копия массива: ${copyArray(arr).joinToString()}")
}
```

```
Введите числа через пробел:
2 4 6 8 3 5 7
Копия массива: 2, 4, 6, 8, 3, 5, 7

Process finished with exit code 0
|
```

41)

```
fun countVowels(str: String): Int = str.count{ it.lowercaseChar() in setOf('a', 'y', 'o', 'и', 'э', 'ы', 'я', 'ю', 'e', 'ё') }

fun main() {
    println("Введите строку:")
    val input = readLine()!!
    println("Количество гласных: ${countVowels(input)}")
}
```

```
Введите строку:
кабачки
Количество гласных: 3

Process finished with exit code 0
```

42)

```
fun firstIndexOf(arr: Array<Int>, element: Int): Int = arr.indexOf(element)

fun main() {
    println("Введите числа через пробел:")
    val arr = readLine()!!.split(" ").map { it.toInt() }.toTypedArray()
    println("Введите элемент для поиска:")
    val element = readLine()!!.toInt()
    println("Индекс элемента: ${firstIndexOf(arr, element)}")
}
```

Введите числа через пробел:

3 5 7 4 2 56 89

Введите элемент для поиска:

7

Индекс элемента: 2

Process finished with exit code 0