Задание 1: Функция для суммы элементов списка

```
fun sumOfList(numbers: List<Int>): Int {
    return numbers.sum()
}
fun main() {
    val numbers = listOf(1, 2, 3, 4, 5)
    println("Сумма элементов: ${sumOfList(numbers)}") // Вывод: 15
}
```

Задание 2: Функция для разности максимального и минимального значения

```
fun differenceMaxMin(numbers: List<Int>): Int {
    return numbers.maxOrNull()!! - numbers.minOrNull()!!
}
fun main() {
    val numbers = listOf(10, 2, 8, 4, 6)
    println("Разность max и min: ${differenceMaxMin(numbers)}") // Вывод: 8
}
```

Задание 3: Функция для объединения двух списков

```
fun mergeLists(list1: List<Int>, list2: List<Int>): List<Int> {
    return list1 + list2
}
fun main() {
    val list1 = listOf(1, 2, 3)
    val list2 = listOf(4, 5, 6)
    println("Объединенный список: ${mergeLists(list1, list2)}") // Вывод: [1, 2, 3, 4, 5, 6]
}
```

Задание 4: Функция для проверки условия с prob, prize и pay

```
fun isProfitable(prob: Double, prize: Double, pay: Double): Boolean {
    return prob * prize > pay
}
```

```
fun main() {
    println("Прибыльно? ${isProfitable(0.8, 100.0, 50.0)}") // Вывод: true
}
```

Задание 5: Повторение (аналогично заданию 4)

```
fun isProfitable(prob: Double, prize: Double, pay: Double): Boolean {
    return prob * prize > pay
}
fun main() {
    println("Прибыльно? ${isProfitable(0.8, 100.0, 50.0)}") // Вывод: true
}
```

Задание 6: Функция для проверки суммы двух чисел

```
fun isSumLessThan100(a: Int, b: Int): Boolean {
    return (a + b) < 100
}
fun main() {
    println("Сумма меньше 100? ${isSumLessThan100(30, 40)}") // Вывод: true
}
```

Задание 7: Функция для проверки делимости на 100

```
fun isDivisibleBy100(number: Int): Boolean {
    return number % 100 == 0
}
fun main() {
    println("Делится на 100? ${isDivisibleBy100(200)}") // Вывод: true
}
```

Задание 8: Функция для подсчета кадров

```
fun calculateFrames(minutes: Int, fps: Int): Int {
    return minutes * 60 * fps
```

```
}
fun main() {
    println("Количество кадров: ${calculateFrames(2, 30)}") // Вывод: 3600
}
```

Задание 9: Функция для проверки k^k == n

```
fun isKPowerKEqualN(n: Int, k: Int): Boolean {
    return Math.pow(k.toDouble(), k.toDouble()).toInt() == n
}
fun main() {
    println("k^k равно n? ${isKPowerKEqualN(16, 4)}") // Вывод: true (4^4 = 256)
}
```

Задание 10: Рекурсивная функция для повторения строки

```
fun repeatString(txt: String, n: Int): String {
    return if (n <= 0) "" else txt + repeatString(txt, n - 1)
}
fun main() {
    println("Повторенная строка: ${repeatString("Hello", 3)}") // Вывод: HelloHelloHello
}
```

Задание 11. Создайте функцию, которая принимает уравнение (например, "1+1") и возвращает ответ.

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

```
        expression.contains("*") -> {
            val parts = expression.split("*")
            parts[0].toDouble() * parts[1].toDouble()
        }
        expression.contains("/") -> {
            val parts = expression.split("/")
            parts[0].toDouble() / parts[1].toDouble()
        }
        else -> throw IllegalArgumentException("Unsupported operation")
    }
}
fun main() {
    val equation = "1+1"
    println("Результат уравнения '$equation': ${evaluateExpression(equation)}")
}
```

Задание 12. Напишите функцию, которая принимает число number, и возвращает слово Google с количеством букв o, равным number.

```
fun generateGoogle(number: Int): String {
    val oCount = number.coerceIn(0, 2)
    return "G${"o".repeat(oCount)}gle"
}
fun main() {
    val number = 3 // Пример числа
    val result = generateGoogle(number)
    println(result) // Вывод: Goooogle (если number = 3)
}
```

13. Приветствие

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

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

14. Сумма двух чисел

```
fun sum(a: Int, b: Int): Int {
    return a + b
}
fun main() {
    val result = sum(3, 5)
    println("Сумма: $result")
}
```

15. Сравнение чисел

```
fun maxOfTwo(a: Int, b: Int): Int {
    return if (a > b) a else b
}
fun main() {
    val max = maxOfTwo(10, 20)
    println("Большее число: $max")
}
```

16. Определение четности

```
fun isEven(number: Int): Boolean {
    return number % 2 == 0
}
fun main() {
    val evenCheck = isEven(4)
    println("Четное число? $evenCheck")
}
```

17. Факториал числа

```
fun factorial(n: Int): Int {
    if (n < 0) throw IllegalArgumentException("Факториал для отрицательных чисел не определен.")
    return if (n == 0) 1 else n * factorial(n - 1)
}
fun main() {
    val fact = factorial(5)
    println("Факториал: $fact")
}
```

18. Проверка на простоту

```
fun isPrime(number: Int): Boolean {
    if (number <= 1) return false
    for (i in 2 until number) {
        if (number % i == 0) return false
    }
    return true
}
fun main() {
    val primeCheck = isPrime(7) // Пример вызова функции
    println("Простое число? $primeCheck")
}
```

19. Сумма чисел в массиве

```
fun sumArray(numbers: IntArray): Int {
    return numbers.sum()
}
fun main() {
    val total = sumArray(intArrayOf(1, 2, 3, 4))
    println("Сумма массива: $total")
}
```

20. Наибольшее число в массиве

```
fun maxInArray(numbers: IntArray): Int? {
    return numbers.maxOrNull()
}
fun main() {
    val maxNumber = maxInArray(intArrayOf(1, 5, 3, 9))
    println("Наибольшее число в массиве: $maxNumber")
}
```

21. Сортировка массива

```
fun sortArray(numbers: IntArray): IntArray {
    return numbers.sortedArray()
}
fun main() {
    val sortedArray = sortArray(intArrayOf(5, 3, 8, 1)) // Пример вызова функции
    println("Отсортированный массив: ${sortedArray.joinToString(", ")}")
}
```

22. Проверка палиндрома

```
fun isPalindrome(input: String): Boolean {
    return input == input.reversed()
}
fun main() {
    val palindromeCheck = isPalindrome("level")
    println("Палиндром? $palindromeCheck")
}
```

23. Количество символов

```
fun countCharacters(input: String): Int {
    return input.length
}
fun main() {
    val charCount = countCharacters("Hello")
```

```
    println("Количество символов: $charCount")
}
```

24. Конвертация в верхний регистр

```
fun toUpperCase(input: String): String {
    return input.uppercase()
}
fun main() {
    val upperCaseString = toUpperCase("hello")
    println("В верхнем регистре: $upperCaseString")
}
```

25. Объединение строк

```
fun concatenateStrings(str1: String, str2: String): String {
    return str1 + str2
}
fun main() {
    val result = concatenateStrings("Hello, ", "world!") // Пример вызова функции
    println(result)
}
```

26. Возвращение последнего элемента массива

```
fun lastElement(array: IntArray): Int? {
    return array.lastOrNull()
}
fun main() {
    val last = lastElement(intArrayOf(1, 2, 3, 4, 5))
    println("Последний элемент: $last")
}
```

27. Проверка наличия элемента

```
fun containsElement(array: IntArray, element: Int): Boolean {
    return element in array
}
fun main() {
    val exists = containsElement(intArrayOf(1, 2, 3, 4), 3)
    println("Элемент присутствует? $exists")
}
```

28. Создание массива от 1 до N

```
fun createArrayFrom1ToN(n: Int): IntArray {
    return IntArray(n) { it + 1 }
}
fun main() {
    val array = createArrayFrom1ToN(5)
    println("Массив от 1 до N: ${array.joinToString(", ")}")
}
```

29. Максимум и минимум

```
fun minMax(array: IntArray): Pair<Int?, Int?> {
    return Pair(array.minOrNull(), array.maxOrNull())
}
fun main() {
    val (min, max) = minMax(intArrayOf(3, 5, 1, 8, 2))
    println("Минимум: $min, Максимум: $max")
}
```

30. Сумма чисел от 1 до N

```
fun sumFrom1ToN(n: Int): Int {
    return (1..n).sum()
}
fun main() {
    val sum = sumFrom1ToN(5) // Пример вызова функции
```

```
    println("Сумма от 1 до N: $sum") // Вывод: Сумма от 1 до N: 15
}
```

31. Преобразование Celsius в Fahrenheit

```
fun celsiusToFahrenheit(celsius: Double): Double {
    return celsius * 9 / 5 + 32
}
fun main() {
    val fahrenheit = celsiusToFahrenheit(25.0)
    println("Температура в Фаренгейтах: $fahrenheit")
}
```

32. Обратный порядок строки

```
fun reverseString(input: String): String {
    return input.reversed()
}
fun main() {
    val reversed = reverseString("Hello")
    println("Обратный порядок строки: $reversed")
}
```

33. Поиск элемента по индексу

```
fun findElementByIndex(array: IntArray, index: Int): Int? {
    return if (index in array.indices) array[index] else null
}
fun main() {
    val element = findElementByIndex(intArrayOf(10, 20, 30), 1)
    println("Элемент по индексу: $element")
}
```

34. Удаление пробелов из строк

```
fun removeSpaces(input: String): String {
```

```
    return input.replace(" ", "")
}
fun main() {
    val noSpaces = removeSpaces("Hello World")
    println("Строка без пробелов: $noSpaces")
}
```

35. Сумма первых N натуральных чисел

```
fun sumFirstNNaturalNumbers(n: Int): Int {
    return (1..n).sum()
}
fun main() {
    val sum = sumFirstNNaturalNumbers(5)
    println("Сумма первых N натуральных чисел: $sum")
}
```

36. Проверка строки на наличие подстроки

```
fun containsSubstring(string: String, substring: String): Boolean {
    return substring in string
}
fun main() {
    val exists = containsSubstring("Hello world", "world")
    println("Подстрока присутствует? $exists")
}
```

37. Печать таблицы умножения

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

```
    printMultiplicationTable(5)
}
```

38. Нахождение длины строки

```
fun stringLength(input: String): Int {
    return input.length
}
fun main() {
    val length = stringLength("Hello")
    println("Длина строки: $length")
}
```

39. Переворот массива

```
fun reverseArray(array: IntArray): IntArray {
    return array.reversedArray()
}
fun main() {
    val reversedArray = reverseArray(intArrayOf(1, 2, 3, 4))
    println("Перевернутый массив: ${reversedArray.joinToString(", ")}")
}
```

40. Копирование массива

```
fun copyArray(array: IntArray): IntArray {
    return array.copyOf()
}
fun main() {
    val original = intArrayOf(1, 2, 3)
    val copied = copyArray(original)
    println("Скопированный массив: ${copied.joinToString(", ")}")
}
```

41. Количество гласных в строке

```
fun countVowels(input: String): Int {
    return input.count { it.lowercaseChar() in "aeiou" }
}
fun main() {
    val vowelsCount = countVowels("Hello World")
    println("Количество гласных в строке: $vowelsCount")
}
```

42. Индекс первого вхождения

```
fun indexOfFirstOccurrence(array: IntArray, element: Int): Int {
    return array.indexOf(element)
}
fun main() {
    val index = indexOfFirstOccurrence(intArrayOf(1, 2, 3, 2), 2)
    println("Индекс первого вхождения элемента: $index")
}
```