

Код внизу, после заданий.

1. Напишите функцию, которая принимает список на вход, и возвращает сумму всех элементов этого списка.
2. Напишите функцию, которая получает на вход список целых чисел и возвращает разность самого большого и самого маленького из них.
3. Создайте функцию для объединения двух списков целых чисел.
4. Создайте функцию, которая принимает три аргумента `prob`, `prize`, `pay` и возвращает `True`, если `prob * prize > pay`, в противном случае возвращает `False`.
5. Создайте функцию, которая принимает три аргумента `prob`, `prize`, `pay` и возвращает `True`, если `prob * prize > pay`, в противном случае возвращает `False`.
6. Функция получает на вход два числа. Она должна вернуть `True`, если сумма этих чисел меньше 100 и `False` в противном случае.
7. Напишите функцию, которая принимает целое число и возвращает `True`, если оно делится на 100. В противном случае функция должна вернуть `False`.
8. Напишите функцию, которая принимает количество минут и частоту кадров (FPS) и возвращает, сколько за это время кадров показывает компьютер при этом FPS.
9. Напишите функцию, которая возвращает `True`, если $k^k == n$ для входных данных (n , k), и возвращает `False` в противном случае.
10. Создайте рекурсивную функцию, которая принимает два параметра и повторяет строку n количество раз. Первый параметр `txt` - это строка, которую нужно повторить, а второй параметр - количество повторений строки. Примеры `repetition("ab", 3) → "ababab"` `repetition("kiwi", 1) → "kiwi"` `repetition("cherry", 2) → "cherrycherry"`
11. Создайте функцию, которая принимает уравнение (например, `"1+1"`) и возвращает ответ. Примеры `equation("1+1") → 2` `equation("7*4-2") → 26` `equation("1+1+1+1+1") → 5`
12. Напишите функцию, которая принимает число `number`, и возвращает слово Google с количеством букв `o`, равным `number`. Примеры `google(5) => "Goooooogle"` `google(0) => "Ggle"` `google(2) => "Google"`
13. Приветствие: Напишите функцию, которая выводит "Привет, мир!" на экран.
14. Сумма двух чисел: Создайте функцию, которая принимает два числа и возвращает их сумму.
15. Сравнение чисел: Напишите функцию, которая принимает два числа и возвращает большее из них.
16. Определение четности: Создайте функцию, которая принимает одно число и возвращает `true`, если оно четное, и `false`, если нечетное.
17. Факториал числа: Реализуйте функцию, которая вычисляет факториал введенного числа.
18. Проверка на простоту: Напишите функцию, которая проверяет, является ли заданное число простым.

19. Сумма чисел в массиве: Создайте функцию, которая принимает массив целых чисел и возвращает их сумму.
20. Наибольшее число в массиве: Напишите функцию, которая находит и возвращает максимальное значение в массиве.
21. Сортировка массива: Создайте функцию, которая сортирует массив чисел в порядке возрастания.
22. Проверка палиндрома: Напишите функцию, которая проверяет, является ли строка палиндромом.
23. Количество символов: Создайте функцию, которая принимает строку и возвращает количество символов в ней.
24. Конвертация в верхний регистр: Напишите функцию, которая принимает строку и возвращает её в верхнем регистре.
25. Объединение строк: Создайте функцию, которая принимает две строки и возвращает их объединение.
26. Возвращение последнего элемента массива: Напишите функцию, которая возвращает последний элемент переданного массива.
27. Проверка наличия элемента: Создайте функцию, которая проверяет, присутствует ли элемент в массиве.
28. Создание массива от 1 до N: Напишите функцию, которая создает массив целых чисел от 1 до N.
29. Максимум и минимум: Создайте функцию, которая находит одновременно максимальное и минимальное значения в массиве.
30. Сумма чисел от 1 до N: Напишите функцию, которая рассчитывает сумму всех чисел от 1 до N.
31. Преобразование Celsius в Fahrenheit: Создайте функцию, которая конвертирует температуру из Цельсия в Фаренгейт.
32. Обратный порядок строки: Напишите функцию, которая принимает строку и возвращает её в обратном порядке.
33. Поиск элемента по индексу: Создайте функцию, которая находит элемент по индексу в массиве.
34. Удаление пробелов из строки: Напишите функцию, которая удаляет все пробелы из переданной строки.
35. Сумма первых N натуральных чисел: Создайте функцию, которая возвращает сумму первых N натуральных чисел.
36. Проверка строки на наличие подстроки: Напишите функцию, которая проверяет, содержится ли одна строка в другой.
37. Печать таблицы умножения: Создайте функцию, которая выводит таблицу умножения от 1 до 10 для заданного числа.
38. Нахождение длины строки: Напишите функцию, которая возвращает длину переданной ей строки.

39. Переворот массива: Создайте функцию, которая переворачивает массив целых чисел.

40. Копирование массива: Напишите функцию, которая копирует массив и возвращает новый массив.

41. Количество гласных в строке: Создайте функцию, которая считает количество гласных в строке.

42. Индекс первого вхождения: Напишите функцию, которая возвращает индекс первого вхождения элемента в массиве, или -1, если элемент не найден.

Код:

```
import kotlin.math.pow
```

```
fun main() {
    while (true) {
        println("\nВыберите номер задачи (1-42, или 0 для выхода):")
        val taskNumber = readLine()?.toIntOrNull() ?: 0

        if (taskNumber == 0) break

        when (taskNumber) {
            1 -> task1()
            2 -> task2()
            3 -> task3()
            4, 5 -> task4And5()
            6 -> task6()
            7 -> task7()
            8 -> task8()
            9 -> task9()
            10 -> task10()
            11 -> task11()
            12 -> task12()
            13 -> task13()
            14 -> task14()
            15 -> task15()
            16 -> task16()
            17 -> task17()
            18 -> task18()
            19 -> task19()
            20 -> task20()
            21 -> task21()
            22 -> task22()
            23 -> task23()
            24 -> task24()
            25 -> task25()
            26 -> task26()
            27 -> task27()
            28 -> task28()
            29 -> task29()
        }
    }
}
```

```

        30 -> task30()
        31 -> task31()
        32 -> task32()
        33 -> task33()
        34 -> task34()
        35 -> task35()
        36 -> task36()
        37 -> task37()
        38 -> task38()
        39 -> task39()
        40 -> task40()
        41 -> task41()
        42 -> task42()
        else -> println("Неверный номер задачи.")
    }
}
}

// Задача 1: Сумма элементов списка
fun task1() {
    print("Введите числа через запятую: ")
    val input = readLine()?.split(",")?.map { it.trim().toInt() } ?: emptyList()
    println("Сумма: ${sumList(input)}")
}

// Задача 2: Разность между максимальным и минимальным элементами
fun task2() {
    print("Введите числа через запятую: ")
    val input = readLine()?.split(",")?.map { it.trim().toInt() } ?: emptyList()
    if (input.isEmpty()) println("Список пуст")
    else println("Разность max и min: ${maxMinDifference(input)}")
}

// Задача 3: Объединение двух списков
fun task3() {
    print("Введите первый список чисел через запятую: ")
    val list1 = readLine()?.split(",")?.map { it.trim().toInt() } ?: emptyList()
    print("Введите второй список чисел через запятую: ")
    val list2 = readLine()?.split(",")?.map { it.trim().toInt() } ?: emptyList()
    println("Объединенный список: ${mergeLists(list1, list2)}")
}

// Задача 4 & 5: Проверка условия prob * prize > pay
fun task4And5() {
    print("Введите вероятность (prob): ")
    val prob = readLine()?.toDoubleOrNull() ?: 0.0
    print("Введите приз (prize): ")
    val prize = readLine()?.toDoubleOrNull() ?: 0.0
    print("Введите плату (pay): ")
    val pay = readLine()?.toDoubleOrNull() ?: 0.0

```

```

        println("Результат: ${checkCondition(prob, prize, pay)}")
    }

// Задача 6: Проверка суммы двух чисел на меньше 100
fun task6() {
    print("Введите первое число: ")
    val num1 = readLine()?.toIntOrNull() ?: 0
    print("Введите второе число: ")
    val num2 = readLine()?.toIntOrNull() ?: 0
    println("Сумма < 100: ${isSumLess100(num1, num2)}")
}

// Задача 7: Проверка на делимость на 100
fun task7() {
    print("Введите число: ")
    val num = readLine()?.toIntOrNull() ?: 0
    println("Делится на 100: ${isDivisibleBy100(num)}")
}

// Задача 8: Количество кадров за заданное время
fun task8() {
    print("Введите количество минут: ")
    val minutes = readLine()?.toIntOrNull() ?: 0
    print("Введите частоту кадров (FPS): ")
    val fps = readLine()?.toIntOrNull() ?: 0
    println("Количество кадров: ${calculateFrames(minutes, fps)}")
}

// Задача 9: Проверка условия k^k == n
fun task9() {
    print("Введите n: ")
    val n = readLine()?.toDoubleOrNull() ?: 0.0
    print("Введите k: ")
    val k = readLine()?.toDoubleOrNull() ?: 0.0
    println("k^k == n: ${checkPower(n, k)}")
}

// Задача 10: Рекурсивное повторение строки
fun task10() {
    print("Введите строку: ")
    val txt = readLine() ?: ""
    print("Введите количество повторений: ")
    val n = readLine()?.toIntOrNull() ?: 0
    println("Результат: ${repetition(txt, n)}")
}

// Задача 11: Вычисление уравнения (простая версия, только сложение)
fun task11() {
    print("Введите уравнение (например, 1+1+2): ")
    val equation = readLine() ?: ""
    println("Результат: ${equation(equation)}")
}

```

```
// Задача 12: Google с количеством букв 'o'
fun task12() {
    print("Введите количество букв 'o': ")
    val num = readLine()?.toIntOrNull() ?: 0
    println("Результат: ${google(num)}")
}
```

```
// Задача 13: Приветствие
fun task13() {
    greet()
}
```

```
// Задача 14: Сумма двух чисел
fun task14() {
    print("Введите первое число: ")
    val a = readLine()?.toIntOrNull() ?: 0
    print("Введите второе число: ")
    val b = readLine()?.toIntOrNull() ?: 0
    println("Сумма: ${a + b}")
}
```

```
// Задача 15: Больше из двух чисел
fun task15() {
    print("Введите первое число: ")
    val a = readLine()?.toIntOrNull() ?: 0
    print("Введите второе число: ")
    val b = readLine()?.toIntOrNull() ?: 0
    println("Максимум: ${maxOf(a, b)}")
}
```

```
// Задача 16: Определение четности
fun task16() {
    print("Введите число: ")
    val num = readLine()?.toIntOrNull() ?: 0
    println("Четное: ${isEven(num)}")
}
```

```
// Задача 17: Факториал
fun task17() {
    print("Введите число: ")
    val num = readLine()?.toIntOrNull() ?: 0
    println("Факториал: ${factorial(num)}")
}
```

```
// Задача 18: Проверка на простоту
fun task18() {
    print("Введите число: ")
    val num = readLine()?.toIntOrNull() ?: 0
    println("Простое: ${isPrime(num)}")
}
```

```
}
```

```
// Задача 19: Сумма чисел в массиве
```

```
fun task19() {  
    print("Введите числа через запятую: ")  
    val input = readLine()?.split(",")?.map { it.trim().toInt() } ?: emptyList()  
    println("Сумма: ${sumList(input)}")  
}
```

```
// Задача 20: Наибольшее число в массиве
```

```
fun task20() {  
    print("Введите числа через запятую: ")  
    val input = readLine()?.split(",")?.map { it.trim().toInt() } ?: emptyList()  
    println("Максимум: ${input.maxOrNull()}")  
}
```

```
// Задача 21: Сортировка массива
```

```
fun task21() {  
    print("Введите числа через запятую: ")  
    val input = readLine()?.split(",")?.map { it.trim().toInt() } ?: emptyList()  
    println("Отсортированный список: ${input.sorted()}")  
}
```

```
// Задача 22: Проверка палиндрома
```

```
fun task22() {  
    print("Введите строку: ")  
    val str = readLine() ?: ""  
    println("Палиндром: ${isPalindrome(str)}")  
}
```

```
// Задача 23: Количество символов
```

```
fun task23() {  
    print("Введите строку: ")  
    val str = readLine() ?: ""  
    println("Длина строки: ${charCount(str)}")  
}
```

```
// Задача 24: Конвертация в верхний регистр
```

```
fun task24() {  
    print("Введите строку: ")  
    val str = readLine() ?: ""  
    println("Верхний регистр: ${toUpperCase(str)}")  
}
```

```
// Задача 25: Объединение строк
```

```
fun task25() {  
    print("Введите первую строку: ")  
    val str1 = readLine() ?: ""  
    print("Введите вторую строку: ")  
    val str2 = readLine() ?: ""
```

```

    println("Объединенные строки: ${concatStrings(str1, str2)}")
}

// Задача 26: Последний элемент массива
fun task26() {
    print("Введите числа через запятую: ")
    val input = readLine()?.split(",")?.map { it.trim().toInt() } ?: emptyList()
    println("Последний элемент: ${input.lastOrNull()}")
}

// Задача 27: Проверка наличия элемента
fun task27() {
    print("Введите числа через запятую: ")
    val input = readLine()?.split(",")?.map { it.trim().toInt() } ?: emptyList()
    print("Введите элемент для поиска: ")
    val element = readLine()?.toIntOrNull() ?: 0
    println("Элемент найден: ${containsElement(input.toIntArray(), element)}")
}

// Задача 28: Создание массива от 1 до N
fun task28() {
    print("Введите N: ")
    val n = readLine()?.toIntOrNull() ?: 0
    println("Массив: ${createArray(n).contentToString()}")
}

// Задача 29: Максимум и минимум одновременно
fun task29() {
    print("Введите числа через запятую: ")
    val input = readLine()?.split(",")?.map { it.trim().toInt() } ?: emptyList()
    println("Минимум и максимум: ${minMax(input.toIntArray())}")
}

// Задача 30: Сумма чисел от 1 до N
fun task30() {
    print("Введите N: ")
    val n = readLine()?.toIntOrNull() ?: 0
    println("Сумма от 1 до N: ${sumToN(n)}")
}

// Задача 31: Преобразование Celsius в Fahrenheit
fun task31() {
    print("Введите температуру в Цельсиях: ")
    val celsius = readLine()?.toDoubleOrNull() ?: 0.0
    println("Температура в Фаренгейтах: ${celsiusToFahrenheit(celsius)}")
}

// Задача 32: Обратный порядок строки
fun task32() {
    print("Введите строку: ")
    val str = readLine() ?: ""
    println("Обратный порядок: ${reverseString(str)}")
}

```



```
}
```

```
// Задача 33: Поиск элемента по индексу
```

```
fun task33() {  
    print("Введите числа через запятую: ")  
    val input = readLine()?.split(",")?.map { it.trim().toInt() } ?: emptyList()  
    print("Введите индекс: ")  
    val index = readLine()?.toIntOrNull() ?: 0  
    println("Элемент по индексу: ${findElementByIndex(input.toIntArray(), index)}")  
}
```

```
// Задача 34: Удаление пробелов из строки
```

```
fun task34() {  
    print("Введите строку: ")  
    val str = readLine() ?: ""  
    println("Строка без пробелов: ${removeSpaces(str)}")  
}
```

```
// Задача 35: Сумма первых N натуральных чисел
```

```
fun task35() {  
    print("Введите N: ")  
    val n = readLine()?.toIntOrNull() ?: 0  
    println("Сумма первых N чисел: ${sumFirstN(n)}")  
}
```

```
// Задача 36: Проверка строки на наличие подстроки
```

```
fun task36() {  
    print("Введите строку: ")  
    val str = readLine() ?: ""  
    print("Введите подстроку: ")  
    val sub = readLine() ?: ""  
    println("Подстрока найдена: ${containsSubstring(str, sub)}")  
}
```

```
// Задача 37: Печать таблицы умножения
```

```
fun task37() {  
    print("Введите число: ")  
    val num = readLine()?.toIntOrNull() ?: 0  
    printMultiplicationTable(num)  
}
```

```
// Задача 38: Нахождение длины строки
```

```
fun task38() {  
    print("Введите строку: ")  
    val str = readLine() ?: ""  
    println("Длина строки: ${stringLength(str)}")  
}
```

```
// Задача 39: Переворот массива
```

```
fun task39() {
```

```

    print("Введите числа через запятую: ")
    val input = readLine()?.split(",")?.map { it.trim().toInt() } ?: emptyList()
    println("Перевернутый массив: ${reverseArray(input.toIntArray()).contentToString()}")
}

```

// Задача 40: Копирование массива

```

fun task40() {
    print("Введите числа через запятую: ")
    val input = readLine()?.split(",")?.map { it.trim().toInt() } ?: emptyList()
    println("Скопированный массив: ${copyArray(input.toIntArray()).contentToString()}")
}

```

// Задача 41: Количество гласных в строке

```

fun task41() {
    print("Введите строку: ")
    val str = readLine() ?: ""
    println("Количество гласных: ${countVowels(str)}")
}

```

// Задача 42: Индекс первого вхождения

```

fun task42() {
    print("Введите числа через запятую: ")
    val input = readLine()?.split(",")?.map { it.trim().toInt() } ?: emptyList()
    print("Введите элемент для поиска: ")
    val element = readLine()?.toIntOrNull() ?: 0
    println("Индекс первого вхождения: ${firstIndexOf(input.toIntArray(), element)}")
}

```

// Вспомогательные функции

```

fun repetition(txt: String, n: Int): String = if (n == 0) "" else txt + repetition(txt, n - 1)

```

```

fun maxOf(a: Int, b: Int) = if (a > b) a else b

```

```

fun isEven(num: Int) = num % 2 == 0

```

```

fun factorial(n: Int): Long = if (n == 0) 1 else n * factorial(n - 1)

```

```

fun isPrime(num: Int): Boolean {
    if (num <= 1) return false
    for (i in 2 until num) {
        if (num % i == 0) return false
    }
    return true
}

```

```

fun greet() = println("Привет, мир!")

```

```

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

```

```

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

```

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

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

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

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

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

```
fun reverseArray(array: IntArray): IntArray = array.reversedArray()
```

```
fun copyArray(array: IntArray): IntArray = array.copyOf()
```

```
fun countVowels(str: String): Int {  
    val vowels = "aeiouAEIOU"  
    var count = 0  
    for (char in str) {  
        if (vowels.contains(char)) count++  
    }  
    return count  
}  
fun firstIndexOf(array: IntArray, element: Int): Int {  
    return array.indexOf(element)  
}
```

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

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

```
fun celsiusToFahrenheit(celsius: Double): Double = (celsius * 9.0 / 5.0) + 32.0
```

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

```
fun findElementByIndex(array: IntArray, index: Int): Int? {  
    return if (index in array.indices) array[index] else null  
}
```

```
fun google(number: Int): String = "G${"o".repeat(number)}gle"
```

```
fun maxMinDifference(list: List<Int>): Int {  
    if (list.isEmpty()) throw IllegalArgumentException("Список пуст")  
    return list.maxOrNull()!! - list.minOrNull()!!  
}
```

```
fun mergeLists(list1: List<Int>, list2: List<Int>): List<Int> = list1 + list2
```

```

fun checkCondition(prob: Double, prize: Double, pay: Double): Boolean = prob * prize > pay

fun isSumLess100(num1: Int, num2: Int): Boolean = num1 + num2 < 100

fun isDivisibleBy100(num: Int): Boolean = num % 100 == 0

fun calculateFrames(minutes: Int, fps: Int): Int = minutes * 60 * fps

fun checkPower(n: Double, k: Double): Boolean = k.pow(k) == n

fun equation(equation: String): Int {
    val parts = equation.split("+").map { it.trim().toIntOrNull() ?: 0 }
    return parts.sum()
}

fun sumList(list: List<Int>): Int = list.sum()

fun charCount(str: String) = str.length

fun sumArray(array: IntArray): Int = array.sum()

fun maxInArray(array: IntArray): Int = array.maxOrNull() ?: throw IllegalArgumentException("Массив пуст")

fun sortArray(array: IntArray): IntArray = array.sortedArray()

fun lastElement(array: IntArray) = array.lastOrNull() ?: throw IllegalArgumentException("Массив пуст")

fun containsElement(array: IntArray, element: Int) = array.contains(element)

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

fun minMax(array: IntArray): Pair<Int, Int> {
    if (array.isEmpty()) throw IllegalArgumentException("Массив пуст")
    return array.minOrNull()!! to array.maxOrNull()!!
}

```

```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\IntelliJ IDEA 2023.
```

Выберите номер задачи (1-42, или 0 для выхода):

1

Введите числа через запятую: 1,6,3,2

Сумма: 12

Выберите номер задачи (1-42, или 0 для выхода):

2

Введите числа через запятую: 4,2,1

Разность max и min: 3

Выберите номер задачи (1-42, или 0 для выхода):

3

Введите первый список чисел через запятую: 1,6,9,2

Введите второй список чисел через запятую: 4,9,5

Объединенный список: [1, 6, 9, 2, 4, 9, 5]

Выберите номер задачи (1-42, или 0 для выхода):

4

Введите вероятность (prob): 0

Введите приз (prize): 3

Введите плату (pay): 2

Результат: false

Выберите номер задачи (1-42, или 0 для выхода):

4

Введите вероятность (prob): 1

Введите приз (prize): 3

Введите плату (pay): 2

Результат: true

Выберите номер задачи (1-42, или 0 для выхода):

5

Введите вероятность (prob):

Введите приз (prize):

Введите плату (pay):

Результат: false

6

Введите первое число: 4

Введите второе число: 2

Сумма < 100: true

Выберите номер задачи (1-42, или 0 для выхода):

6

Введите первое число: 99

Введите второе число: 4

Сумма < 100: false

Выберите номер задачи (1-42, или 0 для выхода):

7

Введите число: 3

Делится на 100: false

Выберите номер задачи (1-42, или 0 для выхода):

7

Введите число: 10

Делится на 100: false

Выберите номер задачи (1-42, или 0 для выхода):

7

Введите число: 900

Делится на 100: true

Выберите номер задачи (1-42, или 0 для выхода):

8

Введите количество минут: 20

Введите частоту кадров (FPS): 60

Количество кадров: 72000

Выберите номер задачи (1-42, или 0 для выхода):

9

Введите n: 4

Введите k: 2

$k^k == n$: true

Выберите номер задачи (1-42, или 0 для выхода):

9

Введите n: 3

Введите k: 7

$k^k == n$: false

Выберите номер задачи (1-42, или 0 для выхода):

10

Введите строку: 1,6,2

Введите количество повторений: 2

Результат: 1,6,21,6,2

Выберите номер задачи (1-42, или 0 для выхода):

11

Введите уравнение (например, 1+1+2): 7*1+2

Результат: 2

Выберите номер задачи (1-42, или 0 для выхода):

11

Введите уравнение (например, 1+1+2): 5*7+2

Результат: 14

Выберите номер задачи (1-42, или 0 для выхода):

12

Введите количество букв 'o': 5

Результат: 6oooooogle

Выберите номер задачи (1-42, или 0 для выхода):

13

Привет, мир!

Выберите номер задачи (1-42, или 0 для выхода):

14

Введите первое число: 7

Введите второе число: 3

Сумма: 10

Выберите номер задачи (1-42, или 0 для выхода):

15

Введите первое число: 7

Введите второе число: 3

Максимум: 7

Выберите номер задачи (1-42, или 0 для выхода):

16

Введите число: 2

Четное: true

Выберите номер задачи (1-42, или 0 для выхода):

16

Введите число: 7

Четное: false

Выберите номер задачи (1-42, или 0 для выхода):

17

Введите число: 3

Факториал: 6

Выберите номер задачи (1-42, или 0 для выхода):

18

Введите число: 4

Простое: false

Выберите номер задачи (1-42, или 0 для выхода):

18

Введите число: 2

Простое: true

Выберите номер задачи (1-42, или 0 для выхода):

19

Введите числа через запятую: 2,7,3,9

Сумма: 21

Выберите номер задачи (1-42, или 0 для выхода):

20

Введите числа через запятую: 2,8,3,4

Максимум: 8

Выберите номер задачи (1-42, или 0 для выхода):

21

Введите числа через запятую: 2,7,4

Отсортированный список: [2, 4, 7]

Выберите номер задачи (1-42, или 0 для выхода):

22

Введите строку: я устал

Палиндром: false

Выберите номер задачи (1-42, или 0 для выхода):

22

Введите строку: лидер бодро, гордо бредил

Палиндром: false

Выберите номер задачи (1-42, или 0 для выхода):

22

Введите строку: 1 2 3 2 1

Палиндром: true

23

Введите строку: *долго делал*

Длина строки: 11

Выберите номер задачи (1-42, или 0 для выхода):

24

Введите строку: *3 2 5 6*

Верхний регистр: 3 2 5 6

Выберите номер задачи (1-42, или 0 для выхода):

24

Введите строку: *дом стоит*

Верхний регистр: ДОМ СТОИТ

Выберите номер задачи (1-42, или 0 для выхода):

25

Введите первую строку: *Дом есть*

Введите вторую строку: *далеко от города*

Объединенные строки: Дом естьдалеко от города

Выберите номер задачи (1-42, или 0 для выхода):

26

Введите числа через запятую: *2,6,9,2,3*

Последний элемент: 3

Выберите номер задачи (1-42, или 0 для выхода):

27

Введите числа через запятую: *3,7,1,5*

Введите элемент для поиска: *1*

Элемент найден: true

Выберите номер задачи (1-42, или 0 для выхода):

28

Введите N: *4*

Массив: [1, 2, 3, 4]

Выберите номер задачи (1-42, или 0 для выхода):

29

Введите числа через запятую: 3,8,1,6

Минимум и максимум: (1, 8)

Выберите номер задачи (1-42, или 0 для выхода):

30

Введите N: 3

Сумма от 1 до N: 6

Выберите номер задачи (1-42, или 0 для выхода):

31

Введите температуру в Цельсиях: 40

Температура в Фаренгейтах: 104.0

Выберите номер задачи (1-42, или 0 для выхода):

32

Введите строку: Сегодня солнце

Обратный порядок: ецнлос яндогеС

Выберите номер задачи (1-42, или 0 для выхода):

33

Введите числа через запятую: 3,7,2,9

Введите индекс: 2

Элемент по индексу: 2

Выберите номер задачи (1-42, или 0 для выхода):

33

Введите числа через запятую: 1

Введите индекс:

Элемент по индексу: 1

Выберите номер задачи (1-42, или 0 для выхода):

33

Введите числа через запятую: 1,7,3,9,5

Введите индекс: 1

Элемент по индексу: 7

34

Введите строку: *Тяжело долго сидеть дома*

Строка без пробелов: Тяжелодолгосидетьдома

Выберите номер задачи (1-42, или 0 для выхода):

35

Введите N: *2*

Сумма первых N чисел: 3

Выберите номер задачи (1-42, или 0 для выхода):

35

Введите N: *12*

Сумма первых N чисел: 78

Выберите номер задачи (1-42, или 0 для выхода):

36

Введите строку: *сейчас утро*

Введите подстроку: *скоро обед*

Подстрока найдена: false

Выберите номер задачи (1-42, или 0 для выхода):

36

Введите строку: *сейчас утро, скоро обед*

Введите подстроку: *скоро обед*

Подстрока найдена: true

37

Введите число: 3

3 * 1 = 3

3 * 2 = 6

3 * 3 = 9

3 * 4 = 12

3 * 5 = 15

3 * 6 = 18

3 * 7 = 21

3 * 8 = 24

3 * 9 = 27

3 * 10 = 30

Выберите номер задачи (1-42, или 0 для выхода):

38

Введите строку: *сейчас утро*

Длина строки: 11

Выберите номер задачи (1-42, или 0 для выхода):

39

Введите числа через запятую: 5,3,9,2

Перевернутый массив: [2, 9, 3, 5]

Выберите номер задачи (1-42, или 0 для выхода):

40

Введите числа через запятую: 3,8,4,6,2

Скопированный массив: [3, 8, 4, 6, 2]

Выберите номер задачи (1-42, или 0 для выхода):

41

Введите строку: *Сейчас утро*

Количество гласных: 4

Выберите номер задачи (1-42, или 0 для выхода):

42

Введите числа через запятую: 4,8,2,1,3

Введите элемент для поиска: 3

Индекс первого вхождения: 4

Выберите номер задачи (1-42, или 0 для выхода):

Выберите номер задачи (1-42, или 0 для выхода):

0

Process finished with exit code 0