

1. Генератор случайных чисел: Напишите программу, которая генерирует и выводит 10 случайных чисел от 1 до 100.

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\P
Случайные числа: [32, 82, 66, 68, 27, 21, 90, 60, 60, 66]

Process finished with exit code 0
|
```

```
fun generateRandomNumbers() {
    val numbers = List(10) { (1..100).random() }
    println("Случайные числа: $numbers")
}
fun main() {
    generateRandomNumbers()
}
```

2. Строковый анализатор: Напишите программу, которая принимает строку и выводит количество гласных и согласных букв.

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaag
Введите строку: выпыволпывп ывпы пывывп ывпывцукчсм
Гласных: 12, Согласных: 21

Process finished with exit code 0
```

```
fun analyzeString(input: String) {
    val vowels = "аеёиоуыэюяАЕЁИОУЫЭЮЯ"
    val vowelCount = input.count { it in vowels }
    val consonantCount = input.count { it.isLetter() && it !in vowels }
    println("Гласных: $vowelCount, Согласных: $consonantCount")
}
fun main() {
    print("Введите строку: ")
    val input = readLine() ?: ""
    analyzeString(input)
}
```

3. Конвертер валют: Реализуйте программу, которая конвертирует одну валюту в другую (например, доллар в евро).

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaag
Введите сумму в рублях: 100000
100000.0 рублей = 1234.567901234568 долларов
100000.0 рублей = 1063.8297872340424 евро

Process finished with exit code 0
|
```

```

fun convertToDollars(rubles: Double, rate: Double): Double {
    return rubles / rate
}
fun convertToEuros(rubles: Double, rate: Double): Double {
    return rubles / rate
}
fun main() {
    val dollarExchangeRate = 81.0
    val euroExchangeRate = 94.0
    print("Введите сумму в рублях: ")
    val rubles = readLine()?.toDoubleOrNull() ?: 0.0
    val dollars = convertToDollars(rubles, dollarExchangeRate)
    val euros = convertToEuros(rubles, euroExchangeRate)
    println("$rubles рублей = $dollars долларов")
    println("$rubles рублей = $euros евро")
}

```

4. Проверка на анаграмму: Напишите функцию, которая проверяет, являются ли две строки анаграммами.

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe
Введите сумму в рублях: 100000
100000.0 рублей = 1234.567901234568 долларов
100000.0 рублей = 1063.8297872340424 евро

Process finished with exit code 0

```

```

fun areAnagrams(str1: String, str2: String): Boolean {
    return str1.lowercase().filter { it.isLetter() }.toList().sorted() ==
        str2.lowercase().filter { it.isLetter() }.toList().sorted()
}
fun main() {
    print("Введите первую строку: ")
    val s1 = readLine() ?: ""
    print("Введите вторую строку: ")
    val s2 = readLine() ?: ""
    println("Анаграммы: ${areAnagrams(s1, s2)}")
}

```

5. Нахождение простых чисел: Реализуйте программу, которая находит и выводит все простые числа до заданного числа N.

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 202
Простые числа до 100: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]

Process finished with exit code 0
|

```

```

fun findPrimes(n: Int): List<Int> {
    return (2..n).filter { num -> (2 until num).all { i -> num % i != 0 } }
}
fun main() {
    val primes = findPrimes(100)
    println("Простые числа до 100: $primes")
}

```

6. Сортировка строк: Напишите функцию, которая принимает массив строк и сортирует его по алфавиту.

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\P
Отсортированные строки: apple, banana, cherry

Process finished with exit code 0
```

```
fun sortStrings(arr: Array<String>): Array<String> {
    return arr.sortedArray()
}

fun main() {
    val sortedArray = sortStrings(arrayOf("banana", "apple", "cherry"))
    println("Отсортированные строки: ${sortedArray.joinToString(", ")}")
}
```

7. Изменение регистра: Создайте программу, которая принимает строку и меняет регистр всех букв на противоположный.

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-
Измененный регистр: hello world

Process finished with exit code 0
```

```
fun changeCase(input: String): String {
    return input.map { if (it.isUpperCase()) it.lowercaseChar() else
it.uppercaseChar() }.joinToString("")
}

fun main() {
    val updatedCase = changeCase("HELLO World")
    println("Измененный регистр: $updatedCase")
}
```

8. Игра "Угадай число": Напишите консольную игру, в которой пользователь должен угадать случайное число от 1 до 100, а программа подсказывает, больше или меньше загаданное число.

```

Угадайте число от 1 до 100:
88
Больше!
Угадайте число от 1 до 100:
93
Меньше!
Угадайте число от 1 до 100:
91
Меньше!
Угадайте число от 1 до 100:
90
Правильно!

Process finished with exit code 0

```

```

fun guessNumber() {
    val randomNumber = (1..100).random()
    var guess: Int?
    do {
        println("Угадайте число от 1 до 100: ")
        guess = readLine()?.toIntOrNull()
        when {
            guess == randomNumber -> println("Правильно!")
            guess != null && guess < randomNumber -> println("Больше!")
            guess != null && guess > randomNumber -> println("Меньше!")
        }
    } while (guess != randomNumber)
}

fun main() {
    guessNumber()
}

```

9. Генератор паролей: Реализуйте упражнение по генерации случайного пароля заданной длины, используя цифры, буквы и специальные символы.

```

C:\Users\Student\jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\
Сгенерированный пароль: MI7D@7kR&I

Process finished with exit code 0

```

```

fun generatePassword(length: Int): String {
    val letters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
    val digits = "0123456789"
    val special = "!@#%&'()*~+-_`=|{}[\]^&*"

    val passwordBuilder = StringBuilder()
    passwordBuilder.append(letters.random())
}

```

```

passwordBuilder.append(digits.random())
passwordBuilder.append(special.random())

val allChars = letters + digits + special
for (i in 3 until length) {
    passwordBuilder.append(allChars.random())
}

return passwordBuilder.toString().toList().shuffled().joinToString("")
}
fun main() {
    val password = generatePassword(10)
    println("Сгенерированный пароль: $password")
}

```

10. По заданной строке реализуйте функцию, которая возвращает самое длинное слово в этой строке. Если есть два или более слов одинаковой длины, верните первое самое длинное слово из строки. Игнорируйте знаки препинания. Входная строка не может быть пустой или незаполненным отображением.

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin
Введите строку:
ыпколрыпол ыфпкволафио ыфпвыар
Самое длинное слово: ыфпкволафио

Process finished with exit code 0

```

```

fun findLongestWord(input: String): String {
    val words = input.split("[^\\p{L}]+".toRegex()).filter { it.isNotEmpty() }

    return words.maxByOrNull { it.length } ?: ""
}

fun main() {
    println("Введите строку:")
    val input = readLine() ?: ""
    val longestWord = findLongestWord(input)
    println("Самое длинное слово: $longestWord")
}

```