

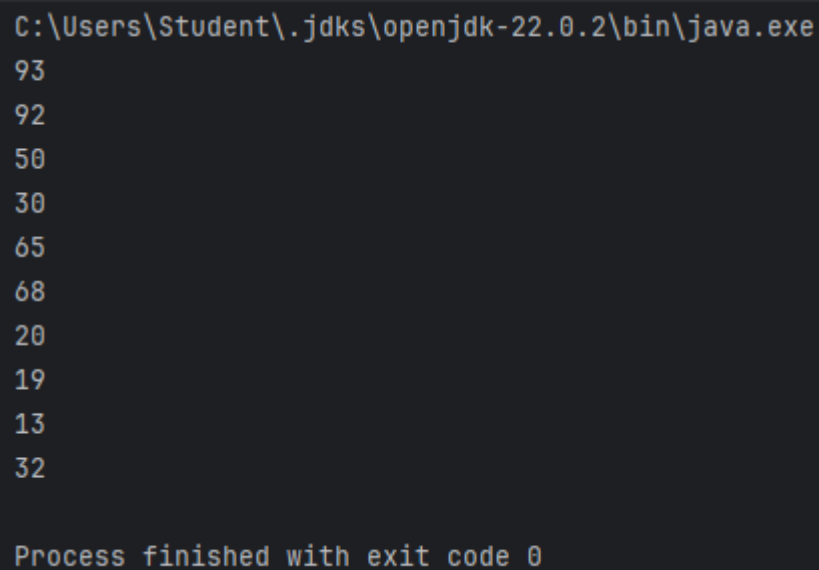
Практическая работа 9

Творческие задачи

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

```
import kotlin.random.Random

fun main(){
    repeat(10){
        println(Random.nextInt(1,101))
    }
}
```



```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe
93
92
50
30
65
68
20
19
13
32

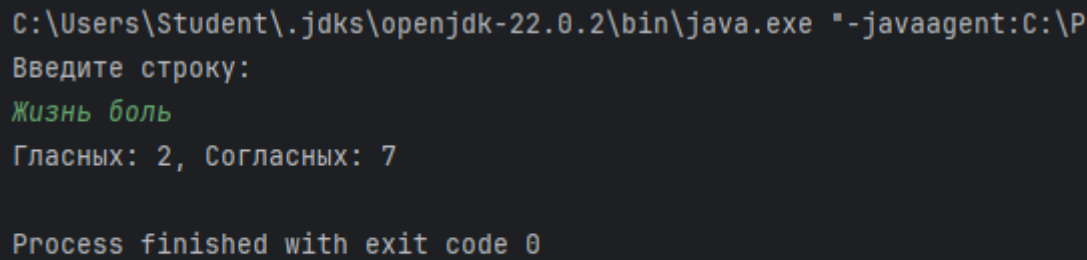
Process finished with exit code 0
```

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

```
fun main() {
    println("Введите строку:")
    val input = readLine() ?: ""
    val vowels = setOf('a','e','i','o','u','y','ф','ё','и','о','у','ы','э','ю','я')
    var vowelCount = 0
    var consonantCount = 0

    input.forEach { char ->
        val lowerChar = char.lowercaseChar()
        if (lowerChar in vowels) {
            vowelCount++
        } else if (lowerChar.isLetter()) {
            consonantCount++
        }
    }
}
```

```
println("Гласных: $vowelCount, Согласных: $consonantCount")
}
```

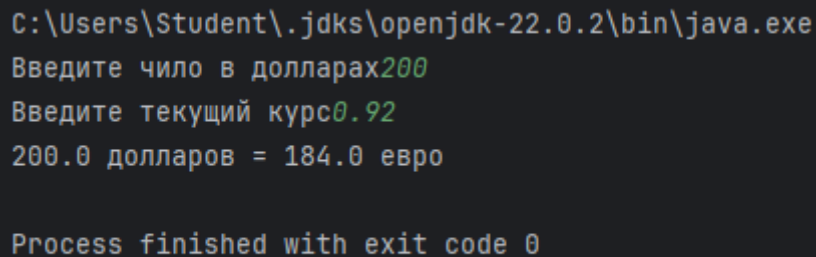


```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe -javaagent:C:\P
Введите строку:
Жизнь боль
Гласных: 2, Согласных: 7

Process finished with exit code 0
```

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

```
fun main() {
    print("Введите чило в долларах")
    val dollars = readLine()?.toDoubleOrNull()?.0.0
    print("Введите текущий курс")
    val exchangeRate = readLine()?.toDoubleOrNull()?.0.0
    val euros = dollars * exchangeRate
    println("$dollars долларов = $euros евро")
}
```



```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe
Введите чило в долларах200
Введите текущий курс0.92
200.0 долларов = 184.0 евро

Process finished with exit code 0
```

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

```
fun areAnagrams(s1: String, s2: String): Boolean {
    val cleanS1 = s1.replace("[^а-яА-Я]".toRegex(), "").lowercase().toCharArray().sorted()
    val cleanS2 = s2.replace("[^а-яА-Я]".toRegex(), "").lowercase().toCharArray().sorted()
    return cleanS1 == cleanS2
}
```

```
fun main() {
    println("Введите первую строку:")
    val str1 = readLine() ?: ""
    println("Введите вторую строку:")
    val str2 = readLine() ?: ""
    println(if (areAnagrams(str1, str2)) "Анаграммы" else "Не анаграммы")
}
```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-ja
Введите первую строку:
нора
Введите вторую строку:
рано
Анаграммы

Process finished with exit code 0
|

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bi
Введите первую строку:
жизнь
Введите вторую строку:
смерть
Не анаграммы

Process finished with exit code 0
|

```

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

```

fun findPrimes(n: Int): List<Int> {
    val sieve = BooleanArray(n + 1) { true }
    sieve[0] = false
    sieve[1] = false
    for (i in 2..n) {
        if (sieve[i]) {
            for (j in i * i..n step i) {
                sieve[j] = false
            }
        }
    }
    return sieve.indices.filter { sieve[it] }.toList()
}

fun main() {
    println("Введите число N:")
    val n = readLine()?.toIntOrNull() ?: 0
    val primes = findPrimes(n)
    println("Простые числа до $n: ${primes.joinToString()}")
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
Введите число N:
61
Простые числа до 61: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61

Process finished with exit code 0
|

```

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

```

fun main() {
    println("Введите строки через запятую:")
    val input = readln().trim()
    val strings = input.split(",").map { it.trim() }.toTypedArray()

    val sorted = strings.sortedArray()
    println("Отсортированный массив: ${sorted.contentToString()}")
}

```

```

C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Program
Введите строки через запятую:
здрaсте, вы, кто, дядя, полицейский
Отсортированный массив: [вы, дядя, здрасте, кто, полицейский]

Process finished with exit code 0
|

```

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

```

fun main() {
    println("Введите строку для инверсии регистра:")
    val input = readln()

    val result = input.map {
        if (it.isLowerCase()) it.uppercaseChar()
        else it.lowercaseChar()
    }.joinToString("")

    println("Результат: $result")
}

```

```


C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaa
Введите строку для инверсии регистра:
Привет ВАМ земляне, Я Ваш ДРУГ
Результат: ПРИВЕТ вам ЗЕМЛЯНЕ, я ВАШ дрУг

Process finished with exit code 0

```

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.

```
fun main() {  
    val secret = (1..100).random()  
    var attempts = 0  
  
    println("Угадайте число от 1 до 100:")  
    while (true) {  
        val guess = readln().toIntOrNull()  
        attempts++  
  
        when {  
            guess == null -> println("Введите число!")  
            guess < secret -> println("Больше")  
            guess > secret -> println("Меньше")  
            else -> {  
                println("Победа! Попыток: $attempts")  
                break  
            }  
        }  
    }  
}
```



```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe -javaagent...  
Угадайте число от 1 до 100:  
93  
Меньше  
54  
Меньше  
37  
Меньше  
18  
Больше  
25  
Меньше  
21  
Меньше  
19  
Победа! Попыток: 7  
  
Process finished with exit code 0
```

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

```

import kotlin.random.Random

fun main() {
    val length = getPasswordLength()
    val password = generatePassword(length)
    println("Ваш пароль: $password")
}

private fun getPasswordLength(): Int {
    while (true) {
        print("Введите длину пароля (8-80 символов): ")
        val input = readln().trim()

        when {
            input.isEmpty() -> println("Вы ничего не ввели!")
            !input.matches(Regex("\\d+")) -> println("Введите число!")
            else -> {
                val length = input.toInt()
                when {
                    length < 8 -> println("Пароль слишком короткий!")
                    length > 80 -> println("Пароль слишком длинный!")
                    else -> return length
                }
            }
        }
    }
}

fun generatePassword(length: Int): String {
    val charGroups = listOf(
        ('a'..'z').toList(),
        ('A'..'Z').toList(),
        ('0'..'9').toList(),
        listOf('!', '@', '#', '$', '%', '^', '&', '*')
    )

    val password = mutableListOf<Char>().apply {
        repeat(length) {
            add(charGroups[Random.nextInt(charGroups.size)].random())
        }
    }

    return password.shuffled().joinToString("")
}

```

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaa
Введите длину пароля (8-80 символов): 15
Ваш пароль: &6si8*Dv5v3Cg%$

Process finished with exit code 0
```

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.

```
fun main() {
    println("Введите текст для анализа:")
    val text = readln()

    val longest = text.split(Regex("[^\\p{L}-]"))
        .filter { it.isNotBlank() && it.any(Char::isLetter) }
        .maxByOrNull { it.length } ?: ""

    println("Самое длинное слово: $longest")
}
```

```
C:\Users\Student\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:0
Введите текст для анализа:
Жили негры в Африке
Самое длинное слово: Африке

Process finished with exit code 0
|
```