Практическая работа 5

1. Вывод чисел от 1 до 10: Напишите программу, которая выводит числа от 1 до 10.

fun main() {  
 *println*((1..10).*joinToString*(", "))  
}



2. Вывод четных чисел от 1 до 20: Напишите программу, которая выводит все четные числа от 1 до 20.

fun main() {  
 for (i in 2..20 *step* 2){  
 *print*(if (i<20) "$i, " else i)  
 }  
}



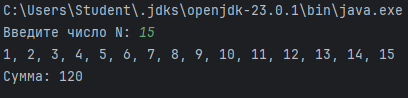
3. Сумма чисел от 1 до N: Пользователь вводит число N, программа суммирует все числа от 1 до N и выводит результат.

fun main() {  
 *print*("Введите число N: ")  
 val n = *readLine*()!!.*toInt*()  
 *print*((1..n).*joinToString*(", "))  
 *println*("\nСумма: ${(1..n).*sum*()}")  
}



4. Факториал числа: Напишите программу, которая вычисляет факториал введенного пользователем числа.

fun main() {  
 *print*("Введите число: ")  
 val n = *readLine*()!!.*toInt*()  
 *println*("Факториал числа $n = ${(1..n).*reduce***{**acc, i **->** acc \* i**}**}")  
}



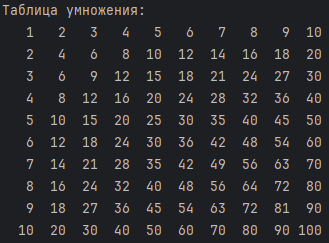
5. Проверка числа на простоту: Пользователь вводит число, программа определяет, является ли оно простым.

fun main() {  
 *print*("Введите число: ")  
 val n = *readLine*()!!.*toInt*()  
 val isPrime = n > 1 && (2..kotlin.math.*sqrt*(n.toDouble()).toInt()).*all* **{** n % **it** != 0 **}** *println*(if (isPrime) "$n - простое число" else "$n - не простое число")  
}



6. Вывод таблицы умножения: Напишите программу, которая выводит таблицу умножения от 1 до 10.

fun main() {  
 *println*("Таблица умножения:")  
 for (i in 1..10){  
 for (j in 1..10){  
 *print*("%4d".*format*(i\*j))  
 }  
 *println*()  
 }  
}



7. Фибоначчи: Сгенерируйте первые N чисел Фибоначчи (где N вводит пользователь).

fun main() {  
 *print*("Введите количество чисел: ")  
 val n = *readLine*()!!.*toInt*()  
 val nums = (1..n).*fold*(*mutableListOf*(0,1))  
 **{**list, \_ **->** list.add(list[list.size - 1] + list[list.size - 2])  
 list  
 **}** *println*(nums.*take*(n).*joinToString*(", "))  
}



8. Наибольший общий делитель (НОД): Напишите программу, которая находит НОД двух введенных чисел с использованием алгоритма Евклида.

fun main() {  
 *print*("Введите два числа через пробел: ")  
 var (a, b) = *readLine*()!!.*split*(" ").*map* **{ it**.*toInt*() **}** while (b!=0) {  
 a = b.*also* **{** b = a % b **}** }  
 *println*("НОД = $a")  
}



9. Обратный порядок: Пользователь вводит строку, и программа выводит ее в обратном порядке.

fun main() {  
 *print*("Введите строку: ")  
 *println*(*readLine*()!!.*reversed*())  
}



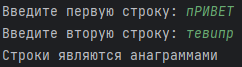
10. Сумма цифр числа: Напишите программу, которая находит сумму цифр введенного числа.

fun main() {  
 *print*("Введите число: ")  
 *println*("Сумма цифр: ${*readLine*()!!.*map***{it**.toString().*toInt*()**}**.*sum*()}")  
}



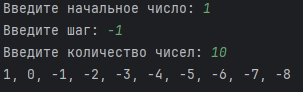
11. Анаграммы: Программа проверяет, являются ли две введенные строки анаграммами.

fun main() {  
 *print*("Введите первую строку: ")  
 val str1 = *readLine*()!!.*lowercase*().*filter* **{ it**.*isLetter*() **}**.*toCharArray*().*sorted*()  
 *print*("Введите вторую строку: ")  
 val str2 = *readLine*()!!.*lowercase*().*filter* **{ it**.*isLetter*() **}**.*toCharArray*().*sorted*()  
 *println*(if (str1 == str2) "Строки являются анаграммами" else "Строки не являются анаграммами")  
}



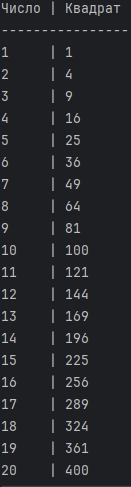
12. Числовая последовательность: Пользователь вводит начальное число и шаг, программа генерирует числовую последовательность.

fun main() {  
 *print*("Введите начальное число: ")  
 val start = *readln*().*toInt*()  
 *print*("Введите шаг: ")  
 val step = *readln*().*toInt*()  
 *print*("Введите количество чисел: ")  
 *println*((0 *until readln*().*toInt*()).*map* **{** start + **it** \* step **}**.*joinToString*(", "))  
}



13. Таблица квадратов: Выведите таблицу квадратов чисел от 1 до 20.

fun main() {  
 *println*("Число | Квадрат")  
 *println*("----------------")  
 (1..20).*forEach* **{** number **->** val square = number \* number  
 *println*(String.*format*("%-5d | %d", number, square))  
 **}**}



14. Генерация случайных чисел: Сгенерируйте и выведите 10 случайных чисел от 1 до 100.

fun main() {  
 *println*((1..10).*map***{**(1..100).*random*()**}**.*joinToString*(", "))  
}



15. Проверка палиндрома: Пользователь вводит строку, и программа проверяет, является ли она палиндромом.

fun main() {  
 *print*("Введите строку: ")  
 val txt = *readln*().*lowercase*().*filter* **{ it**.*isLetterOrDigit*() **}** *println*(if (txt == txt.*reversed*()) "Это палиндром" else "Это не палиндром")  
}





16. Сигма (сумма квадратов): Найдите сумму квадратов всех чисел от 1 до N.

fun main() {  
 *print*("Введите N: ")  
 *println*((1..*readln*().*toInt*()).*sumOf* **{ it** \* **it }**)  
}



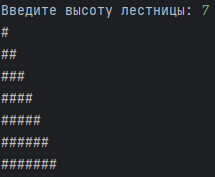
17. Вывод символов: Напишите программу, которая выводит символы строки по одному, используя циклы.

fun main() {  
 *print*("Введите строку: ")  
 *println*(*readln*().*toList*().*joinToString*(", "))  
}



18. Задача на лестницу: Напишите программу, которая выводит лестницу из символа "#" высотой N, где N задает пользователь.

fun main() {  
 *print*("Введите высоту лестницы: ")  
 (1..*readln*().*toInt*()).*forEach***{** *println*("#".*repeat*(**it**))**}**}



19. Сортировка списка: Используя цикл, напишите простую сортировку для двухзначных чисел в массиве.

fun main() {  
 val n = *arrayOf*(45, 12, 34, 85, 32, 89, 39, 99)  
 *println*(n.*sorted*().*joinToString*(", "))  
}



20.Простые числа в диапазоне: Выведите все простые числа в заданном

пользователем диапазоне.

fun main() {

    println("Введите начало диапазона:")

    val start = readLine()!!.toInt()

    println("Введите конец диапазона:")

    val end = readLine()!!.toInt()

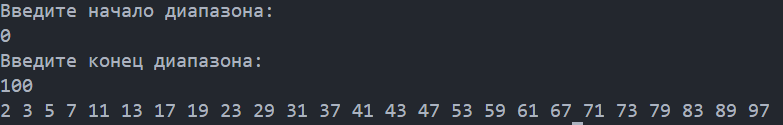
    (start..end)

        .filter { n -> (2..kotlin.math.sqrt(n.toDouble()).toInt())

            .none { n % it == 0 } && n > 1 }

        .forEach { print("$it ") }

}



21.Вывод даты: Пользователь вводит год и месяц, программа выводит все

даты в этом месяце.

import java.time.YearMonth

fun main() {

    println("Введите год:")

    val year = readLine()!!.toInt()

    println("Введите месяц (1-12):")

    val month = readLine()!!.toInt()

    YearMonth.of(year, month).let { ym ->

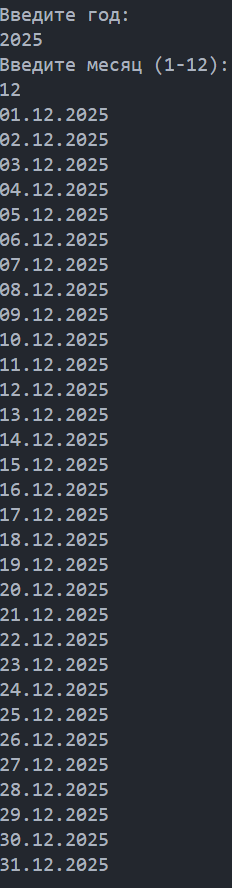
        (1..ym.lengthOfMonth()).forEach { day ->

            println("%02d.%02d.%d".format(day, month, year))

        }

    }

}



22.Угадай число: Напишите игру, в которой пользователь должен угадать

случайное число от 1 до 100.

fun main() {

    val number = (1..100).random()

    while (true) {

        val guess = readLine()!!.toInt()

        when {

            guess > number -> println("Меньше")

            guess < number -> println("Больше")

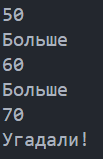
            else -> break

        }

    }

    println("Угадали!")

}



23.Сложение и умножение: Напишите программу, которая запрашивает у

пользователя две цифры и повторяет сложение или умножение, до тех

пор, пока пользователь не введет "стоп".

fun main() {

    println("Введите выражение в формате: число операция число")

    println("Пример: 5+3 или 4 \* 2")

    println("Для выхода введите 'stop'")

    while (true) {

        print("Введите выражение: ")

        val input = readLine()?.trim() ?: break

        if (input.lowercase() == "stop") {

            println("Программа завершена")

            break

        }

        try {

            val parts = input.replace(" ", "").replace("+", " + ").replace("\*", " \* ").split(" ")

            val (a, op, b) = parts

            val result = when(op) {

                "+" -> a.toInt() + b.toInt()

                "\*" -> a.toInt() \* b.toInt()

                else -> throw Exception("Неподдерживаемая операция")

            }

            println("Результат: $result")

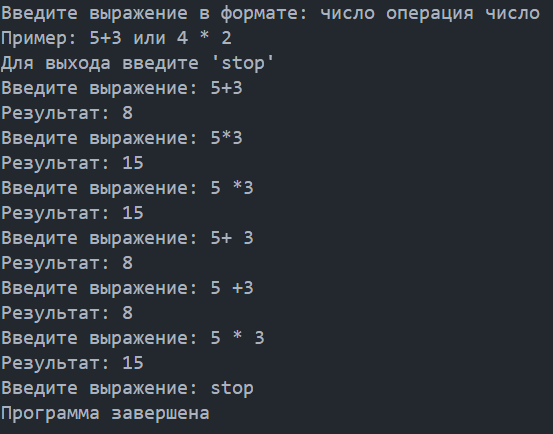
        } catch (e: Exception) {

            println("Ошибка! Проверьте формат ввода")

        }

    }

}



24.Транспонирование матрицы: Напишите программу, которая

транспонирует матрицу (двумерный массив).

fun main() {

    println("Введите размеры матрицы (строки столбцы):")

    val (rows, cols) = readLine()!!.split(" ").map { it.toInt() }

    println("Введите матрицу ${rows}x${cols} построчно:")

    val matrix = Array(rows) { readLine()!!.split(" ").map { it.toInt() }.toTypedArray() }

    println("Результат:")

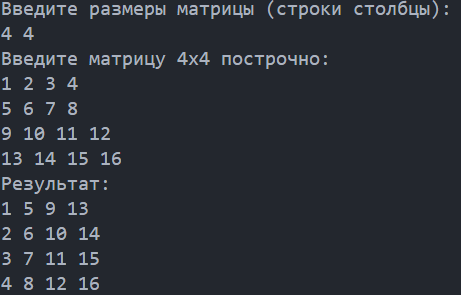
    for (i in 0 until cols) {

        for (j in 0 until rows) print("${matrix[j][i]} ")

        println()

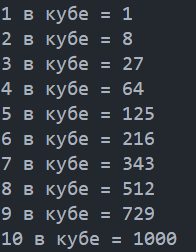
    }

}



25.Кубы чисел: Выведите кубы чисел от 1 до 10.

fun main() = (1..10).forEach { println("$it в кубе = ${it \* it \* it}") }



26.Сумма четных и нечетных чисел: Пользователь вводит N, программа

считает сумму четных и нечетных чисел от 1 до N.

fun main() {

    println("Введите N:")

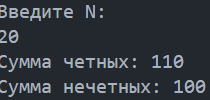
    val n = readLine()!!.toInt()

    val (even, odd) = (1..n).partition { it % 2 == 0 }

    println("Сумма четных: ${even.sum()}")

    println("Сумма нечетных: ${odd.sum()}")

}



27.Печать числа "пирамида": Напишите программу, которая выводит

"пирамиду" из чисел от 1 до N.

fun main() {

    println("Введите N:")

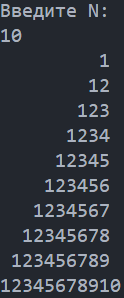
    val n = readLine()!!.toInt()

    for (i in 1..n) {

        println(" ".repeat(n - i) + (1..i).joinToString(""))

    }

}



28.Определение порядка: Программа принимает N чисел и выводит их в

порядке возрастания.

fun main() {

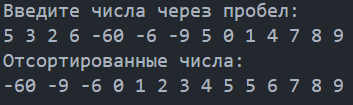
    println("Введите числа через пробел:")

    val numbers = readLine()!!.split(" ").map { it.toInt() }.sorted()

    println("Отсортированные числа:")

    println(numbers.joinToString(" "))

}



29.Сумма ряда: Напишите программу, которая находит сумму ряда 1, 1/2,

1/3, ..., 1/N.

fun main() {

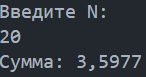
    println("Введите N:")

    val n = readLine()!!.toInt()

    val sum = (1..n).sumOf { 1.0/it }

    println("Сумма: %.4f".format(sum))

}



30.Конвертация в двоичную систему: Напишите программу, которая

конвертирует целое число в двоичную систему.

fun main() = println("Введите число:").let { println(readLine()!!.toInt().toString(2)) }

