

```

//Задача 1: Определить день недели по номеру

fun dayOfWeek(dayNumber: Int): String {
    return when (dayNumber) {
        1 -> "Понедельник"
        2 -> "Вторник"
        3 -> "Среда"
        4 -> "Четверг"
        5 -> "Пятница"
        6 -> "Суббота"
        7 -> "Воскресенье"
        else -> "Неверный номер дня недели"
    }
}

fun main() {
    val dayNumber = 3
    val day = dayOfWeek(dayNumber)
    println("День недели: $day")
}

//Задача 2: Определить тип треугольника по длинам сторон

fun triangleType(sideA: Double, sideB: Double, sideC: Double): String {
    if (sideA <= 0 || sideB <= 0 || sideC <= 0) {
        return "Некорректные длины сторон"
    }
    if ((sideA + sideB <= sideC) || (sideA + sideC <= sideB) || (sideB + sideC <= sideA)) {
        return "Треугольник не существует"
    }
    return when {
        sideA == sideB && sideB == sideC -> "Равносторонний"
        sideA == sideB || sideA == sideC || sideB == sideC -> "Равнобедренный"
        else -> "Разносторонний"
    }
}

val sideA = 3.0
val sideB = 4.0
val sideC = 5.0
val type = triangleType(sideA, sideB, sideC)
println("Тип треугольника: $type")

//Задача 3: Вывод оценок по числовым значениям

fun gradeByScore(score: Int): String {
    return when (score) {
        in 90..100 -> "Отлично"
        in 75..89 -> "Хорошо"
        in 60..74 -> "Удовлетворительно"
        in 0..59 -> "Неудовлетворительно"
        else -> "Оценка вне диапазона"
    }
}

val score = 85
val grade = gradeByScore(score)

```

```

println("Оценка: $grade")

//Задача 4: Определение времени суток

fun timeOfDay(hour: Int): String {
    return when (hour) {
        in 0..11 -> "Утро"
        in 12..15 -> "День"
        in 16..19 -> "Вечер"
        in 20..23 -> "Ночь"
        else -> "Неверное значение часа"
    }
}

val hour = 14
val time1 = timeOfDay(hour)
println("Время суток: $time1")

//Задача 5: Определить знак числа

fun signOfNumber(number: Int): String {
    return when {
        number > 0 -> "Положительное"
        number < 0 -> "Отрицательное"
        else -> "Ноль"
    }
}

val number = -5
val sign = signOfNumber(number)
println("Знак числа: $sign")

//Задача 6: Угадай число

val secretNumber = (Math.random() * 101).toInt() // Генерация случайного
числа от 0 до 100
var userGuess: Int
var attempts = 0

do {
    print("Угадайте число от 0 до 100: ")
    userGuess = readLine()!!.toInt()
    attempts++

    if (userGuess < secretNumber) {
        println("Ваше число меньше загаданного.")
    } else if (userGuess > secretNumber) {
        println("Ваше число больше загаданного.")
    }
} while (userGuess != secretNumber)

println("Поздравляю! Вы угадали число $secretNumber за $attempts
попыток.")

//Задача 7: Определение длины строки

println("Введите строку:")

```

```

        val input = readLine()!!
        val length = input.length
        println("Длина строки: $length")

//Задача 8: Определить время приготовления по типу пищи

fun cookingTime(type: String): Int {
    return when (type.toLowerCase()) {
        "мясо" -> 45
        "рыба" -> 25
        "овощи" -> 15
        else -> 0
    }
}

print("Введите тип продукта (мясо, рыба, овощи): ")
val productType = readLine()!!
val time = cookingTime(productType)
if (time > 0) {
    println("Время приготовления: $time минут")
} else {
    println("Неправильный тип продукта")
}

//Задача 9: Определение длины строки

println("Введите строку:")

fun stringLength(input: String): Int {
    return input.length
}

val input1 = readLine()!!
val length1 = stringLength(input1)
println("Длина строки: $length1")

//Задача 10: Способы оплаты

fun paymentMethod(method: String): String {
    return when (method.toLowerCase()) {
        "наличные" -> "Оплата наличными принята."
        "кредитная карта" -> "Оплата кредитной картой обработана."
        "paypal" -> "Оплата через PayPal завершена."
        else -> "Недоступный метод оплаты."
    }
}

print("Выберите способ оплаты (наличные, кредитная карта, PayPal): ")
val method = readLine()!!
val message = paymentMethod(method)
println(message)

//Задача 11: Группа крови

fun bloodGroupCompatibility(bloodGroup: Char): String {
    return when (bloodGroup) {
        'O' -> "Можно использовать для всех групп крови."
    }
}

```

```

        'A' -> "Можно использовать для групп A и АВ."
        'B' -> "Можно использовать для групп B и АВ."
        'A', 'B' -> "Можно использовать только для группы АВ."
        else -> "Группа крови введена неверно."
    }
}

print("Введите группу крови (A, B, АВ, O): ")
val group = readLine()!![0].uppercaseChar()
val compatibility = bloodGroupCompatibility(group)
println(compatibility)

//Задача 12: Национальности

val countriesToNationalities = mapOf(
    "США" to "американец",
    "Россия" to "русский",
    "Япония" to "японец",
    "Германия" to "немец",
    "Франция" to "француз",
    "Китай" to "китаец",
    "Дота" to "дотер",
    "СаторуГоджо" to "Фиолетович"
)

print("Введите название страны (США, Россия, Япония и т.д.): ")
val country = readLine()!!.capitalize()

val nationality = countriesToNationalities[country]
if (nationality != null) {
    println("Национальность: $nationality")
} else {
    println("Страна не найдена")
}

//Задача 13: Коды ошибок

fun errorCode(code: Int): String {
    return when (code) {
        100 -> "Ошибка сети"
        200 -> "Ошибка сервера"
        300 -> "Ошибка клиента"
        else -> "Неизвестная ошибка"
    }
}

print("Введите код ошибки (100, 200, 300): ")
val code = readLine()!!.toInt()
val errorMessage = errorCode(code)
println(errorMessage)
}

```