

1.

```
fun getDayOfWeek(dayNumber: Int):  
    String {return when(dayNumber){  
        1 -> "Понедельник"  
        2 -> "Вторник"  
        3 -> "Среда"  
        4 -> "Четверг"  
        5 -> "Пятница"  
        6 -> "Суббота"  
        7 -> "Воскресенье"  
        else -> "Некорректный номер дня"  
    }}  
fun main() {  
    print("Введите номер дня(1-7): ")  
    val dayNumber = readLine()!!.toInt()  
    println(getDayOfWeek(dayNumber))  
}
```

2.

```
fun getTriangleType(a: Double, b: Double, c: Double):  
    String{return when{  
        a<=0||b<=0||c<=0 -> "Некорректные значения сторон"  
        a + b <= c || b+c <= a || a+c <= b -> "Треугольник не существует"  
        a==b && b==c -> "Равносторонний треугольник"  
        a==b || b==c || a==c -> "Равнобедренный треугольник"  
        a*a + b*b == c*c || b*b + c*c == a*a || a*a + c*c == b*b -> "Прямоугольный треугольник"  
        else -> "Разносторонний треугольник"  
    }}  
fun main() {  
    println("Введите длину первой стороны: ")  
    val a = readLine()!!.toDouble()  
    println("Введите длину второй стороны: ")  
    val b = readLine()!!.toDouble()  
    println("Введите длину третьей стороны: ")  
    val c = readLine()!!.toDouble()  
    if (a != null && b != null && c != null) {  
        println(getTriangleType(a, b, c))  
    } else {  
        println("Некорректные значения сторон")  
    }  
}
```

3.

```
fun getGrade(score: Int): String {
    return when(score) {
        2 -> "Неудовлетворительно"
        3 -> "Удовлетворительно"
        4 -> "Хорошо"
        5 -> "Отлично"
        else -> "Некорректное значение"
    }
}

fun main() {
    println("Введите оценку")
    val score = readLine()!!.toInt()
    println(getGrade(score))
}
```

4.

```
fun getTimeOfDay(hour: Int): String {
    return when (hour) {
        in 0 ≤ .. ≤ 5 -> "Ночь"
        in 6 ≤ .. ≤ 11 -> "Утро"
        in 12 ≤ .. ≤ 17 -> "День"
        in 18 ≤ .. ≤ 23 -> "Вечер"
        else -> "Некорректное время"
    }
}

fun main(){
    println("Введите текущий час: ")
    val hour = readLine()!!.toInt()
    println(getTimeOfDay(hour))
}
```

5.

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

fun main(){
    println("Введите число: ")
    val number = readLine()!!.toInt()
    println(getNumberSign(number))
}
```

6.

```
fun main() {
    val number = (1 ≤ .. ≤ 100).random()
    while (true) {
        println("Угадайте число от 1 до 100:")
        val guess = readLine()?.toIntOrNull() ?: continue
        when {
            guess < number -> println("Больше")
            guess > number -> println("Меньше")
            else -> {
                println("Угадали!")
                break
            }
        }
    }
}
```

7.

```
fun main() {
    println("Введите строку:")
    val text = readLine()
    println("Длина строки: ${text?.length ?: 0}")
}
```

8.

```
fun cookingTime(foodType: String): String {
    return when (foodType.toLowerCase()) {
        "паста" -> "15 минут"
        "мясо" -> "30 минут"
        "овощи" -> "10 минут"
        else -> "Неизвестный тип пищи"
    }
}

fun main() {
    println(cookingTime( foodType: "мясо")) // Вывод: 30 минут
}
```

9.

```
fun processPayment(paymentMethod: String): String {
    return when(paymentMethod.lowercase()) {
        "наличные" -> """
Оплата наличными:
1. Сумма к оплате будет указана на кассе
2. Подготовьте точную сумму или близкую к ней
3. Получите сдачу и кассовый чек
4. Проверьте правильность суммы и сдачи
Комиссия: нет
Время обработки: моментально
"""
        "кредитная карта", "карта" -> """
Оплата картой:
1. Вставьте карту в терминал или приложите к считывателю
2. Выберите способ оплаты (кредит/дебет)
3. Введите PIN-код при необходимости
4. Дождитесь подтверждения операции
5. Сохраните чек
Комиссия: согласно условиям банка
Время обработки: 1-2 минуты
"""
        "paypal" -> """
Оплата через PayPal:
1. Войдите в свой аккаунт PayPal
2. Проверьте email получателя платежа
3. Укажите сумму перевода
4. Выберите источник средств
5. Подтвердите транзакцию
Комиссия: 0-2% в зависимости от типа операции
Время обработки: 1-24 часа
"""
        else -> """
Доступные способы оплаты:
1. Наличные
2. Кредитная карта
3. PayPal

Пожалуйста, выберите один из указанных способов оплаты.
"""
    }
}

fun main() {
    println("Выберите способ оплаты (наличные, кредитная карта, PayPal):")
    val paymentMethod = readLine()
    println(processPayment(paymentMethod: paymentMethod ?: ""))
}
```

10.

```
fun paymentMethod(method: String): String {
    return when (method.toLowerCase()) {
        "наличные" -> "Вы выбрали оплату наличными."
        "кредитная карта" -> "Вы выбрали оплату кредитной картой."
        "paypal" -> "Вы выбрали оплату через PayPal."
        else -> "Некорректный способ оплаты."
    }
}

fun main() {
    println(paymentMethod( method: "кредитная карта")) // Вывод: Вы выбрали оплату кредитной картой.
}
```

11.

```
fun bloodTypeCompatibility(bloodType: String): String {
    return when (bloodType.toUpperCase()) {
        "A" -> "Можно переливать: A, AB"
        "B" -> "Можно переливать: B, AB"
        "AB" -> "Можно переливать: AB"
        "O" -> "Можно переливать: A, B, AB, O"
        else -> "Некорректная группа крови"
    }
}

fun main() {
    println(bloodTypeCompatibility( bloodType: "A")) // Вывод: Можно переливать: A, AB
}
```

12.

```
fun nationalityByCountry(country: String): String {
    return when (country.toUpperCase()) {
        "США" -> "Американец/Американка"
        "РОССИЯ" -> "Русский/Русская"
        "ЯПОНИЯ" -> "Японец/Японка"
        "ГЕРМАНИЯ" -> "Немец/Немка"
        else -> "Неизвестная страна"
    }
}

fun main() {
    println(nationalityByCountry( country: "Россия")) // Вывод: Русский/Русская
}
```

13.

```
fun errorCodeMessage(code: Int): String {  
    return when (code) {  
        100 -> "Ошибка сети"  
        200 -> "Ошибка сервера"  
        300 -> "Ошибка доступа"  
        else -> "Неизвестный код ошибки"  
    }  
}  
  
fun main() {  
    println(errorCodeMessage( code: 200)) // Вывод: Ошибка сервера  
}
```