


```

fun main() {
    // 1. Калькулятор
    println("Введите первое число:")
    val num1 = readLine()?.toDoubleOrNull() ?: return
    println("Введите оператор (+, -, *, /):")
    val operator = readLine()
    println("Введите второе число:")
    val num2 = readLine()?.toDoubleOrNull() ?: return
    val result = when (operator) {
        "+" -> num1 + num2
        "-" -> num1 - num2
        "*" -> num1 * num2
        "/" -> num1 / num2
        else -> {
            println("Неверный оператор")
            return
        }
    }
    println("Результат: $result")
}

```

```

// 2. Палиндром
fun  isPalindrome(word: String): Boolean {
    return word == word.reversed()
}

fun main() {
    println("Введите слово:")
    val word = readLine() ?: return

    if (isPalindrome(word)) {
        println("$word является палиндромом")
    } else {
        println("$word не является палиндромом")
    }
}

```

```
// 3. Очки команды
fun calculatePoints(wins: Int, draws: Int, losses: Int): Int {
    return wins * 3 + draws * 1 + losses * 0
}

fun main() {
    println("Введите количество побед:")
    val wins = readLine()?.toIntOrNull() ?: return

    println("Введите количество ничьих:")
    val draws = readLine()?.toIntOrNull() ?: return

    println("Введите количество поражений:")
    val losses = readLine()?.toIntOrNull() ?: return

    val points = calculatePoints(wins, draws, losses)
    println("Команда набрала $points очков")
}
```

```
// 4. Самое маленькое число в списке
fun findMinNumber(numbers: List<Int>): Int? {
    return numbers.minOrNull()
}

fun main() {
    println("Введите список чисел через пробел:")
    val input = readLine() ?: return
    val numbers = input.split(...delimiters: " ").mapNotNull { it.toIntOrNull() }

    val minNumber = findMinNumber(numbers)
    if (minNumber != null) {
        println("Самое маленькое число: $minNumber")
    } else {
        println("Список пуст или содержит нечисловые значения")
    }
}
```

```
// 5. Равенство двух чисел
fun areNumbersEqual(num1: Int, num2: Int): Boolean {
    return num1 == num2
}

fun main() {
    println("Введите первое число:")
    val num1 = readLine()?.toIntOrNull() ?: return

    println("Введите второе число:")
    val num2 = readLine()?.toIntOrNull() ?: return

    if (areNumbersEqual(num1, num2)) {
        println("Числа равны")
    } else {
        println("Числа не равны")
    }
}
}
```

```
// 6. 21
fun playBlackjack() {
    val deck = mutableListOf(2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10, 11)
    deck.shuffle()
    var playerScore = 0
    var dealerScore = 0
    playerScore += deck.removeAt(index: 0)
    playerScore += deck.removeAt(index: 0)
    dealerScore += deck.removeAt(index: 0)
    dealerScore += deck.removeAt(index: 0)
    println("Ваши карты: $playerScore")
    println("Карты дилера: $dealerScore")
    while (playerScore < 21) {
        println("Хотите взять еще карту? (да/нет)")
        val choice = readLine() ?: return
        if (choice.equals(other: "да", ignoreCase = true)) {
            playerScore += deck.removeAt(index: 0)
            println("Ваши карты: $playerScore")
        } else {
            break
        }
    }
}
}
```

```
while (dealerScore < 17) {  
    dealerScore += deck.removeAt(index: 0)  
}  
println("Ваши карты: $playerScore")  
println("Карты дилера: $dealerScore")  
when {  
    playerScore > 21 -> println("Вы проиграли")  
    dealerScore > 21 -> println("Вы выиграли")  
    playerScore > dealerScore -> println("Вы выиграли")  
    playerScore < dealerScore -> println("Вы проиграли")  
    else -> println("Ничья")  
}  
}  
fun main() {  
    playBlackjack()  
}
```