1 задание

```kotlin
fun main() {
    println("Введите первое число:")
    val number1 = readLine()?.toDoubleOrNull() ?: return println("Некорректный ввод")

    println("Введите второе число:")
    val number2 = readLine()?.toDoubleOrNull() ?: return println("Некорректный ввод")

    println("Выберите операцию (+, -, *, /):")
    val operation = readLine()

    val result = when (operation) {
        "+" -> number1 + number2
        "-" -> number1 - number2
        "*" -> number1 * number2
        "/" -> {
            if (number2 != 0.0) number1 / number2 else "На ноль делить нельзя"
        }
        else -> "Некорректная операция"
    }

    println("Результат: $result")
}
```

```
Введите первое число:
124124
Введите второе число:
2
Выберите операцию (+, -, *, /):
*
Результат: 248248.0

Process finished with exit code 0
```

2 задание

```
fun main() {
    println("Введите слово:")
    val input = readLine()?.trim() ?: return println("Некорректный ввод")

    val isPalindrome = input.equals(input.reversed(), ignoreCase = true)

    if (isPalindrome) {
        println("$input является палиндромом.")
    } else {
        println("$input не является палиндромом.")
    }
}
```

```
Введите слово:
поп
поп является палиндромом.

Process finished with exit code 0
```

3 задание

```
fun calculatePoints(wins: Int, draws: Int, losses: Int): Int {
    return wins * 3 + draws
}

fun main() {
    val wins = 5
    val draws = 6
    val losses = 2
    val points = calculatePoints(wins, draws, losses)
    println("Очки команды: $points")
}
```

```
C:\Users\Student\.jdks\openjdk-24\bin\java.exe "-javaagent:C:\Pro
Очки команды: 21
```

```kotlin
fun findMinimum(numbers: List<Int>): Int? {
    return numbers.minOrNull()
}

fun main() {
    val numberList = listOf(5, 3, 8, 1, 4)
    val minimum = findMinimum(numberList)
    println("Самое маленькое число: $minimum")
}
```

```
Самое маленькое число: 1

Process finished with exit code 0
```

```kotlin
fun areEqual(num1: Int, num2: Int): Boolean {
    return num1 == num2
}

fun main() {
    val number1 = 10
    val number2 = 10
    val result = areEqual(number1, number2)
    println("Числа равны: $result")
}
```

```
Числа равны: true

Process finished with exit code 0
```

4 задание

```
fun main() {
    var playerScore = 0
    var dealerScore = 0
    val deck = (1 ≤ .. ≤ 11).toList() + (1 ≤ .. ≤ 10).toList().repeat( times: 4)

    while (true) {
        playerScore += drawCard(deck)
        println("Ваш счёт: $playerScore")

        if (playerScore > 21) {
            println("Вы перебрали! Игра окончена.")
            break
        }

        println("Хотите взять ещё карту? (y/n)")
        val input = readLine()
        if (input != "y") break
    }

    if (playerScore <= 21) {
        while (dealerScore < 17) {
            dealerScore += drawCard(deck)
        }
        println("Счёт дилера: $dealerScore")

        when {
            dealerScore > 21 -> println("Дилер перебрал! Вы выиграли!")
            dealerScore == playerScore -> println("Ничья!")
            dealerScore > playerScore -> println("Дилер выиграл!")
            else -> println("Вы выиграли!")
        }
    }
}

fun drawCard(deck: List<Int>): Int {
    val card = deck.random()
    println("Вы вытянули карту: $card")
    return card
}
```

```
    return card
}

fun List<Int>.repeat(times: Int): List<Int> {
    return this.flatMap { List(times) { it } }
}
```

```
Хотите взять ещё карту? (y/n)
y
Вы вытянули карту: 2
Ваш счёт: 5
Хотите взять ещё карту? (y/n)
y
Вы вытянули карту: 3
Ваш счёт: 8
Хотите взять ещё карту? (y/n)
y
Вы вытянули карту: 0
Ваш счёт: 8
Хотите взять ещё карту? (y/n)
y
Вы вытянули карту: 3
Ваш счёт: 11
Хотите взять ещё карту? (y/n)
y
Вы вытянули карту: 11
Ваш счёт: 22
Вы перебрали! Игра окончена.
y
Process finished with exit code 0
```