

```

fun printNumber(n: Int): List<Int> {
    // Если n равно 0, возвращаем пустой список
    if (n == 0) return emptyList()

    // Создаем список от n до 1 с шагом -1
    return (n downTo 1).toList()
}

fun pyramid(n: Int) {
    for (i in 1..n) {
        // Количество символов # на текущем уровне
        val hashes = i * 2 - 1
        // Количество пробелов с каждой стороны
        val spaces = n - i

        // Создаем строку с пробелами и символами #
        val level = " ".repeat(spaces) + "#".repeat(hashes) + "
    ".repeat(spaces)
        println(level)
    }
}

fun caesarCipher(text: String, shift: Int): String {
    return text.map { char ->
        when {
            char.isUpperCase() -> {
                // Шифруем заглавные буквы
                val start = 'A'.toInt()
                ((char.toInt() - start + shift) % 26) + start).toChar()
            }
            char.isLowerCase() -> {
                // Шифруем строчные буквы
                val start = 'a'.toInt()
                ((char.toInt() - start + shift) % 26) + start).toChar()
            }
            else -> char // Не буквы оставляем как есть
        }
    }.joinToString("")
}

fun fizzBuzz(n: Int): List<Any> {
    return (1..n).map { num ->
        when {
            num % 15 == 0 -> "БизллБизлл" // Кратно 3 и 5
            num % 3 == 0 -> "Физллл"      // Кратно 3
            num % 5 == 0 -> "Бизлллл"     // Кратно 5
            else -> num                    // Остальные числа
        }
    }
}

fun main() {
    println(printNumber(22)) // []
    println(printNumber(17)) // [2, 1]
    println(printNumber(7))  // [5, 4, 3, 2, 1]

    pyramid(100)
    // #

    pyramid(2)
    // #
    // ###

    pyramid(3)

```

```
//      #
//      ###
//      #####

println(caesarCipher("Hello, World!", 3)) // Koor, Zruog!
println(caesarCipher("Koor, Zruog!", -3)) // Hello, World!

println(fizzBuzz(5)) // [1, 2, Физлл, 4, Бизллл]
println(fizzBuzz(16)) // [1, 2, Физлл, 4, Бизллл, Физлл, 7, 8, Физлл,
Бизллл, 11, Физлл, 13, 14, БизллБизлл, 16]
}
```