

1.

```
fun printNumber(n: Int): List<Int> {
    if (n <= 0) {
        return emptyList()
    }

    val result = mutableListOf<Int>()
    for (i in n downTo 1) {
        result.add(i)
    }
    return result
}

fun main() {
    println(printNumber(0))
    println(printNumber(2))
    println(printNumber(5))
}
```

2.

```
fun generatePyramid(n: Int): List<String> {
    if (n <= 0) {
        return emptyList()
    }

    val pyramid = mutableListOf<String>()
    val maxWidth = 2 * n - 1

    for (i in 1..n) {
        val numHashes = 2 * i - 1
        val numSpaces = (maxWidth - numHashes) / 2

        val row = " ".repeat(numSpaces) + "#".repeat(numHashes) + "
".repeat(numSpaces)
        println(row)
        pyramid.add(row)
    }

    return pyramid
}

fun main() {
    generatePyramid(3)
    println("----")
    generatePyramid(5)
    println("----")
    generatePyramid(1)
    println("----")
    generatePyramid(0)
}
```

3.

```
fun caesarCipher(text: String, shift: Int): String {
    val shiftedAlphabet = ('a'..'z').toList().let {
        val shifted = it.subList(shift % 26, it.size) + it.subList(0, shift %
26)
        shifted.joinToString("")
    }

    val result = StringBuilder()
    for (char in text) {
        when {
```

```

        char.isLowerCase() -> {
            val index = char - 'a'
            result.append(shiftedAlphabet[index])
        }
        char.isUpperCase() -> {
            val lowerChar = char.lowercaseChar()
            val index = lowerChar - 'a'
            result.append(shiftedAlphabet[index].uppercaseChar())
        }
        else -> {
            result.append(char)
        }
    }
}
return result.toString()
}

fun main() {
    val text = "Hello, World!"
    val shift = 3
    val encryptedText = caesarCipher(text, shift)
    println("Исходный текст: $text")
    println("Зашифрованный текст: $encryptedText")

    val decryptedText = caesarCipher(encryptedText, 26 - shift)
    println("Расшифрованный текст: $decryptedText")

    val textWithSpecialChars = "abc123XYZ"
    val encryptedTextWithSpecialChars = caesarCipher(textWithSpecialChars, 1)
    println("Исходный текст: $textWithSpecialChars")
    println("Зашифрованный текст: $encryptedTextWithSpecialChars")
}

```

4.

```

fun fizzBuzz(n: Int): List<String> {
    val result = mutableListOf<String>()
    for (i in 1..n) {
        when {
            i % 3 == 0 && i % 5 == 0 -> result.add("FizzBuzz")
            i % 3 == 0 -> result.add("Fizz")
            i % 5 == 0 -> result.add("Buzz")
            else -> result.add(i.toString())
        }
    }
    return result
}

fun main() {
    println(fizzBuzz(15))
}

fun fizzBuzz(n: Int): List<String> {
    val result = mutableListOf<String>()
    for (i in 1..n) {
        when {
            i % 3 == 0 && i % 5 == 0 -> result.add("FizzBuzz")
            i % 3 == 0 -> result.add("Fizz")
            i % 5 == 0 -> result.add("Buzz")
            else -> result.add(i.toString())
        }
    }
    return result
}

fun main() {

```

```
println(fizzBuzz(15))  
}
```