

Задание 1. При заданном натуральном числе  $n$  реализуйте функцию, которая возвращает список, представляющий все числа от  $n$  до 1. Если  $n$  значение равно нулю, то должен быть возвращен пустой список.

```
fun printNumber(n: Int): List<Int> {  
    return if (n <= 0) {  
        emptyList()  
    } else {  
        (n downTo 1).toList()  
    }  
}
```

```
fun main() {  
    println(printNumber(0))  
    println(printNumber(2))  
    println(printNumber(5))  
    println(printNumber(100))  
}
```

Задание 2. Учитывая положительное целое число,  $n$  реализуйте функцию, которая возвращает список, представляющий форму `generatePyramid` с  $n$  уровнями. Функция должна консольно регистрировать форму `generatePyramid` с  $N$  уровнями, используя символ `#`. Убедитесь, что в `generatePyramid` есть пробелы как с левой, так и с правой стороны.

```
fun generatePyramid(n: Int): List<String> {  
    val pyramid = mutableListOf<String>()  
  
    for (i in 1..n) {  
        val spaces = " ".repeat(n - i)  
        val hashes = "#".repeat(2 * i - 1)
```

```

        pyramid.add("$spaces$hashes$spaces")
    }
    return pyramid
}

fun main() {
    println("Введите положительное целое число n:")
    val input = readLine()
    val n = input?.toIntOrNull()

    if (n != null && n > 0) {
        val pyramid = generatePyramid(n)
        pyramid.forEach { println(it) }
    } else {
        println("Ошибка: введите корректное положительное целое число.")
    }
}

```

Задание 3. Написать программу, реализующую Шифр цезаря

```

fun caesarCipher(text: String, shift: Int): String {
    val result = StringBuilder()
    val normalizedShift = shift % 26 // Нормализуем сдвиг для алфавита

    for (char in text) {
        when {
            char.isUpperCase() -> {
                val shiftedChar = ((char - 'A' + normalizedShift + 26) % 26 +
                'A'.toInt()).toChar()
                result.append(shiftedChar)
            }
        }
    }
}

```

```

    }
    char.isLowerCase() -> {
        val shiftedChar = ((char - 'a' + normalizedShift + 26) % 26 +
'a'.toInt()).toChar()
        result.append(shiftedChar)
    }
    else -> {
        result.append(char)
    }
}
}

return result.toString()
}

```

```

fun main() {
    println("Введите текст для шифрования:")
    val inputText = readLine() ?: ""

    println("Введите величину сдвига:")
    val shiftInput = readLine()?.toIntOrNull()

    if (shiftInput != null) {
        val encryptedText = caesarCipher(inputText, shiftInput)
        println("Зашифрованный текст: $encryptedText")
    } else {
        println("Ошибка: введите корректное число для величины сдвига.")
    }
}
}

```

Задание 4. При заданном натуральном числе  $n$  реализуем функцию, которая возвращает список чисел от 1 до  $n$ . Однако для чисел, кратных трем, список должен содержать слово Физллл вместо числа, а для чисел, кратных пяти, список должен содержать слово Бизлллл. Для чисел, кратных трем и пяти, список должен содержать ВизллБизлл.

```
fun fizBuzz(n: Int): List<String> {  
    val result = mutableListOf<String>()  
  
    for (i in 1..n) {  
        when {  
            i % 15 == 0 -> result.add("ВизллБизлл") // Кратные 15 (3 и 5)  
            i % 3 == 0 -> result.add("Физллл")      // Кратные 3  
            i % 5 == 0 -> result.add("Бизлллл")     // Кратные 5  
            else -> result.add(i.toString())        // Остальные числа  
        }  
    }  
  
    return result  
}  
  
fun main() {  
    println("Введите натуральное число n:")  
    val input = readLine()?.toIntOrNull()  
  
    if (input != null && input > 0) {  
        val resultList = fizBuzz(input)  
        println(resultList)  
    } else {
```

```
println("Ошибка: введите корректное натуральное число.")  
}  
}
```