

## Лабораторная работа №10

1.

```
fun printNumber(n: Int): List<Int> {  
    return if (n > 0) {  
        (n downTo 1).toList()  
    } else {  
        emptyList()  
    }  
}  
  
fun main() {  
    println(printNumber(0)) // []  
    println(printNumber(2)) // [2, 1]  
    println(printNumber(5)) // [5, 4, 3, 2, 1]  
}
```

2.

```
fun pyramid(n: Int) {  
    for (i in 0 until n) {  
        val spaces = " ".repeat(n - i - 1)  
        val hashes = "#".repeat(2 * i + 1)  
        println("$spaces$hashes$spaces")  
    }  
}  
  
fun main() {  
    pyramid(1)  
    println()  
    pyramid(2)  
    println()  
    pyramid(3)  
}
```

3.

```
fun main() {  
    println("Выберите действие:")
```

```

println("1. Зашифровать")
println("2. Расшифровать")
val choice = readln().toIntOrNull() ?: run {
    println("Некорректный выбор")
    return
}

println("Введите текст:")
val text = readln()

println("Введите сдвиг (целое число):")
val shift = readln().toIntOrNull() ?: run {
    println("Некорректный сдвиг")
    return
}

val result = when (choice) {
    1 -> caesarCipher(text, shift)
    2 -> caesarCipher(text, -shift)
    else -> {
        println("Некорректный выбор")
        return
    }
}

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

fun caesarCipher(text: String, shift: Int): String {
    return text.map { char ->
        when {

```

```

char.isUpperCase() -> {
    val base = 'A'.code
    ((char.code - base + shift).mod(26) + base).toChar()
}
char.isLowerCase() -> {
    val base = 'a'.code
    ((char.code - base + shift).mod(26) + base).toChar()
}
else -> char
}
}.joinToString("")
}

```

4.

```

fun fizzBuzz(n: Int): List<Any> {
    val result = mutableListOf<Any>()

    for (i in 1..n) {
        when {
            i % 3 == 0 && i % 5 == 0 -> result.add("ВизллБизлл")
            i % 3 == 0 -> result.add("Физллл")
            i % 5 == 0 -> result.add("Бизлллл")
            else -> result.add(i)
        }
    }

    return result
}

```

```

fun main() {
    val result1 = fizzBuzz(5)
    println(result1) // [1, 2, "Физллл", 4, "Бизлллл"]
}

```

```
val result2 = fizzBuzz(16)

println(result2) // [1, 2, "Физлл", 4, "Бизллл", "Физлл", 7, 8, "Физлл", "Бизллл", 11,
"Физлл", 13, 14, "ВизллБизлл", 16]
}
```