

Практическая работа 10

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.

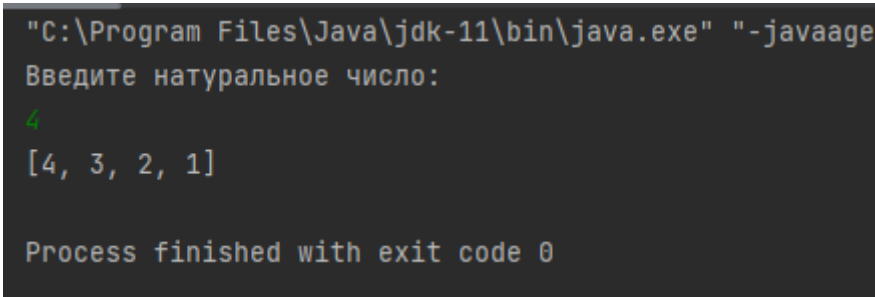
Примеры

```
printNumber(0) // []
```

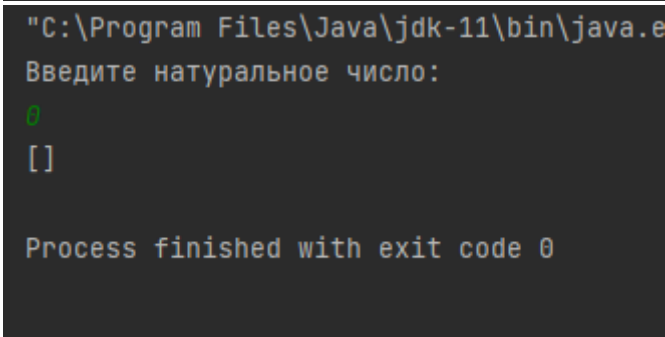
```
printNumber(2) // [2, 1]
```

```
printNumber(5) // [5, 4, 3, 2, 1]
```

```
fun printNumber(): List<Int> {  
    println("Введите натуральное число:")  
    val n = readLine()?.toIntOrNull() ?: 0  
    return if (n <= 0) emptyList() else (n downTo 1).toList()  
}  
  
fun main(){  
    println(printNumber())  
}
```



```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaage  
Введите натуральное число:  
4  
[4, 3, 2, 1]  
  
Process finished with exit code 0
```



```
"C:\Program Files\Java\jdk-11\bin\java.e  
Введите натуральное число:  
0  
[]  
  
Process finished with exit code 0
```

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.

Примеры

```
pyramid(1) '#'
```

```
pyramid(2) ' # '###'
```

```
pyramid(3) ' # ' '### '#####'
```

```
fun pyramid(): List<String> {  
    println("Введите количество уровней пирамиды:")  
    val n = readLine()?.toIntOrNull() ?: 0  
    val result = mutableListOf<String>()  
    for (i in 1..n) {
```

```

        val spaces = " ".repeat(n - i)
        val hashes = "#".repeat(2 * i - 1)
        val line = "$spaces$hashes$spaces"
        result.add(line)
        println(line) // Выводим пирамиду в консоль
    }
    return result
}
fun main(){
    pyramid()
}

```

```

"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaage
Введите количество уровней пирамиды:
8
      #
     ###
    #####
   #####
  #####
 #####
#####
#####
#####

Process finished with exit code 0

```

3. Написать программу, реализующую Шифр цезаря

```

fun main() {
    print("Введите текст: ")
    val text = readLine() ?: ""
    print("Введите сдвиг: ")
    val shift = readLine()?.toIntOrNull() ?: 0

    val result = text.map { c ->
        when {
            c in 'A'..'Z' -> 'A' + (c - 'A' + shift) % 26
            c in 'a'..'z' -> 'a' + (c - 'a' + shift) % 26
            c in 'А'..'Я' -> 'А' + (c - 'А' + shift) % 33
            c in 'а'..'я' -> 'а' + (c - 'а' + shift) % 33
            else -> c
        }
    }.joinToString("")

    println("Результат: $result")
}

```

```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\IntelliJ IDEA 2023.1.
Введите текст: привет, boss
Введите сдвиг: 3
Результат: тулеих, егvv

Process finished with exit code 0
```

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 Бизллл. Для чисел, кратных трем и пяти, список должен содержать ВизллБизлл.

Примеры

```
fizzBuzz(5) // [1, 2, " Физллл ", 4, " Бизллл "]
```

```
fizzBuzz(16) // [1, 2, " Физллл ", 4, " Бизллл ", " Физллл ", 7, 8, " Физллл ", " Бизллл ", 11, " Физллл ", 13, 14, " ВизллБизлл ", 16]
```

```
fun fizBiz(): List<Any>{
    println("Введите натурально число: ")
    val n = readLine()?.toIntOrNull():0
    return if (n <= 0) emptyList() else (1..n).map { num ->
        when{
            num % 15 == 0 -> "ВизллБизлл"
            num % 3 == 0 -> "Физллл"
            num % 5 == 0 -> "Бизллл"
            else -> num
        }
    }
}

fun main(){
    println(fizBiz())
}
```

```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\IntelliJ IDEA 2023.1.3\lib\idea_rt.jar=60598:C:\Inte
Введите натурально число:
16
[1, 2, Физллл, 4, Бизллл, Физллл, 7, 8, Физллл, Бизллл, 11, Физллл, 13, 14, ВизллБизлл, 16]

Process finished with exit code 0
```