**HW 9 for Beginners**

**Part 1 - Maven, JUnit 5.7.2**

* Все задания нужно выполнять в проекте Algorithms04, фреймворк Maven + JUnit 5.7.2.
* Код задания должен быть написан в папке main -> java.
* Тесты, проверяющие работу алгоритма, должны быть написаны в папке test -> java.
* Test Data написаны для тестирования Happy Path. Тесты с числами, указанными в TestData обязательны, и должны выполняться первыми.
* Вам нужно придумать негативные сценарии, которые могут поломать алгоритм  
    
  **Вы можете выбрать, выполнять обе части задания, или только одну часть (Part 1 или Part 2).**

1. Создать алгоритм OddEven, который принимает на вход целое число, возвращает “Odd”, если число нечетное, и “Even”, если число четное. Во всех остальных случаях результат будет “Undefined”.

Test Data:

-345 → “Odd”

0 → “Even”

222222 → “Even”

2147483647 + 1 → “Undefined”

1. Написать алгоритм OddIndices, который принимает массив чисел, и возвращает массив значений нечетных индексов

Test Data:

Input = {-45, 590, 234, 985, 12, 68}

Expected Result = {590, 985, 68}

1. Написать алгоритм SumArray, который возвращает сумму всех чисел массива

Test Data:

{0, 1, 2, 3, 4, 5} → 15

{-7, -3} → -10

1. Напишите алгоритм BiggerValue, который из двух параметров типа int возвращает бОльшее значение.

Test Data:  
3333, 9999

Expected Result = 9999

1. Напишите алгоритм IsPositiveNumber, который возвращает true, если численный параметр больше или равен нулю, и возвращает false, если параметр меньше 0.

Проверьте работу метода на числах 555, 0 и -555.

1. Напишите алгоритм AreNumbersEqual, который принимает на вход 2 любых int числа, и возвращает

0, если числа равны

-1, если первое число меньше второго

1, если первое число больше второго

Test Data:

89, 89

Expected result: 0

-89, 89

Expected result: -1

89, -89

Expected result: 1

1. Given an integer M perform the following conditional actions:

* If M is multiple of 3 and 5 then return "Good Number".
* If M is only multiple of 3 and not of 5 then return "Bad Number"
* If M is only multiple of 5 and not of 3 then return "Poor Number"
* If M doesn't satisfy any of the above conditions then return "-1"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Part 2  
  
8. Написать алгоритм MinMaxAve, который принимает массив чисел int[] и 2 значения индексов. Алгоритм возвращает массив, который содержит минимальное значение, максимальное значение, и среднее среди всех значений между 2-мя индексами.

Test Data:

({1, 2, 3, 4, 5, 6, 7, 8}, 2, 6) → {3, 7, 5}

9. Написать алгоритм Intersection, который принимает на вход 2 массива целых чисел и возвращает массив общих элементов.  
  
Test Data:

{1, 2, 4, 5, 89}, {8, 9, 4, 2} → {2, 4}  
{1, 2, 4, 5, 8, 9}, {8, 9, -4, -2} → {8, 9}

{1, 2, 4, 5, 89}, {8, 9, 45} → {}

10. Написать алгоритм PeakElement, который принимает на вход массив целых чисел и возвращает значения пиковых элементов (элементы, которые больше своих соседей).

Test Data:  
{3, 2, 7, 5, 1, 9, 23, 1} → {3, 7, 23}

11. Написать алгоритм ReverseArray, который принимает на вход массив целых чисел, и возвращает “перевернутый” массив.

Test Data:  
{2, 7, 3, 10} → {10, 3, 7, 2}

12. Написать алгоритм SortArray, который принимает на вход массив целых чисел, и сортирует элементы массива в порядке возрастания.  
Test Data:

{4, 3, 7, 12, 5, 2, 9, 4, 12} → {2, 3, 4, 4, 5, 7, 9, 12, 12}

13. Написать алгоритм KthLargest, который принимает на вход массив целых чисел и число k, и возвращает k-тый максимальный элемент.  
Test Data:

({4, 3, 7, 12, 5, 2, 9, 4, 12}, 3) → 9  
 (12 и 12 - первый и второй самые большие элементы)

13. Написать алгоритм NegativeOnTheRight, который принимает на вход массив целых чисел, и возвращает массив, в котором все негативные числа находятся во второй половине массива  
Test Data:

{4, -3, 7, -12, 5, -2, 9, 4, 12} → {4, 7, 5, 9, 4, 12, -3, -12, -2}

14. Написать алгоритм SumOfTwo, который принимает на вход массив целых чисел, и число n. Алгоритм возвращает пары элементов, которые в сумме дают число n.  
Test Data:  
({4, 3, 7, 12, 5, 2, 9, 4, 12}, 12) → {{3, 9}, {7, 5}}

15. Написать алгоритм NumberOccurrences, который принимает на вход массив целых чисел, и возвращает массив пар   
{число, сколько раз число встречается в массиве}  
 Test Data:  
{3, 2, 5, 3, 1, 5, 4, 2, 1, 4, 5, 3, 2, 1, 4, 5, 3, 2, 1} →   
{{1, 4}, {2, 4}, {3, 4}, {4, 3}, {5, 4}}